Logika i teoria typów

Wykład 10

14 grudnia 2022

Tydzień temu: logika liniowa

With (wraz): An object of type $\alpha \,\&\, \beta$ is a "virtual" pair of objects (one of type α, the other of type β), of which exactly one can be potentially created from the same resources. In other words, $\alpha \,\&\, \beta$ is a "right of choice" between α or β. This right belongs to the consumer.

$$\frac{\Gamma, \alpha \vdash \rho, \Sigma}{\Gamma, \alpha \,\&\, \beta \vdash \rho, \Sigma}\ (\mathrm{L}\&) \qquad \frac{\Gamma \vdash \alpha, \Sigma \quad \Gamma \vdash \beta, \Sigma}{\Gamma \vdash \alpha \,\&\, \beta, \Sigma}\ (\mathrm{R}\&)$$

Plus: An object of type $\alpha \oplus \beta$ is a pair consisting of an object of type α or of type β, and a flag showing which case actually holds. The right of choice between α or β belongs to the producer. The consumer opens a box and uses the contents according to the instruction on the flag.

$$\frac{\Gamma, \varphi \vdash \Sigma \quad \Gamma, \psi \vdash \Sigma}{\Gamma, \varphi \oplus \psi \vdash \Sigma}\ (\mathrm{L}\oplus) \qquad \frac{\Gamma \vdash \varphi_i, \Sigma}{\Gamma \vdash \varphi_1 \oplus \varphi_2, \Sigma}\ (\mathrm{R}\oplus)$$

Duality: Plus ($\oplus$) is the other side of With ($\&$):

$$(\alpha \oplus \beta)^\perp \multimap\!\!\multimap \alpha^\perp \,\&\, \beta^\perp \qquad (\alpha \,\&\, \beta)^\perp \multimap\!\!\multimap \alpha^\perp \oplus \beta^\perp$$

(Receiving a surprise is sending the right of choice.)

Tensor

An object of type $\alpha \otimes \beta$ is a pair of objects: one of type α, the other of type β. Creating each component of the pair requires separate resources. Consuming a pair requires using both components.

$$\frac{\Sigma, \alpha, \beta \vdash \rho, \Pi}{\Sigma, \alpha \otimes \beta \vdash \rho, \Pi}\ (L\otimes) \qquad \frac{\Gamma \vdash \alpha, \Pi \quad \Delta \vdash \beta, \Sigma}{\Gamma, \Delta \vdash \alpha \otimes \beta, \Pi, \Sigma}\ (R\otimes)$$

What is dual to $\otimes$?

What is dual to $\otimes$?

To powinien być spójnik $\parr$ o takich własnościach:

$$(\alpha \otimes \beta)^\perp \multimap\!\!\multimap \alpha^\perp \parr \beta^\perp \qquad (\alpha \parr \beta)^\perp \multimap\!\!\multimap \alpha^\perp \otimes \beta^\perp$$

Z tych własności można wywnioskować reguły:

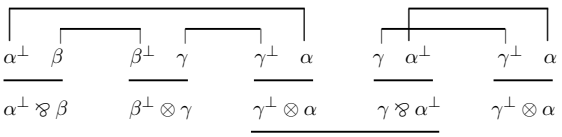
$$\frac{\Gamma, \alpha \vdash \Sigma \quad \Delta, \beta \vdash \Pi}{\Gamma, \Delta, \alpha \parr \beta \vdash \Sigma, \Pi}\ (\mathrm{L}\parr) \qquad \frac{\Gamma \vdash \alpha, \beta, \Sigma}{\Gamma \vdash \alpha \parr \beta, \Sigma}\ (\mathrm{R}\parr)$$

Par: Type $\alpha \parr \beta$ represents *communication*: a "fair contract" between α and β. (Receiving a pair of type $\alpha \otimes \beta$ is the same as sending $\alpha^\perp \parr \beta^\perp$ i.e., sending entanglement of $\alpha^\perp$ and $\beta^\perp$.)

$$(\alpha \multimap \beta) \multimap\!\!\multimap \alpha^\perp \parr \beta \qquad (\alpha \multimap \beta)^\perp \multimap\!\!\multimap \alpha \otimes \beta^\perp$$

Sieć dowodowa

Wejście typu φ to to samo co wyjście typu $\varphi^\perp$.



Nierozstrzygalność

Konfiguracja $\langle q, m, n\rangle$ automatu dwulicznikowego może być reprezentowana przez liniowy osąd postaci:

$$!\Gamma, q_f, c^m, d^n \vdash q,$$

gdzie Γ zawiera instrukcje automatu:

- $p \multimap q$: przejdź ze stanu q do stanu p;
- $(c \multimap p) \multimap q$: jak wyżej i dodaj 1 do licznika c;
- $c \multimap p \multimap q$: jak wyżej i odejmij 1 od licznika c.
- $p_1 \,\&\, p_2 \multimap q$: przejdź do obu stanów p_1 i p_2.

Test na zero? Z tym gorzej.

Ale to może być automat alternujący!

Alternation can test for zero.

Assume that the automaton only accepts in $\langle q_f, 0, 0\rangle$.

Replace "`if` $c = 0$ `then` *go to A* `else` *go to B*" by:

1: *go to 2* `or` *go to 3*;

2: $c := c - 1$; $c := c + 1$; *go to B*;

3: *go to A* `and` *go to 4*;

4: *go to 5* `or` *go to* q_f;

5: $d := d - 1$; *go to 4*.

Główny lemat

Sekwent $!\Gamma, q_f, c^m, d^n \vdash q$ ma dowód

wtedy i tylko wtedy, gdy

automat akceptuje konfigurację $\langle q, m, n\rangle$.

The boring part

Assume that $c := c - 1$; goto p is the instruction in state q. This is represented by the assumption $!(c \multimap p \multimap q)$ in $!\Gamma$.

Suppose that $!\Gamma, q_f, c^m, d^n \vdash q$ is provable.

Then there is a "focused" proof of (essentially) the form

$$\dfrac{\dfrac{c \vdash c \qquad \dfrac{!\Gamma, q_f, c^{m-1}, d^n \vdash p \quad q \vdash q}{!\Gamma, q_f, c^{m-1}, d^n, p \multimap q \vdash q}}{!\Gamma, q_f, c^m, d^n, c \multimap p \multimap q \vdash q}}{\vdots}$$
$$!\Gamma, q_f, c^m, d^n \vdash q$$

Other fragments

- Nasze kodowanie automatu dowodzi nierozstrzygalności fragmentu[1] $\&$, $!$, $\multimap$.
- Fragment $\oplus$, $\otimes$, $!$, $\multimap$ też jest nierozstrzygalny.
- Fragment addytywno-multiplikatywny (bez $!$, $?$) jest Pspace-zupełny.
- Nie wiadomo, czy fragment $\otimes$, $!$, $\multimap$ jest rozstrzygalny. Ten fragment jest co najmniej tak trudny jak osiągalność w sieciach Petriego... ... a więc niepierwotny (akermańsko trudny). (Czerwiński, Orlikowski, FOCS 2021)

[1] Zob. też Forster, Larchey-Wendling, CPP 2019.

Logika pierwszego rzędu

Logika pierwszego rzędu: składnia

Sygnatura: Ustalone symbole relacyjne, funkcyjne i stałe.

Termy: Zmienne $a, b, \ldots$, stałe, wyrażenia $ft_1 \ldots t_n$.

Formuły:

– atomowe: $rt_1 \ldots t_n$, $\bot$;

– $\varphi \to \psi$, $\varphi \vee \psi$, $\varphi \wedge \psi$;

– $\forall a\, \varphi$, $\exists a\, \varphi$.

Zmienne wolne: $\mathrm{FV}(\forall a\, \varphi) = \mathrm{FV}(\exists a\, \varphi) = \mathrm{FV}(\varphi) - \{a\}$.

Utożsamiamy formuły różniące się tylko zmiennymi związanymi.

Interpretacja BHK

- A construction of $\forall a\, \varphi(a)$ is a method that turns any possible value d of a into a construction of $\varphi(\mathrm{d})$;
- A construction of $\exists a\, \varphi$ is a pair consisting of a value d of a and a construction of $\varphi(\mathrm{d})$.

Przykłady niewątpliwe

- $\forall a(\varphi \to \psi) \to (\forall a \varphi \to \forall a \psi)$;
- $\forall a(\varphi \to \psi) \to (\exists a \varphi \to \exists a \psi)$;
- $\neg \exists a \varphi \leftrightarrow \forall a \neg \varphi$;
- $\forall a \varphi \to \exists a \varphi$;
- $\exists a(\psi \wedge \varphi(a)) \leftrightarrow \psi \wedge \exists a \varphi(a)$, gdy $a \notin \mathrm{FV}(\psi)$.

Przykłady wątpliwe

- $\neg \forall a \varphi \leftrightarrow \exists a \neg \varphi$;
- $\forall a(\psi \vee \varphi(a)) \to \psi \vee \forall a\, \varphi(a)$;
- $\neg\neg \forall a\, (\varphi(a) \vee \neg \varphi(a))$;
- $\exists b(\varphi(b) \to \forall a \varphi(a))$;
- $\exists a(\exists b \varphi(b) \to \varphi(a))$.

Naturalna dedukcja

$$(\forall \mathrm{I})\ \frac{\Gamma \vdash \varphi}{\Gamma \vdash \forall a\, \varphi}\ (a \notin \mathrm{FV}(\Gamma)) \qquad (\forall \mathrm{E})\ \frac{\Gamma \vdash \forall a\, \varphi}{\Gamma \vdash \varphi[a := t]}$$

$$(\exists \mathrm{I})\ \frac{\Gamma \vdash \varphi[a := t]}{\Gamma \vdash \exists a\, \varphi} \qquad (\exists \mathrm{E})\ \frac{\Gamma \vdash \exists a\, \varphi \quad \Gamma, \varphi \vdash \psi}{\Gamma \vdash \psi}\ (*)$$

$(*)$ $a \notin \mathrm{FV}(\Gamma, \psi)$

(Te same reguły dla logiki klasycznej.)

Rachunek sekwentów

$$(\mathrm{L}\forall)\ \frac{\Gamma, \varphi[a := t] \vdash \sigma}{\Gamma, \forall a \varphi \vdash \sigma} \qquad (\mathrm{R}\forall)\ \frac{\Gamma \vdash \varphi}{\Gamma \vdash \forall a \varphi}\ (a \notin \mathrm{FV}(\Gamma))$$

$$(\mathrm{L}\exists)\ \frac{\Gamma, \varphi \vdash \sigma}{\Gamma, \exists a \varphi \vdash \sigma}\ (a \notin \mathrm{FV}(\Gamma, \sigma)) \qquad (\mathrm{R}\exists)\ \frac{\Gamma \vdash \varphi[a := t]}{\Gamma \vdash \exists a \varphi}$$

(Podobne reguły dla logiki klasycznej.)

Rachunek sekwentów

Twierdzenie o eliminacji cięcia pozostaje prawdziwe.

Wniosek 1: Jeśli $\vdash \exists a\, \varphi$, to istnieje takie t, że $\vdash \varphi[a := t]$.

Dowód: Ostatnia musiała być reguła $(\mathrm{R}\exists)$.

Wniosek 2: Rachunek pierwszego rzędu jest konserwatywny nad rachunkiem zdań.

Dowód: Z zasady podformuł: dowód bez cięcia dla formuły bez kwantyfikatorów nie zawiera kwantyfikatorów.

Translacja Kołmogorowa

$$\begin{aligned}
k(rt_1 \dots t_n) &= \neg\neg\, rt_1 \dots t_n \\
k(\bot) &= \bot \\
k(\tau \vee \sigma) &= \neg\neg(k(\tau) \vee k(\sigma)) \\
k(\tau \to \sigma) &= \neg\neg(k(\tau) \to k(\sigma)) \\
k(\tau \wedge \sigma) &= \neg\neg(k(\tau) \wedge k(\sigma)) \\
k(\forall a\, \tau) &= \neg\neg \forall a\, k(\tau) \\
k(\exists a\, \tau) &= \neg\neg \exists a\, k(\tau)
\end{aligned}$$

Translacja Kołmogorowa

Twierdzenie:

$$\Gamma \vdash_{klas} \alpha \qquad \Leftrightarrow \qquad k(\Gamma) \vdash_{int} k(\alpha).$$

Dowód: ($\Rightarrow$) Ćwiczenia?

($\Leftarrow$) Formuły α i $k(\alpha)$ są klasycznie równoważne.

Wniosek: Logika intuicjonistyczna pierwszego rzędu jest nierozstrzygalna. Klasyczny Entscheidungsproblem redukuje się do intuicjonistycznego.

Rachunek lambda (1)

$$(\forall \mathrm{I})\ \frac{\Gamma \vdash M : \varphi}{\Gamma \vdash \lambda a\, M : \forall a\, \varphi}\ (a \notin FV(\Gamma))$$

$$(\forall \mathrm{E})\ \frac{\Gamma \vdash M : \forall a\, \varphi}{\Gamma \vdash Mt : \varphi[a := t]}$$

$$(\lambda a\, M)t \Rightarrow_\beta M[a := t]$$

Twierdzenie: *ten rachunek ma własność silnej normalizacji.*

Typy zależne (*Dependent types*)

$P(\mathrm{a}_1, \dots, \mathrm{a}_n)$ — a type that *depends* on the choice of individual values $\mathrm{a}_1, \dots, \mathrm{a}_n$.

$\forall a\, P(a)$ — a type of a function that turns an arbitrary value **a** into something of type $P(\mathrm{a})$.

Typy zależne (*Dependent types*)

$M : \mathrm{array}(n)$	M jest tablicą rozmiaru n;
$\mathrm{array} : \mathrm{Int} \to *$	array jest *konstruktorem typu*
$Samezera(n) : \mathrm{array}(n)$	tablica z samych zer;
$Samezera : \mathrm{Int} \to \mathrm{array}(?)$	*tak się nie da...*
$Samezera : (n : \mathrm{Int}) \to \mathrm{array}(n)$	

Typy zależne (*Dependent types*)

$M : \mathrm{array}(n)$	M jest tablicą rozmiaru n;
$\mathrm{array} : \mathrm{Int} \to *$	array jest *konstruktorem typu*
$Samezera(n) : \mathrm{array}(n)$	tablica z samych zer;
$Samezera : \mathrm{Int} \to \mathrm{array}(?)$	*tak się nie da...*
$Samezera : \Pi n : \mathrm{Int}.\ \mathrm{array}(n)$	

Typy zależne (*Dependent types*)

$M : \mathrm{array}(n)$ — M jest tablicą rozmiaru n;

$\mathrm{array} : \mathrm{Int} \to *$ — array jest *konstruktorem typu*

$\mathit{Samezera}(n) : \mathrm{array}(n)$ — tablica z samych zer;

$\mathit{Samezera} : \mathrm{Int} \to \mathrm{array}(?)$ — *tak się nie da...*

$\mathit{Samezera} : \forall n : \mathrm{Int}.\, \mathrm{array}(n)$

Rachunek lambda (2)

$$(\exists \mathrm{I})\ \frac{\Gamma \vdash M : \varphi[a := t]}{\Gamma \vdash [t, M] : \exists a\, \varphi}$$

$$(\exists \mathrm{E})\ \frac{\Gamma \vdash M : \exists a\, \varphi \quad \Gamma, y : \varphi \vdash N : \psi}{\Gamma \vdash \mathrm{let}\ M = [a, y{:}\varphi]\ \mathrm{in}\ N : \psi}\ (a \notin FV(\Gamma \cup \{\psi\}))$$

Typy egzystencjalne = abstrakcyjne

$[t, M] : \exists a\, \varphi(a)$ – spakowana implementacja $M : \varphi(t)$

$\mathrm{let}\ M = [a, y{:}\varphi]\ \mathrm{in}\ N^{\psi} : \psi$

– użycie danej implementacji M w kontekście N.

Redukcja: $\mathrm{let}\ [t, M] = [a, y{:}\varphi]\ \mathrm{in}\ N \Rightarrow_\beta N[a := t][y := M]$

$\mathrm{Array} = \exists n.\, \mathrm{array}(n)$

$\mathit{Suma} : \mathrm{Array} \to \mathrm{int}$

$\mathit{Append} : \mathrm{Array} \to \mathrm{Array} \to \mathrm{Array}$

„Zależny iloczyn kartezjański"

Jeśli $n : \mathrm{int}$ i $M : \varphi(n)$,
to para uporządkowana $[t, M]$ jest typu $\mathrm{int} \times ..??..$

Można to napisać tak: $[t, M] : (n : \mathrm{int}) \times \varphi(n)$.

$\forall n : \mathrm{int}.\, \varphi(n) \ = \ (n : \mathrm{int}) \to \varphi(n)$
$\exists n : \mathrm{int}.\, \varphi(n) \ = \ (n : \mathrm{int}) \times \varphi(n)$

Kwantyfikator egzystencjalny tak się ma do koniunkcji
jak ogólny do implikacji.

Koprodukt

Jeszcze inaczej:

$\forall n : \mathrm{int}.\, \varphi(n) \ = \ (n : \mathrm{int}) \to \varphi(n)$
$\forall n : \mathrm{int}.\, \varphi(n) \ = \ \Pi n : \mathrm{int}.\, \varphi(n)$

$\exists n : \mathrm{int}.\, \varphi(n) \ = \ (n : \mathrm{int}) \times \varphi(n)$
$\exists n : \mathrm{int}.\, \varphi(n) \ = \ \Sigma n : \mathrm{int}.\, \varphi(n)$

Curry-Howard

Theorem
A first-order formula α is intuitionistically valid if and only if there exists M such that $\vdash M : \alpha$.

Warning: A normal inhabitant $\lambda x{:}\forall a\alpha.[b, xb]$ of $\forall a\alpha \to \exists a\alpha$ has free variable b.

W logice pierwszego rzędu obowiązuje dogmat o niepustości dziedziny. Ten dogmat nie zawsze jest odpowiedni.

Dlatego w Coqu zmienne trzeba jawnie deklarować.

Semantyka algebraiczna

Definicje:

Algebra Heytinga jest *zupełna*, wtedy i tylko wtedy, gdy każdy podzbiór ma kres górny i dolny.

Niech $\mathcal{H}$ – (zupełna) algebra Heytinga. Wtedy *$\mathcal{H}$-strukturą* nazywamy twór postaci $\mathcal{A} = \langle A, f^A, g^A, \ldots, r^A, s^A, \ldots \rangle$, gdzie $r^A : A^{n_r} \to \mathcal{H}$, $s^A : A^{n_s} \to \mathcal{H}$,...

(Relacje rozumiemy jako funkcje o wartościach w $\mathcal{H}$.)

Semantyka algebraiczna

Znaczenie formuły φ w $\mathcal{H}$-strukturze: $[\![\varphi]\!]_\zeta \in \mathcal{H}$.

- $[\![r(t_1, \ldots, t_n)]\!]_\zeta = r^{\mathcal{A}}(\zeta(t_1), \ldots, \zeta(t_n))$;
- $[\![\alpha \to \beta]\!]_\zeta = [\![\alpha]\!]_\zeta \Rightarrow [\![\beta]\!]_\zeta$;
- $[\![\alpha \vee \beta]\!]_\zeta = [\![\alpha]\!]_\zeta \cup [\![\beta]\!]_\zeta$;
- $[\![\alpha \wedge \beta]\!]_\zeta = [\![\alpha]\!]_\zeta \cap [\![\beta]\!]_\zeta$;
- $[\![\bot]\!]_\zeta = 0$;
- $[\![\exists a\, \varphi]\!]_\zeta = \sup_{d \in A} [\![\varphi]\!]_{\zeta[a \mapsto d]}$
- $[\![\forall a\, \varphi]\!]_\zeta = \inf_{d \in A} [\![\varphi]\!]_{\zeta[a \mapsto d]}$

Piszemy $\mathcal{A}, \zeta \models \varphi$, gdy $[\![\varphi]\!]_\zeta = 1_{\mathcal{H}}$.

Twierdzenie o pełności

Twierdzenie:

1) Jeśli $\Gamma \vdash \varphi$, to $\Gamma \models \varphi$.
2) Jeśli $\Gamma \models \varphi$, to $\Gamma \vdash \varphi$.

Dowód: (1) Indukcja.

(2) Budujemy algebrę formuł $\mathcal{L}_\Gamma$: $[\alpha]_\sim = \{\beta \mid \Gamma \vdash \alpha \leftrightarrow \beta\}$

W tej algebrze zachodzi:

$[\forall a\, \alpha(a)]_\sim = \inf_t [\alpha(t)]_\sim$ oraz $[\exists a\, \alpha(a)]_\sim = \sup_t [\alpha(t)]_\sim$

W algebrze termów $\mathcal{A}$ niech $r^{\mathcal{A}}(t_1, \ldots, t_n) = [r t_1 \ldots t_n]_\sim$.
Wtedy $\mathcal{A}$ jest $\mathcal{L}_\Gamma$-strukturą i $[\![\varphi]\!]_{id} = [\varphi]_\sim$ dla każdego φ.

Jeśli $\Gamma \models \varphi$, to $[\![\varphi]\!]_{id} = 1$ i dobrze. . . ??

Nie bardzo: algebra $\mathcal{L}_\Gamma$ nie musi być zupełna.