

Logika i teoria typów

Wykład 7

30 listopada 2020

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), from 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.

$$\frac{\Gamma, \alpha \vdash \rho, \Sigma}{\Gamma, \alpha \& \beta \vdash \rho, \Sigma} \text{ (L}\oplus\text{)}$$

$$\frac{\Gamma \vdash \alpha, \Sigma \quad \Gamma \vdash \beta, \Sigma}{\Gamma \vdash \alpha \& \beta, \Sigma} \text{ (R}\oplus\text{)}$$

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.

$$\frac{\Gamma, \varphi \vdash \Sigma \quad \Gamma, \psi \vdash \Sigma}{\Gamma, \varphi \oplus \psi \vdash \Sigma} \text{ (L}\oplus\text{)}$$

$$\frac{\Gamma \vdash \varphi_i, \Sigma}{\Gamma \vdash \varphi_1 \oplus \varphi_2, \Sigma} \text{ (R}\oplus\text{)}$$

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

$$(\alpha \oplus \beta)^\perp \multimap \multimap \alpha^\perp \& \beta^\perp$$

$$(\alpha \& \beta)^\perp \multimap \multimap \alpha^\perp \oplus \beta^\perp$$

(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.

$$\frac{\Sigma, \alpha, \beta \vdash \rho, \Pi}{\Sigma, \alpha \otimes \beta \vdash \rho, \Pi} (L\otimes)$$

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

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.

$$\frac{\Sigma, \alpha, \beta \vdash \rho, \Pi}{\Sigma, \alpha \otimes \beta \vdash \rho, \Pi} (L\otimes)$$

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

What is dual to \otimes ?

What is dual to \otimes ?

What is dual to \otimes ?

$$(\alpha \otimes \beta)^\perp \dashv\vdash \alpha^\perp \wp \beta^\perp$$

$$(\alpha \wp \beta)^\perp \dashv\vdash \alpha^\perp \otimes \beta^\perp$$

$$\frac{\Gamma, \alpha \vdash \Sigma \quad \Delta, \beta \vdash \Pi}{\Gamma, \Delta, \alpha \wp \beta \vdash \Sigma, \Pi} \text{ (L}\wp\text{)}$$

$$\frac{\Gamma \vdash \alpha, \beta, \Sigma}{\Gamma \vdash \alpha \wp \beta, \Sigma} \text{ (R}\wp\text{)}$$

What is dual to \otimes ?

$$(\alpha \otimes \beta)^\perp \multimap \alpha^\perp \wp \beta^\perp$$

$$(\alpha \wp \beta)^\perp \multimap \alpha^\perp \otimes \beta^\perp$$

$$\frac{\Gamma, \alpha \vdash \Sigma \quad \Delta, \beta \vdash \Pi}{\Gamma, \Delta, \alpha \wp \beta \vdash \Sigma, \Pi} \text{ (L}\wp\text{)}$$

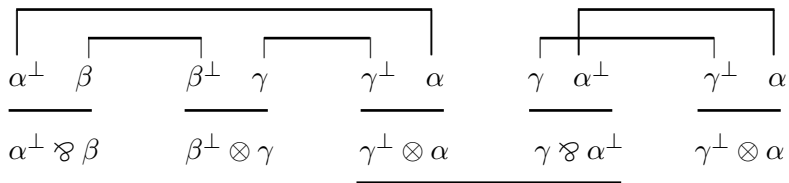
$$\frac{\Gamma \vdash \alpha, \beta, \Sigma}{\Gamma \vdash \alpha \wp \beta, \Sigma} \text{ (R}\wp\text{)}$$

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

$$(\alpha \multimap \beta) \multimap \alpha^\perp \wp \beta$$

Sieć dowodowa

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



Translacja Girarda

- ▶ $p^* = p, \top^* = 1, \perp^* = 0$;
- ▶ $(\alpha \wedge \beta)^* = \alpha^* \& \beta^*$;
- ▶ $(\alpha \vee \beta)^* = !\alpha^* \oplus !\beta^*$;
- ▶ $(\alpha \rightarrow \beta)^* = !\alpha^* \multimap \beta^*$.

Twierdzenie:

Następujące warunki są równoważne:

- 1) $\Gamma \vdash \varphi$ w logice intuicjonistycznej;
- 2) $!\Gamma^* \vdash \varphi^*$ w intuicjonistycznej logice liniowej;
- 3) $!\Gamma^* \vdash \varphi^*$ w klasycznej logice liniowej.

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 .

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 .

Test na zero?

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 .

Test na zero? Z tym gorzej.

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 $c_i := c_i - 1$; `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 (essentially) of the form

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

Other fragments

- ▶ Nasze kodowanie automatu dowodzi nierozstrzygalności fragmentu $\&$, $!$, \neg .

Other fragments

- ▶ Nasze kodowanie automatu dowodzi nierozstrzygalności fragmentu $\&$, $!$, \neg .
- ▶ Fragment \oplus , \otimes , $!$, \neg też jest nierozstrzygalny.

Other fragments

- ▶ Nasze kodowanie automatu dowodzi nierozstrzygalności fragmentu $\&$, $!$, \neg .
- ▶ Fragment \oplus , \otimes , $!$, \neg też jest nierozstrzygalny.
- ▶ Fragment addytywno-multiplikatywny (bez $!$, $?$) jest Pspace-zupełny.

Other fragments

- ▶ Nasze kodowanie automatu dowodzi nierozstrzygalności fragmentu $\&$, $!$, \rightarrow .
- ▶ Fragment \oplus , \otimes , $!$, \rightarrow też jest nierozstrzygalny.
- ▶ Fragment addytywno-multiplikatywny (bez $!$, $?$) jest Pspace-zupełny.
- ▶ Nie wiadomo, czy fragment \otimes , $!$, \rightarrow jest rozstrzygalny. Ten fragment jest co najmniej tak trudny jak osiągalność w sieciach Petriego...

Other fragments

- ▶ Nasze kodowanie automatu dowodzi nierozstrzygalności fragmentu $\&$, $!$, \rightarrow .
- ▶ Fragment \oplus , \otimes , $!$, \rightarrow też jest nierozstrzygalny.
- ▶ Fragment addytywno-multiplicatywny (bez $!$, $?$) jest Pspace-zupełny.
- ▶ Nie wiadomo, czy fragment \otimes , $!$, \rightarrow jest rozstrzygalny. Ten fragment jest co najmniej tak trudny jak osiągalność w sieciach Petriego...
... a więc nieelementarny. (Czerwiński *et al*, STOC 2019)

Logika pierwszego rzędu

Logika pierwszego rzędu: składnia

Sygnatura: Ustalone symbole relacyjne, funkcyjne i stałe.

Logika pierwszego rzędu: składnia

Sygnatura: Ustalone symbole relacyjne, funkcyjne i stałe.

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

Logika pierwszego rzędu: składnia

Sygnatura: Ustalony symbole relacyjne, funkcyjne i stałe.

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

Formuły:

– atomowe: $rt_1 \dots t_n, \perp$;

Logika pierwszego rzędu: składnia

Sygnatura: Ustalone symbole relacyjne, funkcyjne i stałe.

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

Formuły:

- atomowe: $rt_1 \dots t_n, \perp$;
- $\varphi \rightarrow \psi, \varphi \vee \psi, \varphi \wedge \psi$;

Logika pierwszego rzędu: składnia

Sygnatura: Ustalony symbole relacyjne, funkcyjne i stałe.

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

Formuły:

- atomowe: $rt_1 \dots t_n, \perp$;
- $\varphi \rightarrow \psi, \varphi \vee \psi, \varphi \wedge \psi$;
- $\forall a \varphi, \exists a \varphi$.

Logika pierwszego rzędu: składnia

Sygnatura: Ustalone symbole relacyjne, funkcyjne i stałe.

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

Formuły:

- atomowe: $rt_1 \dots t_n, \perp$;
- $\varphi \rightarrow \psi, \varphi \vee \psi, \varphi \wedge \psi$;
- $\forall a \varphi, \exists a \varphi$.

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

Logika pierwszego rzędu: składnia

Sygnatura: Ustalony symbole relacyjne, funkcyjne i stałe.

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

Formuły:

- atomowe: $rt_1 \dots t_n, \perp$;
- $\varphi \rightarrow \psi, \varphi \vee \psi, \varphi \wedge \psi$;
- $\forall a \varphi, \exists a \varphi$.

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

Utożsamiamy formuły różniące się tylko zmiennymi związanymi.

Interpretacja BHK

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

Przykłady niewątpliwe

- ▶ $\forall a(\varphi \rightarrow \psi) \rightarrow (\forall a\varphi \rightarrow \forall a\psi)$;

Przykłady niewątpliwe

▶ $\forall a(\varphi \rightarrow \psi) \rightarrow (\forall a\varphi \rightarrow \forall a\psi);$

▶ $\forall a(\varphi \rightarrow \psi) \rightarrow (\exists a\varphi \rightarrow \exists a\psi);$

Przykłady niewątpliwe

▶ $\forall a(\varphi \rightarrow \psi) \rightarrow (\forall a\varphi \rightarrow \forall a\psi);$

▶ $\forall a(\varphi \rightarrow \psi) \rightarrow (\exists a\varphi \rightarrow \exists a\psi);$

▶ $\neg\exists a\varphi \leftrightarrow \forall a\neg\varphi;$

Przykłady niewątpliwe

▶ $\forall a(\varphi \rightarrow \psi) \rightarrow (\forall a\varphi \rightarrow \forall a\psi);$

▶ $\forall a(\varphi \rightarrow \psi) \rightarrow (\exists a\varphi \rightarrow \exists a\psi);$

▶ $\neg\exists a\varphi \leftrightarrow \forall a\neg\varphi;$

▶ $\forall a\varphi \rightarrow \exists a\varphi;$

Przykłady niewątpliwe

▶ $\forall a(\varphi \rightarrow \psi) \rightarrow (\forall a\varphi \rightarrow \forall a\psi);$

▶ $\forall a(\varphi \rightarrow \psi) \rightarrow (\exists a\varphi \rightarrow \exists a\psi);$

▶ $\neg\exists a\varphi \leftrightarrow \forall a\neg\varphi;$

▶ $\forall a\varphi \rightarrow \exists a\varphi;$

▶ $\exists a(\psi \wedge \varphi(a)) \leftrightarrow \psi \wedge \exists a\varphi(a),$ gdy $a \notin FV(\psi).$

Przykłady wątpliwe

► $\neg\forall a\varphi \leftrightarrow \exists a\neg\varphi$;

Przykłady wątpliwe

▶ $\neg\forall a\varphi \leftrightarrow \exists a\neg\varphi$;

▶ $\forall a(\psi \vee \varphi(a)) \rightarrow \psi \vee \forall a\varphi(a)$;

Przykłady wątpliwe

▶ $\neg\forall a\varphi \leftrightarrow \exists a\neg\varphi$;

▶ $\forall a(\psi \vee \varphi(a)) \rightarrow \psi \vee \forall a\varphi(a)$;

▶ $\neg\neg\forall a(\varphi(a) \vee \neg\varphi(a))$;

Przykłady wątpliwe

- ▶ $\neg\forall a\varphi \leftrightarrow \exists a\neg\varphi$;
- ▶ $\forall a(\psi \vee \varphi(a)) \rightarrow \psi \vee \forall a\varphi(a)$;
- ▶ $\neg\neg\forall a(\varphi(a) \vee \neg\varphi(a))$;
- ▶ $\exists b(\varphi(b) \rightarrow \forall a\varphi(a))$;

Przykłady wątpliwe

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

Naturalna dedukcja

$$(\forall I) \frac{\Gamma \vdash \varphi}{\Gamma \vdash \forall a \varphi} \quad (a \notin \text{FV}(\Gamma))$$

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

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

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

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

(Te same reguły dla logiki klasycznej.)

Rachunek sekwentów

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

$$(R\forall) \frac{\Gamma \vdash \varphi}{\Gamma \vdash \forall a \varphi} \quad (a \notin FV(\Gamma))$$

$$(L\exists) \frac{\Gamma, \varphi \vdash \sigma}{\Gamma, \exists a \varphi \vdash \sigma} \quad (a \notin FV(\Gamma, \sigma))$$

$$(R\exists) \frac{\Gamma \vdash \varphi[a := t]}{\Gamma \vdash \exists a \varphi}$$

(Podobne reguły dla logiki klasycznej.)

Rachunek sekwentów

Twierdzenie o eliminacji cięcia pozostaje prawdziwe.

Rachunek sekwentów

Twierdzenie o eliminacji cięcia pozostaje prawdziwe.

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

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 $(R\exists)$.

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 $(R\exists)$.

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

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 $(R\exists)$.

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

$$k(rt_1 \dots t_n) = \neg\neg rt_1 \dots t_n$$

$$k(\perp) = \perp$$

$$k(\tau \vee \sigma) = \neg\neg(k(\tau) \vee k(\sigma))$$

$$k(\tau \rightarrow \sigma) = \neg\neg(k(\tau) \rightarrow k(\sigma))$$

$$k(\tau \wedge \sigma) = \neg\neg(k(\tau) \wedge k(\sigma))$$

$$k(\forall a \tau) = \neg\neg\forall a k(\tau)$$

$$k(\exists a \tau) = \neg\neg\exists a k(\tau)$$

Translacja Kołmogorowa

Twierdzenie:

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

Dowód: (\Rightarrow) Ćwiczenia.

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

Translacja Kołmogorowa

Twierdzenie:

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

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)

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

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

$$(\lambda a.M)t \Rightarrow_{\beta} M[a := t]$$

Twierdzenie: *ten rachunek ma własność silnej normalizacji.*

Typy zależne (*Dependent types*)

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

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

Typy zależne (*Dependent types*)

$M : \mathbf{array}(n)$

M jest tablicą rozmiaru n ;

Typy zależne (*Dependent types*)

$M : \text{array}(n)$

M jest tablicą rozmiaru n ;

$\text{array} : \text{Int} \rightarrow \text{„Typ”}$

array jest *konstruktorem typu*
(*rodziną typów*)

Typy zależne (*Dependent types*)

$M : \mathbf{array}(n)$

M jest tablicą rozmiaru n ;

$\mathbf{array} : \mathbf{Int} \rightarrow \text{„Typ”}$

\mathbf{array} jest *konstruktorem typu*
(*rodziną typów*)

$\mathbf{Samezera}(n) : \mathbf{array}(n)$

tablica z samych zer;

Typy zależne (*Dependent types*)

$M : \mathbf{array}(n)$

M jest tablicą rozmiaru n ;

$\mathbf{array} : \mathbf{Int} \rightarrow \text{„Typ”}$

\mathbf{array} jest *konstruktorem typu*
(*rodziną typów*)

$\mathbf{Samezera}(n) : \mathbf{array}(n)$

tablica z samych zer;

$\mathbf{Samezera} : \mathbf{Int} \rightarrow \mathbf{array}(?)$

tak się nie da...

Typy zależne (*Dependent types*)

$M : \mathbf{array}(n)$

M jest tablicą rozmiaru n ;

$\mathbf{array} : \mathbf{Int} \rightarrow \text{„Typ”}$

\mathbf{array} jest *konstruktorem typu*
(*rodziną typów*)

$\mathit{Samezera}(n) : \mathbf{array}(n)$

tablica z samych zer;

$\mathit{Samezera} : \mathbf{Int} \rightarrow \mathbf{array}(?)$

tak się nie da...

$\mathit{Samezera} : (n : \mathbf{Int}) \rightarrow \mathbf{array}(n)$

Typy zależne (*Dependent types*)

$M : \mathbf{array}(n)$

M jest tablicą rozmiaru n ;

$\mathbf{array} : \mathbf{Int} \rightarrow \text{„Typ”}$

\mathbf{array} jest *konstruktorem typu*
(*rodziną typów*)

$\mathit{Samezera}(n) : \mathbf{array}(n)$

tablica z samych zer;

$\mathit{Samezera} : \mathbf{Int} \rightarrow \mathbf{array}(?)$

tak się nie da...

$\mathit{Samezera} : \prod n : \mathbf{Int}. \mathbf{array}(n)$

To się nazywa produkt.

Typy zależne (*Dependent types*)

$M : \mathbf{array}(n)$

M jest tablicą rozmiaru n ;

$\mathbf{array} : \mathbf{Int} \rightarrow \text{„Typ”}$

\mathbf{array} jest *konstruktorem typu*
(rodziną typów)

$\mathbf{Samezera}(n) : \mathbf{array}(n)$

tablica z samych zer;

$\mathbf{Samezera} : \mathbf{Int} \rightarrow \mathbf{array}(?)$

tak się nie da...

$\mathbf{Samezera} : \forall n : \mathbf{Int}. \mathbf{array}(n)$

To się nazywa produkt.

Rachunek lambda (2)

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

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

Typy egzystencjalne = abstrakcyjne

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

unpack M as $[a, y:\varphi]$ in $N^\psi : \psi$

– użycie danej implementacji M w kontekście N .

Typy egzystencjalne = abstrakcyjne

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

unpack M as $[a, y:\varphi]$ in $N^\psi : \psi$

– użycie danej implementacji M w kontekście N .

Redukcja:

unpack $[t, M]$ as $[a, y:\varphi]$ in $N \Rightarrow_\beta N[a := t][y := M]$

Typy egzystencjalne = abstrakcyjne

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

unpack M as $[a, y:\varphi]$ in $N^\psi : \psi$

– użycie danej implementacji M w kontekście N .

Redukcja:

unpack $[t, M]$ as $[a, y:\varphi]$ in $N \Rightarrow_\beta N[a := t][y := M]$

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

Suma : **Array** \rightarrow **int**

Append : **Array** \rightarrow **Array** \rightarrow **Array**

„Zależny iloczyn kartezjański”

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

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

$$\forall n : \mathbf{int}. \varphi(n) = (n : \mathbf{int}) \rightarrow \varphi(n)$$

$$\exists n : \mathbf{int}. \varphi(n) = (n : \mathbf{int}) \times \varphi(n)$$

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

Koprodukt

Jeszcze inaczej:

$$\forall n : \mathbf{int}. \varphi(n) = (n : \mathbf{int}) \rightarrow \varphi(n)$$

$$\forall n : \mathbf{int}. \varphi(n) = \prod n : \mathbf{int}. \varphi(n)$$

$$\exists n : \mathbf{int}. \varphi(n) = (n : \mathbf{int}) \times \varphi(n)$$

$$\exists n : \mathbf{int}. \varphi(n) = \sum n : \mathbf{int}. \varphi(n)$$

Curry-Howard

Theorem

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

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 $\forall a\alpha \rightarrow \exists a\alpha$ has free variable b .

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 $\forall a\alpha \rightarrow \exists a\alpha$ 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 deklorować.

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 $\mathcal{A} = \langle A, f^A, g^A, \dots, r^A, s^A, \dots \rangle$, gdzie $r^A : A^{n_r} \rightarrow \mathcal{H}$, $s^A : A^{n_s} \rightarrow \mathcal{H}, \dots$

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

Semantyka algebraiczna

Znaczenie formuły φ w \mathcal{H} -strukturze: $\llbracket \varphi \rrbracket_{\zeta} \in \mathcal{H}$.

- $\llbracket r(t_1, \dots, t_n) \rrbracket_{\zeta} = r^A(\zeta(t_1), \dots, \zeta(t_n));$
- $\llbracket \alpha \rightarrow \beta \rrbracket_{\zeta} = \llbracket \alpha \rrbracket_{\zeta} \Rightarrow \llbracket \beta \rrbracket_{\zeta};$
- $\llbracket \alpha \vee \beta \rrbracket_{\zeta} = \llbracket \alpha \rrbracket_{\zeta} \cup \llbracket \beta \rrbracket_{\zeta};$
- $\llbracket \alpha \wedge \beta \rrbracket_{\zeta} = \llbracket \alpha \rrbracket_{\zeta} \cap \llbracket \beta \rrbracket_{\zeta};$
- $\llbracket \perp \rrbracket_{\zeta} = 0;$
- $\llbracket \exists a \varphi \rrbracket_{\zeta} = \sup_{a \in A} \llbracket \varphi \rrbracket_{\zeta[a \mapsto a]}$
- $\llbracket \forall a \varphi \rrbracket_{\zeta} = \inf_{a \in A} \llbracket \varphi \rrbracket_{\zeta[a \mapsto a]}$

Piszemy $\mathcal{A}, \zeta \models \varphi$, gdy $\llbracket \varphi \rrbracket_{\zeta} = 1_{\mathcal{H}}$.

Twierdzenie o pełności

Twierdzenie:

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

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.

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}_Γ : $[\alpha]_\sim = \{\beta \mid \Gamma \vdash \alpha \leftrightarrow \beta\}$

W tej algebrze zachodzi:

$$[\forall a \alpha(a)]_\sim = \inf_t [\alpha(t)]_\sim \quad \text{oraz} \quad [\exists a \alpha(a)]_\sim = \sup_t [\alpha(t)]_\sim$$

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}_Γ : $[\alpha]_\sim = \{\beta \mid \Gamma \vdash \alpha \leftrightarrow \beta\}$

W tej algebrze zachodzi:

$$[\forall a \alpha(a)]_\sim = \inf_t [\alpha(t)]_\sim \quad \text{oraz} \quad [\exists a \alpha(a)]_\sim = \sup_t [\alpha(t)]_\sim$$

W algebrze termów \mathcal{A} niech $r^{\mathcal{A}}(t_1, \dots, t_n) = [rt_1 \dots t_n]_\sim$.
Wtedy \mathcal{A} jest \mathcal{L}_Γ -strukturą i $\llbracket \varphi \rrbracket_{id} = [\varphi]_\sim$ dla każdego φ .

Jeśli $\Gamma \models \varphi$, to $\llbracket \varphi \rrbracket_{id} = 1$ i dobrze...

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}_Γ : $[\alpha]_\sim = \{\beta \mid \Gamma \vdash \alpha \leftrightarrow \beta\}$

W tej algebrze zachodzi:

$$[\forall a \alpha(a)]_\sim = \inf_t [\alpha(t)]_\sim \quad \text{oraz} \quad [\exists a \alpha(a)]_\sim = \sup_t [\alpha(t)]_\sim$$

W algebrze termów \mathcal{A} niech $r^{\mathcal{A}}(t_1, \dots, t_n) = [rt_1 \dots t_n]_\sim$.
Wtedy \mathcal{A} jest \mathcal{L}_Γ -strukturą i $\llbracket \varphi \rrbracket_{id} = [\varphi]_\sim$ dla każdego φ .

Jeśli $\Gamma \models \varphi$, to $\llbracket \varphi \rrbracket_{id} = 1$ i dobrze...??

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}_Γ : $[\alpha]_\sim = \{\beta \mid \Gamma \vdash \alpha \leftrightarrow \beta\}$

W tej algebrze zachodzi:

$$[\forall a \alpha(a)]_\sim = \inf_t [\alpha(t)]_\sim \quad \text{oraz} \quad [\exists a \alpha(a)]_\sim = \sup_t [\alpha(t)]_\sim$$

W algebrze termów \mathcal{A} niech $r^{\mathcal{A}}(t_1, \dots, t_n) = [rt_1 \dots t_n]_\sim$.
Wtedy \mathcal{A} jest \mathcal{L}_Γ -strukturą i $\llbracket \varphi \rrbracket_{id} = [\varphi]_\sim$ dla każdego φ .

Jeśli $\Gamma \models \varphi$, to $\llbracket \varphi \rrbracket_{id} = 1$ i dobrze... ??

Nie bardzo: algebra \mathcal{L}_Γ nie musi być zupełna.

Uzupełnianie algebr Heytinga

Twierdzenie: Niech \mathcal{H} – algebra Heytinga. Istnieje taka zupełna algebra Heytinga \mathcal{G} , że:

- ▶ Algebra \mathcal{H} jest podalgebrą \mathcal{G} ;
- ▶ Jeśli dla $B \subseteq \mathcal{H}$ istnieje $\sup_{\mathcal{H}} B$, to $\sup_{\mathcal{G}} B = \sup_{\mathcal{H}} B$;
- ▶ Jeśli dla $B \subseteq \mathcal{H}$ istnieje $\inf_{\mathcal{H}} B$, to $\inf_{\mathcal{G}} B = \inf_{\mathcal{H}} B$.

Uzupełnianie algebr Heytinga

Twierdzenie: Niech \mathcal{H} – algebra Heytinga. Istnieje taka zupełna algebra Heytinga \mathcal{G} , że:

- ▶ Algebra \mathcal{H} jest podalgebrą \mathcal{G} ;
- ▶ Jeśli dla $B \subseteq \mathcal{H}$ istnieje $\sup_{\mathcal{H}} B$, to $\sup_{\mathcal{G}} B = \sup_{\mathcal{H}} B$;
- ▶ Jeśli dla $B \subseteq \mathcal{H}$ istnieje $\inf_{\mathcal{H}} B$, to $\inf_{\mathcal{G}} B = \inf_{\mathcal{H}} B$.

Jaki z tego pożytek?

Jeśli ζ jest wartościowaniem w \mathcal{H} , to $\llbracket \alpha \rrbracket_{\zeta}^{\mathcal{H}} = \llbracket \alpha \rrbracket_{\zeta}^{\mathcal{G}}$,

dla każdej formuły α . Zatem jeśli $\Gamma \models \alpha$ oraz $\mathcal{H} := \mathcal{L}_{\Gamma}$, to:

- po pierwsze $\mathcal{G}, id \models \Gamma$, bo $\mathcal{L}_{\Gamma}, id \models \Gamma$;
- po drugie, wtedy $\mathcal{G}, id \models \alpha$, więc $\mathcal{L}_{\Gamma}, id \models \alpha$, czyli $\Gamma \vdash \alpha$.

Uzupełnianie algebr Heytinga

Twierdzenie: Niech \mathcal{H} – algebra Heytinga. Istnieje taka zupełna algebra Heytinga \mathcal{G} , że:

- ▶ Algebra \mathcal{H} jest podalgebrą \mathcal{G} ;
- ▶ Jeśli dla $B \subseteq \mathcal{H}$ istnieje $\sup_{\mathcal{H}} B$, to $\sup_{\mathcal{G}} B = \sup_{\mathcal{H}} B$;
- ▶ Jeśli dla $B \subseteq \mathcal{H}$ istnieje $\inf_{\mathcal{H}} B$, to $\inf_{\mathcal{G}} B = \inf_{\mathcal{H}} B$.

Uzupełnianie algebr Heytinga

Twierdzenie: Niech \mathcal{H} – algebra Heytinga. Istnieje taka zupełna algebra Heytinga \mathcal{G} , że:

- ▶ Algebra \mathcal{H} jest podalgebrą \mathcal{G} ;
- ▶ Jeśli dla $B \subseteq \mathcal{H}$ istnieje $\sup_{\mathcal{H}} B$, to $\sup_{\mathcal{G}} B = \sup_{\mathcal{H}} B$;
- ▶ Jeśli dla $B \subseteq \mathcal{H}$ istnieje $\inf_{\mathcal{H}} B$, to $\inf_{\mathcal{G}} B = \inf_{\mathcal{H}} B$.

Dowód: Ideał zupełny w \mathcal{H} to taki zbiór $I \subseteq \mathcal{H}$, że

- ▶ Jeśli $a \sqsubseteq b \in I$, to $a \in I$;
- ▶ Jeśli $A \subseteq I$ ma kres górny w \mathcal{H} , to $\sup_{\mathcal{H}} A \in I$.

Jako \mathcal{G} bierzemy zbiór wszystkich ideałów zupełnych w \mathcal{H} . Włożenie $i : \mathcal{H} \rightarrow \mathcal{G}$ określamy $i(a) = a \downarrow = \{b \mid b \sqsubseteq a\}$.

Kontrprzykłady topologiczne

Kontrprzykłady topologiczne

$$\mathcal{H} = \mathcal{O}(\mathbb{R}), \quad A = \mathbb{N} - \{0\}$$

Kontrprzykłady topologiczne

$$\mathcal{H} = \mathcal{O}(\mathbb{R}), \quad A = \mathbb{N} - \{0\}$$

Przykład 1: $\neg\neg\forall x(r(x) \vee \neg r(x))$

Relacja: $r^A(n) = \mathbb{R} - \{w_n\}$, gdzie $\mathbb{Q} = \{w_i \mid i \in \mathbb{N} - \{0\}\}$.

Kontrprzykłady topologiczne

$$\mathcal{H} = \mathcal{O}(\mathbb{R}), \quad A = \mathbb{N} - \{0\}$$

Przykład 1: $\neg\neg\forall x(r(x) \vee \neg r(x))$

Relacja: $r^A(n) = \mathbb{R} - \{w_n\}$, gdzie $\mathbb{Q} = \{w_i \mid i \in \mathbb{N} - \{0\}\}$.

Przykład 2: $\forall x(p \vee r(x)) \rightarrow p \vee \forall x.r(x)$

Relacje: $r^A(n) = (-\frac{1}{n}, \frac{1}{n})$, $p^A = \mathbb{R} - \{0\}$.