

Homotopiczna teoria typów

Zeimer

13 stycznia 2019

- 1 Wstęp
- 2 Teoria homotopii
- 3 Teoria typów
- 4 Interpretacja homotopiczna
- 5 Zbiory i logika
- 6 Równoważności

Czym jest HoTT?

- Homotopiczna teoria typów (w skrócie HoTT) to połączenie teorii typów i teorii homotopii.
- Jest kolejnym stadium ewolucji teorii typów.
- Jest syntetyczną teorią homotopii, dającą nam łatwy dostęp do skomplikowanych pojęć topologicznych.
- Jest pomysłem na nowe podstawy matematyki, alternatywne wobec teorii zbiorów.
- Jest bardzo potężnym funkcyjnym językiem programowania.

Innowacje HoTT

- Homotopiczna interpretacja teorii typów, mocno wspomagająca wyobraźnię zarówno w rozumowaniu, jak i pozwalająca dogłębnie zrozumieć różne detale teorii typów.
- Aksjomat uniwalencji $(A \simeq B) \simeq (A = B)$, który głosi, że rzeczy mające tę samą strukturę są identyczne. Rozwiązuje to odwieczny problem nieformalnego utożsamiania poprzez nadużycie języka.
- Wyższe typy induktywne, pozwalające w teorii typów:
 - Zdefiniować wiele niemożliwych dotychczas obiektów, np. typy ilorazowe albo prezentacje obiektów algebraicznych.
 - Konstruktywnie rozwiązać wiele problemów, które dotychczas wymagały logiki klasycznej (konstrukcja liczb rzeczywistych Cauchy'ego)
 - Wyrazić klasyczne pojęcia logiczne (dysjunkcja, kwantyfikator egzystencjalny, aksjomat wyboru) z niemożliwą wcześniej w teorii typów precyzją.

Teoria homotopii 1 - homotopia

- Co to jest homotopia?
- Zgodnie z wikipedią, jeżeli f i g są funkcjami ciągłymi z przestrzeni topologicznej X w przestrzeń topologiczną Y , to $H : X \times [0; 1] \rightarrow Y$ jest homotopią, gdy jest funkcją ciągłą spełniającą $H(0, x) = f(x) \wedge H(1, x) = g(x)$.
- Jeżeli nieco pogmeramy w symbolach, to możemy to zapisać tak: $H : [0; 1] \rightarrow (X \rightarrow Y)$ jest homotopią, gdy jest ciągła i spełnia $H(0) = f \wedge H(1) = g$.
- Nie przejmuj się, jeżeli definicja cię nie oświeca. Moim zdaniem władowanie jej do nazwy całej teorii jest głupie.

Teoria homotopii 2 - ścieżka

- Bardziej podstawowym pojęciem jest ścieżka.
- Ścieżka w przestrzeni topologicznej X to funkcja ciągła z $[0; 1]$ w X .
- Łatwo to sobie wyobrazić: odcinek $[0; 1]$ z pewnością jest ścieżką prowadzącą od 0 do 1. Jego obrazem, czyli ścieżką, jest więc pewien ciągły zawijasek, który prowadzi z $f(0)$ do $f(1)$.
- Ostatecznie możemy powiedzieć, że homotopia to ścieżka między funkcjami.
- Teoria homotopii nie jest jednak teorią ścieżek między funkcjami. Jest to raczej po prostu teoria ścieżek.

Teoria homotopii 4 - grupa podstawowa

- Pętla w punkcie x to ścieżka, która zaczyna się i kończy w punkcie x .
- Grupa podstawowa przestrzeni X w punkcie x to grupa, której nośnikiem jest zbiór wszystkich pętli w punkcie x . Działaniem grupowym jest sklejanie pętli (najpierw pójdz pierwszą pętlą, a potem drugą). Odwrotność to pójście pętlą w przeciwnym kierunku. Element neutralny to stanie w miejscu.
- Grupa podstawowa jest fajna, bo jeżeli przestrzenie są izomorficzne, to ich grupy podstawowe też są. Wobec tego jeżeli grupy podstawowe (w dowolnym punkcie) są różne, to przestrzenie też są różne.

Teoria typów 1 - podstawy

- Teorię typów w ujęciu HoTTowym można opisać jako system formalny, który za pomocą reguł (osądów) opisuje byty zwane typami. Kluczową innowacją HoTT jest interpretacja typów i wymyślone na jej podstawie aksjomaty rzucające światło na naturę kosmosu.
- Reguły dzielą się na ciekawe i nieciekawe.
- Niekawne to te, które muszą być, żeby wszystko działało, np. do zamieniania kolejności rzeczy w kontekście.
- Ciekawe to te, które faktycznie opisują typy. Jest ich pięć rodzajów: reguły formacji, wprowadzania, eliminacji, obliczania i unikalności.

Teoria typów 2 - pięć rodzajów reguł

- Reguły formacji mówią, skąd się biorą typy.
- Reguły wprowadzania mówią, jak zrobić elementy danego typu.
- Reguły eliminacji mówią, jak zrobić coś z elementami danego typu.
- Reguły obliczania mówią, jak reguły eliminacji mają się do reguł wprowadzania.
- Reguły unikalności mówią, jak reguły wprowadzania mają się do reguł eliminacji.

Teoria typów 3 - reguły dla funkcji

- Ćwiczenie: nazwij każdą z reguł (tzn. która to reguła formacji, która obliczania etc.)

$$\frac{\Gamma \vdash A : \mathcal{U} \quad \Gamma \vdash B : \mathcal{U}}{\Gamma \vdash A \rightarrow B : \mathcal{U}}$$

$$\frac{\Gamma, x : A \vdash b : B}{\Gamma \vdash \lambda x : A. b : A \rightarrow B}$$

$$\frac{\Gamma \vdash f : A \rightarrow B \quad \Gamma \vdash x : A}{\Gamma \vdash f \ x : B}$$

$$\frac{\Gamma, x : A \vdash b : B \quad \Gamma \vdash a : A}{\Gamma \vdash (\lambda x : A. b) \ a \equiv b[x := a] : B}$$

$$\frac{\Gamma \vdash f : A \rightarrow B}{\Gamma \vdash \lambda x : A. f \ x \equiv f : A \rightarrow B}$$

Teoria typów 4 - ciekawostki o regułach

- Każdy typ musi mieć regułę formacji - inaczej nie byłby typem.
- Jednak nie każdy typ musi mieć pozostałe reguły.
- Typ **0** nie ma reguły wprowadzania, bo jest pusty i nie ma żadnych elementów.
- Uniwersum nie ma reguły eliminacji.
- Typ **0** nie ma także reguły obliczania, co jest oczywiste - nie może jej mieć, skoro nie ma reguły wprowadzania.
- Wiele typów, np. sumy i produkty, nie mają reguły unikalności. W zamian za to mają one zdaniową regułę unikalności, tzn. można udowodnić twierdzenie wyglądające dokładnie jak reguła unikalności.
- Reguła formacji zawsze jest jedna, bo każdy typ można sformować tylko na jeden sposób. Pozostałych reguł może być więcej. Sumy mają 2 reguły wprowadzania, a produkty 2 reguły eliminacji i wobec tego 2 reguły obliczania.

Teoria typów 5 - cztery style definiowania

- Formalnie rzeczy definiujemy za pomocą reguł wprowadzania i eliminacji.
- Przykład: funkcję $\text{swap} : \Pi A B : \mathcal{U}. A \times B \rightarrow B \times A$ możemy zdefiniować jako $\text{swap} \equiv \lambda A : \mathcal{U}. \lambda B : \mathcal{U}. \lambda x : A \times B. (\pi_2 \ x, \pi_1 \ x)$
- Zamiast tego często będziemy jednak definiować poprzez dopasowanie do wzorca, jednocześnie pomijając argumenty, które można wywnioskować z kontekstu: $\text{swap} \ (a, b) \equiv (b, a)$
- Możemy też definiować słownie: niech swap będzie funkcją, która zamienia miejscami elementy pary. Ten sposób będziemy wykorzystywać do dowodzenia twierdzeń.
- Ostatnim stylem jest obrazkowy styl definiowania. Nie jest on używany w książce, ale ja postaram się go wykorzystać podczas tej prezentacji, gdyż dobrze działa na wyobraźnię.

Interpretacja typów 1 - zbiory

- Jak interpretować/rozumieć typy?
- Najprostszy sposób każe nam myśleć, że typy to po prostu zbiory.
- W takim ujęciu typ \mathbb{N} to taki worek, w którym jest $0, 1, 2, \dots$ etc.
- Takie rozumienie było przez długi czas dominujące. Jest ono dość intuicyjne i powszechne przy myśleniu nieformalnym.
- Były też inne dziwne interpretacje, jak (chyba) częściowe relacje równoważności, ale tego to obchodzi.

Interpretacja typów 2 - grupoidy

- Aż tu nagle w pracy z 1995 zatytułowanej “The groupoid interpretation of type theory” panowie Hofmann i Streicher wpadli na pomysł, żeby zinterpretować typy jako grupoidy.
- Upraszczając, grupoid to graf skierowany, w którym:
 - Każdy wierzchołek ma krawędź do samego siebie.
 - Jeżeli jest krawędź z A do B , to jest krawędź z B do A .
 - Jeżeli jest krawędź z A do B i z B do C , to jest krawędź z A do C .
- Jeszcze bardziej upraszczając: grupoid to kolekcja kropek, między którymi są strzałki spełniające pewne warunki.
- Wymyślenie ciągu dalszego tej bajki zajęło dobre 15 lat.

Interpretacja typów 3 - ω -grupoidy

- Aż tu nagle w okolicach roku 2010 Awodey i Warren (a także Voevodsky, van den Berg i Garner) wpadli na pomysł, żeby zinterpretować typy jako ω -grupoidy.
- ω -grupoid to kolekcja kropek, między którymi są strzałki spełniające warunki jak dla grupoidu. Co więcej, między strzałkami też mogą być strzałki spełniające te warunki. Są też strzałki między strzałkami między strzałkami i tak dalej aż do nieskończoności.
- Jeżeli pomyślimy o naszych “strzałkach” jak o ścieżkach w przestrzeni, to dostajemy homotopiczną interpretację teorii typów. W zasadzie to każdy ω -grupoid jest reprezentacją jakiejś przestrzeni topologicznej.

1.12 Ścieżki 1 - reguły

$$\frac{\Gamma \vdash A : \mathcal{U}_i \quad \Gamma \vdash a : A \quad \Gamma \vdash b : A}{\Gamma \vdash a =_A b : \mathcal{U}_i} =\text{-FORM} \qquad \frac{\Gamma \vdash A : \mathcal{U}_i \quad \Gamma \vdash a : A}{\Gamma \vdash \text{refl}_a : a =_A a} =\text{-INTRO}$$

$$\frac{\Gamma, x:A, y:A, p:x=_A y \vdash C:\mathcal{U}_i \quad \Gamma, z:A \vdash c:C[z, z, \text{refl}_z/x, y, p] \quad \Gamma \vdash a:A \quad \Gamma \vdash b:A \quad \Gamma \vdash p':a=_A b}{\Gamma \vdash \text{ind}_{=_A}(x.y.p.C, z.c, a, b, p') : C[a, b, p'/x, y, p]} =\text{-ELIM}$$

$$\frac{\Gamma, x:A, y:A, p:x=_A y \vdash C : \mathcal{U}_i \quad \Gamma, z:A \vdash c : C[z, z, \text{refl}_z/x, y, p] \quad \Gamma \vdash a : A}{\Gamma \vdash \text{ind}_{=_A}(x.y.p.C, z.c, a, a, \text{refl}_a) \equiv c[a/z] : C[a, a, \text{refl}_a/x, y, p]} =\text{-COMP}$$

In $\text{ind}_{=A}$, x , y , and p are bound in C , and z is bound in c .

Powyższe reguły opisują rodzinę typów, która zazwyczaj nazywana bywa typem identycznościowym (ang. identity type), ale zgodnie z interpretacją homotopiczną będę go nazywał typem ścieżek.

1.12 Ścieżki 2 - interpretacja reguł

- Reguła formacji: jeżeli mamy typ A i dwa jego elementy a, b , to możemy sformować typ $a =_A b$. Jest to typ, którego elementami są ścieżki z a do b . Jeżeli mamy element tego typu, to a i b są równe.
- Reguła wprowadzania: każda rzecz jest równa sama sobie. Ścieżka poświadczająca ten fakt nazywa się refl. Jest to skrót od ang. reflexivity, czyli zwrotność.
- Reguła eliminacji: C jest tutaj rodziną typów zależącą od ścieżki $p : x = y$. Reguła głosi, że żeby zdefiniować element $C(x, y, p)$ wystarczy mieć element $C(x, x, \text{refl}_x)$.
- Reguła obliczania: chodzi o to, że jeżeli wyeliminujemy element $C(z, z, \text{refl}_z)$, to dostaniemy go spowrotem, tylko po odpowiednim podstawieniu.

1.12 Ścieżki 3 - indukcja po ścieżkach

- Reguła eliminacji dla ścieżek nosi nazwę indukcji po ścieżkach (ang. path induction).
- Zaprezentowany powyżej wariant precyzyjniej nazywa się unbased path induction. Polega na zastąpieniu dwóch obiektów a, b i ścieżki p przez generyczny obiekt z i ścieżkę refl_z .
- Inny wariant nosi nazwę based path induction. Polega on na zastąpieniu obiektu b przez obiekt a oraz ścieżki $p : a = b$ przez ścieżkę refl_a .
- Oba warianty są równoważne. Dowód: HoTT Book, podrozdział 1.12.2.

1.12 Ścieżki 4 - interpretacja indukcji po ścieżkach

- Tak jak indukcję na liczbach naturalnych możemy zobrazować za pomocą domina, tak indukcję po ścieżkach możemy wyobrażać sobie jako ściągnięcie/zwinięcie ścieżki $p : a = b$ do ścieżki trywialnej.
- W wariancie unbased oba końce ścieżki p są wolne. Wybieramy jakiś punkt z na ścieżce i ciągniemy oba końce w jego kierunku. Ostatecznie dostajemy ścieżkę refl_z .
- W wariancie based lewy koniec ścieżki p jest sztywny, a prawy jest wolny. Chwytny więc prawy koniec b i ciągniemy go po ścieżce w kierunku lewego końca a . Ostatecznie dostajemy ścieżkę refl_a .
- Zauważmy, że jeżeli oba końce ścieżki są sztywne, to nie możemy robić indukcji - spróbuj pociągnąć linę okręconą wokół latarni. O tym, czy koniec jest sztywny czy wolny, decyduje to, czy jest skwantyfikowany uniwersalnie czy nie.

1.12 Ścieżki 5 - wątpliwości i ciekawostki

- Reguła eliminacji dla typu `bool` intuicyjnie mówi, że jedynymi elementami typu `bool` są `true` oraz `false`.
- Czy więc indukcja po ścieżkach mówi, że jedyną ścieżką jest `refl`?
- Zanim odpowiemy, garść ciekawostek.
- Indukcja po ścieżkach nie jest HoTTową innowacją. Jedynie nazwa jest nowa. W teorii typów bywa często nazywana J .
- Zdanie mówiące, że każda ścieżka jest trywialna, nazywa się “Aksjomat K ”.
- Związek z facetami w czerni jest przypadkowy.
- Inne zdanie, mówiące że jest tylko jedna ścieżka, nazywa się w ang. UIP, co jest skrótem od “Uniqueness of Identity Proofs”.
- To właśnie badanie nad tego typu zagadnieniami doprowadziły do homotopicznej interpretacji teorii typów.

1.12 Ścieżki 6 - rozwiązanie wątpliwości

- Indukcja po ścieżkach nie głosi, że jest tylko jedna ścieżka.
- Formalna różnica jest taka, że typ `bool` jest generowany induktywnie, podczas gdy w przypadku ścieżek, które są rodziną typów, to cała rodzina jest generowana induktywnie, a nie pojedynczy typ $x = y$.
- Parafrazując, nie można w tym przypadku rozważać samych ścieżek w oderwaniu od ich końców.
- Nie możemy zatem udowodnić, że każda ścieżka $p : x = x$ jest trywialna.
- Ale możemy udowodnić, że każda ścieżka razem z jej końcami jest trywialna: zachodzi $(x, y, p) = (x, x, \text{refl}_x)$, gdzie równość jest w typie $\Sigma x \ y : A, x = y$. Odpowiada to indukcji po ścieżkach w wersji `unbased`.
- Podobnie dla ustalonego $a : A$ możemy pokazać, że $(x, p) = (a, \text{refl}_a)$ w typie $\Sigma x : A, a = x$. Odpowiada to indukcji po ścieżkach w wersji `based`.

1.12 Ścieżki 7 - skąd się biorą ścieżki

- Póki co wiemy, że jest ścieżka trywialna.
- Wiemy też, że indukcja po ścieżkach nie wyklucza istnienia innych ścieżek.
- Rodzi się jednak pytanie: skąd się biorą ścieżki?
- Cztery główne źródła ścieżek, które zobaczymy w przyszłości, to:
 - Aksjomat ekstensjonalności dla funkcji - ścieżki powstają z homotopii.
 - Aksjomat uniwalencji - ścieżki powstają z równoważności.
 - Wyższe typy induktywne - możemy wrzucić do typu dowolne ścieżki.
 - Struktura ω -grupoidu - powyższe trzy rodzaje (potencjalnie) nietrywialnych ścieżek mogą ze sobą oddziaływać za pośrednictwem struktury ω -grupoidu, tworząc jeszcze więcej nietrywialnych ścieżek.

2.1 Typy to wyższe grupoidy 1 - definicje

Definition (Lemat 2.1.1 - ścieżka odwrotna)

$$(-)^{-1} : \prod A : \mathcal{U}. \prod x \ y : A. x = y \rightarrow y = x$$

$$\text{refl}_x^{-1} :\equiv \text{refl}_x$$

Definition (Lemat 2.1.2 - sklejanie ścieżek)

$$\cdot : \prod A : \mathcal{U}. \prod x \ y \ z : A. x = y \rightarrow y = z \rightarrow x = z$$

$$\text{refl}_x \cdot \text{refl}_x :\equiv \text{refl}_x$$

Equality	Homotopy	∞ -Groupoid
reflexivity	constant path	identity morphism
symmetry	inversion of paths	inverse morphism
transitivity	concatenation of paths	composition of morphisms

2.1 Typy to wyższe grupoidy 2 - właściwości

Theorem (Lemat 2.1.4 - właściwości operacji na ścieżkach)

Niech $A : \mathcal{U}$ będzie typem, $a, b, c, d : A$ punktami, zaś $p : a = b, q : b = c, r : c = d$ ścieżkami. Wtedy:

- $\text{refl}_x \cdot p = p$
- $p \cdot \text{refl}_y = p$
- $p \cdot p^{-1} = \text{refl}_x$
- $p^{-1} \cdot p = \text{refl}_y$
- $p \cdot (q \cdot r) = (p \cdot q) \cdot r$

Ćwiczenie: udowodnij.

2.1 Typy to wyższe grupoidy 3 - prostujcie ścieżki Pana

- Zauważmy, że cała wyższogrupoidowa struktura typów wynika wprost z indukcji po ścieżkach.
- Zauważmy też, że powyższe właściwości operacji na ścieżkach są wyrażone za pomocą ścieżek między ścieżkami.
- Tak naprawdę, to te właściwości są operacjami, które biorą na wejściu ścieżki i zwracają ścieżki między ścieżkami.
- Wobec tego można domniemywać, że te właściwości same spełniają jakieś właściwości, które są wyrażane przez ścieżki jeszcze wyższego rzędu...
- ... i tak do nieskończoności.
- Katolicy bywają zachęcani do tego, żeby “prostować ścieżki Pana”. Atoli zachęcam ja was: prostujcie ω -grupoid Pana (oczywiście za pomocą indukcji po ścieżkach).

2.2 Funkcje to funktory 1 - definicje

Definition (Lemat 2.2.1 - aplikacja funkcji do ścieżki)

$$\begin{aligned} \text{ap} : \prod A B : \mathcal{U}. \prod f : A \rightarrow B. x =_A y \rightarrow f(x) =_B f(y) \\ \text{ap}_f(\text{refl}_x) :\equiv \text{refl}_{f(x)} \end{aligned}$$

Klasycznie powyższą definicję moglibyśmy odczytać jako twierdzenie mówiące, że wszystkie funkcje zachowują równość.

W interpretacji homotopicznej twierdzenie to (które jednocześnie definiuje pewną funkcję) głosi, że funkcje zachowują ścieżki.

Zauważ też, że również tutaj nie wyczerpujemy tematu. Funkcje zachowują nie tylko ścieżki jednowymiarowe, ale także np. pięciowymiarowe pętle. Podobnie zachowują się funkcje zależne, o których tutaj milczymy.

2.2 Funkcje to funktory 2 - właściwości

Lemma 2.2.2. *For functions $f : A \rightarrow B$ and $g : B \rightarrow C$ and paths $p : x =_A y$ and $q : y =_A z$, we have:*

$$(i) \operatorname{ap}_f(p \bullet q) = \operatorname{ap}_f(p) \bullet \operatorname{ap}_f(q).$$

$$(ii) \operatorname{ap}_f(p^{-1}) = \operatorname{ap}_f(p)^{-1}.$$

$$(iii) \operatorname{ap}_g(\operatorname{ap}_f(p)) = \operatorname{ap}_{g \circ f}(p).$$

$$(iv) \operatorname{ap}_{\operatorname{id}_A}(p) = p.$$

Proof. Left to the reader.

□

Ćwiczenie: udowodnij.

2.3 Mu synku mu, czyli transport 1 - definicja

Definition (Lemat 2.3.1 - transport)

$\text{transport} : \prod A : \mathcal{U}. \prod B : A \rightarrow \mathcal{U}. \prod x \ y : A. x = y \rightarrow P(x) \rightarrow P(y)$
 $\text{transport}(\text{refl}_x) \equiv \text{id}_{P(x)}$

Notacja: $p_* \equiv \text{transport}(p)$

Powyższe klasycznie można odczytać jako jedną stronę równoważności, której Leibniz użył do zdefiniowania równości: “dwie rzeczy są równe wtedy i tylko wtedy, gdy mają takie same właściwości”.

Homotopicznie sprawa jest nieco ciekawsza: jeżeli mamy ścieżkę $p : x =_A y$ i jakiś obiekt typu $P(x)$, to możemy go przenieść (czyli właśnie przetransportować) do typu $P(y)$ wzdłuż ścieżki p .

2.3 Mu synku mu, czyli transport 2 - właściwości

Lemma 2.3.9. *Given $P : A \rightarrow \mathcal{U}$ with $p : x =_A y$ and $q : y =_A z$ while $u : P(x)$, we have*

$$q_*(p_*(u)) = (p \bullet q)_*(u).$$

Lemma 2.3.10. *For a function $f : A \rightarrow B$ and a type family $P : B \rightarrow \mathcal{U}$, and any $p : x =_A y$ and $u : P(f(x))$, we have*

$$\text{transport}^{P \circ f}(p, u) = \text{transport}^P(\text{ap}_f(p), u).$$

Lemma 2.3.11. *For $P, Q : A \rightarrow \mathcal{U}$ and a family of functions $f : \prod_{(x:A)} P(x) \rightarrow Q(x)$, and any $p : x =_A y$ and $u : P(x)$, we have*

$$\text{transport}^Q(p, f_x(u)) = f_y(\text{transport}^P(p, u)).$$

Ćwiczenie: udowodnij.

2.3 Aplikacja funkcji zależnych do ścieżek

Definition (Lemat 2.3.4)

$$\begin{aligned} \text{apd} : \Pi A : \mathcal{U}. \Pi P : A \rightarrow \mathcal{U}. \Pi f : (\Pi x : A. P(x)). \Pi p : x = \\ y. p_*(f(x)) = f(y) \\ \text{apd}_f(\text{refl}_x) \equiv \text{refl}_{f(x)} \end{aligned}$$

Aplikacja funkcji zależnych do ścieżek jest analogiczna do aplikacji funkcji niezależnych do ścieżek, ale jest mały twist - musimy użyć transportu, bo wyniki funkcji dla x i y żyją w różnych typach.

Najłatwiej jest o tym myśleć w ten sposób: funkcje zależne są równoważne funkcjom niezależnym między odpowiednimi Σ -typami.

2.4 Homotopie

Definition 2.4.1. Let $f, g : \prod_{(x:A)} P(x)$ be two sections of a type family $P : A \rightarrow \mathcal{U}$. A **homotopy** from f to g is a dependent function of type

$$(f \sim g) \equiv \prod_{x:A} (f(x) = g(x)).$$

Note that a homotopy is not the same as an identification ($f = g$). However, in §2.9 we will introduce an axiom making homotopies and identifications “equivalent”.

The following proofs are left to the reader.

Lemma 2.4.2. *Homotopy is an equivalence relation on each dependent function type $\prod_{(x:A)} P(x)$. That is, we have elements of the types*

$$\begin{aligned} & \prod_{f:\prod_{(x:A)} P(x)} (f \sim f) \\ & \prod_{f,g:\prod_{(x:A)} P(x)} (f \sim g) \rightarrow (g \sim f) \\ & \prod_{f,g,h:\prod_{(x:A)} P(x)} (f \sim g) \rightarrow (g \sim h) \rightarrow (f \sim h). \end{aligned}$$

Ćwiczenie: udowodnij.

2.4 Równoważności 1 - koncert życzeń

- Chcielibyśmy mieć pojęcie mówiące, że dwa typy są równoważne. Oczywiście mamy równość (ścieżki), ale chcemy czegoś fajniejszego.
- Prymitywny pomysł: bijekcja, czyli funkcja będącą surjekcją i injekcją (względnie: dla każdego y istnieje dokładnie jeden $x...$).
- Bardziej wzniosły pomysł: izomorfizm, czyli funkcja mająca obustronną odwrotność.

2.4 Równoważności 2 - kwaziodwrotność

Definition 2.4.6. For a function $f : A \rightarrow B$, a **quasi-inverse** of f is a triple (g, α, β) consisting of a function $g : B \rightarrow A$ and homotopies $\alpha : f \circ g \sim \text{id}_B$ and $\beta : g \circ f \sim \text{id}_A$.

Thus, (2.4.5) is *the type of quasi-inverses of f* ; we may denote it by $\text{qinv}(f)$.

Example 2.4.7. The identity function $\text{id}_A : A \rightarrow A$ has a quasi-inverse given by id_A itself, together with homotopies defined by $\alpha(y) \equiv \text{refl}_y$ and $\beta(x) \equiv \text{refl}_x$.

Example 2.4.8. For any $p : x =_A y$ and $z : A$, the functions

$$\begin{aligned} (p \cdot -) : (y =_A z) &\rightarrow (x =_A z) & \text{and} \\ (- \cdot p) : (z =_A x) &\rightarrow (z =_A y) \end{aligned}$$

have quasi-inverses given by $(p^{-1} \cdot -)$ and $(- \cdot p^{-1})$, respectively; see Exercise 2.6.

Example 2.4.9. For any $p : x =_A y$ and $P : A \rightarrow \mathcal{U}$, the function

$$\text{transport}^P(p, -) : P(x) \rightarrow P(y)$$

has a quasi-inverse given by $\text{transport}^P(p^{-1}, -)$; this follows from Lemma 2.3.9.

2.4 Równoważności 3 - pobożne życzenia

- Dlaczego nazwaliśmy pojęcie odpowiadające izomorfizmowi kwaziodwrotnością?
- Jak się okaże, jest ono wadliwe.
- Chcielibyśmy, żeby definicja równoważności isequiv spełniała następujące warunki:
 - $\text{qinv}(f) \rightarrow \text{isequiv}(f)$
 - $\text{isequiv}(f) \rightarrow \text{qinv}(f)$
 - $\prod e_1 \ e_2 : \text{isequiv}(f), e_1 = e_2$
- Parafrazując: isequiv to niemal to samo co qinv , ale każda funkcja może być równoważnością na co najwyżej jeden sposób.

2.4 Równoważności 4 - co poszło nie tak

$qinv$ powyższych warunków nie spełnia. Jeżeli narysujemy odpowiednio plastyczny rysunek, to ujrzymy uzasadnienie dla poniższego twierdzenia:

Theorem (Twierdzenie 4.1.1 - $qinv$ to pętla)

$$\prod A \ B : \mathcal{U}. \prod f : A \rightarrow B. qinv(f) \rightarrow qinv(f) \simeq \prod x : A. x = x$$

2.4 Równoważności 5 - do wyboru do koloru

Niech $A, B : \mathcal{U}$ będą typami, a $f : A \rightarrow B$ funkcją.

Definition (Równoważność 1)

$$\text{isequiv}(f) := \left(\sum_{g:B \rightarrow A} f \circ g \sim \text{id}_B \right) \times \left(\sum_{h:B \rightarrow A} h \circ f \sim \text{id}_A \right)$$

Definition (Równoważność 2)

$$\text{isequiv}(f) := \sum_{g:B \rightarrow A} \sum_{\eta:g \circ f \sim \text{id}_A} \sum_{\epsilon:f \circ g \sim \text{id}_B} \prod_{x:A} \text{ap}_f(\eta(x)) = \epsilon(\text{ap}_f(x))$$

Definition (Równoważność 3)

$$\text{isequiv}(f) := \prod_{y:B} \text{isContr} \left(\sum_{x:A} f(x) = y \right)$$

2.4 Równoważności 6 - koniec

Definition (Równoważność 4)

$$\text{isequiv}(f) :\equiv ||\text{qinv}(f)||$$

2.5-2.8 Wyższogrupoidowa struktura typów

- todo

2.9 Ekstensjonalność

- todo

2.10 Uniwalencja

- todo

Innowacje HoTT w przykładach - uniwalencja 1

- Rozważmy dwa poniższe typy (tak naprawdę powinniśmy też podać reguły eliminacji i obliczania, ale nie są one istotne dla przykładu).
- Niech $\mathbb{N} := 0 \mid S \mathbb{N}$ i niech $\mathbb{N}' := 0' \mid S' \mathbb{N}'$
- Rodzi się pytanie: czy \mathbb{N} i \mathbb{N}' to to samo, czy coś innego?
- Odpowiedź klasyczna: istnieje oczywisty izomorfizm $\mathbb{N} \cong \mathbb{N}'$. Na mocy nadużycia języka będziemy utożsamiać \mathbb{N} i \mathbb{N}' , tzn. traktować je tak, jakby $\mathbb{N} = \mathbb{N}'$ mimo, że formalnie tak nie jest.
- Odpowiedź HoTTowa: istnieje oczywista równoważność $e : \mathbb{N} \simeq \mathbb{N}'$. Wobec tego na mocy aksjomatu uniwalencji mamy ścieżkę $ua(e) : \mathbb{N} = \mathbb{N}'$.

Innowacje HoTT w przykładach - uniwalencja 2

- Aksjomat uniwalencji nie tylko usuwa nieprzyjemny filozoficzny smrodek, ale daje nam też nowe sposoby rozumowania.
- Przykład: każda równoważność jest monomorfizmem.
- Dowód klasyczny: każda równoważność ma odwrotność. Użyj jej.
- Dowód HoTTowy: każda równoważność pochodzi od jakiejś ścieżki. Na mocy indukcji po ścieżkach możemy założyć, że ścieżka ta jest trywialna, a zatem nasza równoważność jest identycznością.

2.14 Przykład: równość struktur

- todo

Hierarchia n-typów

- Hierarchia n-typów: unit, funkcje sortujące, pusty, zbiory, grupoidy, okrąg, uniwersum.

Metoda encode-decode dla liczb naturalnych

- todo - przenieść to w miejsce, gdzie już wiadomo, co to zbiory

3.11 Kontraktowalność

- todo

3.3 Zdania

- todo

3.1 Zbiory

- todo

3.7 Trunkacja

- todo

3.4 Logika klasyczna i intuicjonistyczna

- todo

Innowacje HoTT w przykładach - logika 1

- Znaczenie klasycznej dysjunkcji jest inne, niż konstruktywnej. Klasycznie $P \vee Q$ znaczy, że zachodzi P lub Q lub oba na raz, ale nie wiemy, które. Konstruktywnie $P \vee Q$ znaczy, że zachodzi P lub Q i wiemy, z którym przypadkiem mamy do czynienia

3.8 Aksjomat Wyboru



Innowacje HoTT w przykładach - logika 2

Theorem (Aksjomat wyboru)

$$\begin{aligned} & \Pi(A : \mathcal{U})(B : A \rightarrow \mathcal{U})(R : \Pi x : A, B\ x \rightarrow \mathcal{U}), \\ & (\Pi x : A, \sum y : B\ x, R\ x\ y) \rightarrow \\ & \sum f : (\Pi x : A, B\ x), \Pi x : A, R\ x\ (f\ x) \end{aligned}$$

Dowód.

Na tablicy.



Powyższe twierdzenie jest problematyczne, gdyż wygląda jak aksjomat wyboru, ale nie ma tutaj żadnego wybierania.

3.9 Zasada unikalnego wyboru



wut



4.6 Surjekcje i injekcje



Wielkie odkrycie filozoficzne - iniekcja to surjekcja



