Homotopiczna teoria typów

Zeimer

14 stycznia 2019

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 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.
- Nieciekawe 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.)

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

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

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

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

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

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 $\mathbf{0}$ nie ma reguły wprowadzania, bo jest pusty i nie ma żadnych elementów.
- Uniwersum nie ma reguły eliminacji.
- Typ $\mathbf{0}$ 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 swap : $\Pi A\ B : \mathcal{U}. A \times B \to B \times A$ możemy zdefiniować jako swap $:\equiv \lambda A : \mathcal{U}. \lambda B : \mathcal{U}. \lambda x : A \times B. (\mathsf{pr}_2(x), \mathsf{pr}_1(x))$
- Zamiast tego często będziemy jednak definiować poprzez dopasowanie do wzorca, jednocześnie pomijając argumenty, które można wywnioskować z kontekstu: 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.

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 $H : X \times [0; 1] \to Y$ 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: $H : [0; 1] \to (X \to Y)$ 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 3 - topologia (algebraiczna)

- Po co to wszystko?
- Topologia jest całkiem użyteczna. Ostatnio popularna robi się topologiczna analiza danych. Zamiast prymitywnie przypasowywać do danych proste (regresja liniowa), ludzie próbują lepiej opisywać kształt danych. Topologia bada kształty, więc pasuje jak ulał.
- Chcemy więc wiedzieć więcej o topologii, np. czy dwie przestrzenie są takie same czy inne. Tutaj wkracza topologia algebraiczna, czyli dziedzina badająca przestrzenie topologiczne za pomocą metod algebraicznych.

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 pójdź 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 homotopii 5 - okrąg i liczby całkowite

- Okrąg to taka przestrzeń topologiczna, że... wyobraź sobie, pewnie kiedyś widziałeś okrąg.
- Grupa podstawowa okręgu w dowolnym punkcie jest izomorficzna z grupą liczb całkowitych z dodawaniem.
- Stanie w miejscu reprezentuje 0.
- n okrążeń zgodnie z ruchem wskazówek zegara reprezentuje liczbę n.
- n okrążeń przeciwnie do ruchu wskazówek zegara reprezentuje liczbę $-n$.

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 $0, 1, 2, \ldots$ 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 kogo 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 wierzhołek 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

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

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

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

In $\mathsf{ind}_{=_A}$, 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 $C(x, x, \mathrm{refl}_x)$.
- Reguła obliczania: chodzi o to, że jeżeli wyeliminujemy element $C(z, z, \mathrm{refl}_z)$, 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. Chwytamy 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 - rozwianie 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 $(x, y, p) = (x, x, \mathrm{refl}_x)$, 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 $(x, p) = (a, \mathrm{refl}_a)$ 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.

Operacje na ścieżkach 1 - definicje

Definition (Lemat 2.1.1 - ścieżka odwrotna)

$(-)^{-1} : \Pi A : \mathcal{U}.\Pi x\ y : A.x = y \to y = x$
$\mathsf{refl}_x^{-1} :\equiv \mathsf{refl}_x$

Definition (Lemat 2.1.2 - sklejanie ścieżek)

$\cdot : \Pi A : \mathcal{U}.\Pi x\ y\ z : A.x = y \to y = z \to x = z$
$\mathsf{refl}_x \cdot \mathsf{refl}_x :\equiv \mathsf{refl}_x$

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

Operacje na ścieżkach 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:

- $refl_x \cdot p = p$
- $p \cdot refl_y = p$
- $p \cdot p^{-1} = refl_x$
- $p^{-1} \cdot p = refl_y$
- $p \cdot (q \cdot r) = (p \cdot q) \cdot r$

Ćwiczenie: udowodnij.

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).

Aplikacja funkcji do ścieżki 1 - definicja

Definition (Lemat 2.2.1 - aplikacja funkcji do ścieżki)

$\mathrm{ap} : \Pi A\ B : \mathcal{U}.\Pi f : A \to B.x =_A y \to f(x) =_B f(y)$

$\mathrm{ap}_f(\mathrm{refl}_x) :\equiv \mathrm{refl}_{f(x)}$

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.

Aplikacja funkcji do ścieżki 2 - właściwości

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

(i) $\mathsf{ap}_f(p \cdot q) = \mathsf{ap}_f(p) \cdot \mathsf{ap}_f(q)$.

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

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

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

Proof. Left to the reader. □

Ćwiczenie: udowodnij.

Transport 1 - definicja

Definition (Lemat 2.3.1 - transport)

$\mathrm{transport} : \Pi A : \mathcal{U}.\Pi B : A \to \mathcal{U}.\Pi x\ y : A.x = y \to P(x) \to P(y)$
$\mathrm{transport}(\mathrm{refl}_x) :\equiv \mathrm{id}_{P(x)}$

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

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.

Transport 2 - właściwości

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

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

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

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

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

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

Ćwiczenie: udowodnij.

Aplikacja funkcji zależnej do ścieżki

Definition (Lemat 2.3.4)

$\mathsf{apd} : \Pi A : \mathcal{U}.\Pi P : A \to \mathcal{U}.\Pi f : (\Pi x : A.P(x)).\Pi p : x = y.p_*(f(x)) = f(y)$

$\mathsf{apd}_f(\mathsf{refl}_x) :\equiv \mathsf{refl}_{f(x)}$

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.

Homotopie 1 - definicje i właściwości

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

$$\prod_{f:\prod_{(x:A)} P(x)} (f \sim f)$$

$$\prod_{f,g:\prod_{(x:A)} P(x)} (f \sim g) \to (g \sim f)$$

$$\prod_{f,g,h:\prod_{(x:A)} P(x)} (f \sim g) \to (g \sim h) \to (f \sim h).$$

Ćwiczenie: udowodnij.

Homotopie 2 - intuicja

- Klasycznie (czyli płasko) $f \sim g$ możemy czytać jako "f i g są ekstensjonalnie równe".
- HoTTowym odpowiednikiem ekstensjonalnej równości jest homotopia, która intuicyjnie znaczy, że dla każdego elementu dziedziny wyniki funkcji f i g są połączone ścieżką w przeciwdziedzinie.
- To pojęcie homotopii różni się jednak od tego zaprezentowanego w pierwszych slajdach, gdyż homotopie i ścieżki między funkcjami nie są a priori tym samym - nie da się tego pokazać w podstawowej wersji naszej teorii.
- Dlatego w bliskiej przyszłości będziemy dążyć do tego, żeby załatać tę sytuację za pomocą aksjomatu ekstensjonalności.

Równoważności 1 - pomysły

Homotopicznie zinterpretowawszy typy, zdążajmy teraz ku aksjomatowi uniwalencji. Żeby go sformułować, potrzebne nam będzie pojęcie równoważności typów. Przyjrzyjmy się zatem tradycyjnym pojęciom o podobnym charakterze:

- Bijekcja - funkcja będąca surjekcją i injekcją.
- Bijekcja v2 - dla każdego elementu przeciwdziedziny istnieje dokładnie jeden element dziedziny.
- Izomorfizm - morfizm mający obustronną odwrotność.

Równoważności 2 - kwaziodwrotność

Definition

$$\mathrm{qinv} : \Pi A\ B : \mathcal{U}.(A \to B) \to \mathcal{U}$$
$$\mathrm{qinv}(f) :\equiv \sum_{g:B\to A} g \circ f \sim \mathrm{id}_A \times f \circ g \sim \mathrm{id}_B$$

Funkcję mającą odwrotność (wraz z dowodami na to, że faktycznie jest to odwrotność) będziemy nazywać kwaziodwrotnością.

Ćwiczenio-przykład. Pokaż, że kwaziodwrotnościami są następujące funkcje:

- $\mathrm{id}_A : A \to A$
- $(p \cdot -) : y = z \to x = z$
- $(- \cdot q) : x = y \to x = z$
- $\mathrm{transport}(p, -) : P(x) \to P(y)$

Równoważności 3 - co poszło nie tak

Dlaczego nazwaliśmy funkcje mające odwrotność kwaziodwrotnościami, a nie izomorfizmami? Okazuje się, że są one wadliwe. Jeżeli narysujemy odpowiednio plastyczny rysunek, to ujrzymy uzasadnienie dla poniższego twierdzenia:

Theorem (Twierdzenie 4.1.1 - *qinv* to pętle)

$$\Pi A\ B : \mathcal{U}.\Pi f : A \to B.qinv(f) \to (qinv(f) = \Pi x : A.x = x)$$

Twierdzenie to głosi, że jeżeli funkcja f jest kwaziodwrotnością, to typ qinv(f) jest równy typowi funkcji zależnych, które każdemu punktowi przyporządkowują jakąś pętlę. Dlaczego twierdzenie to nas niepokoi? Jak już wiemy, pętli w danym punkcie może być wiele. Mimo, że dana funkcja może mieć tylko jedną odwrotność, to dowodów tego faktu (czyli odpowiednich par ścieżek) może być wiele. Wobec tego funkcja może być kwaziodwrotnością na wiele sposobów.

Równoważności 4 - pobożne życzenia

- Chcielibyśmy, żeby definicja równoważności isequiv spełniała następujące warunki:
  - $\mathrm{qinv}(f) \to \mathrm{isequiv}(f)$
  - $\mathrm{isequiv}(f) \to \mathrm{qinv}(f)$
  - $\Pi e_1\ e_2 : isequiv(f), e_1 = e_2$
- Parafrazując: isequiv to niemal to samo co qinv, ale każda funkcja może być równoważnością na co najwyżej jeden sposób.

Równoważności 5 - definicje

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

Definition (Równoważność 1)

$$\mathrm{isequiv}(f) :\equiv \left(\sum_{g:B\to A} f \circ g \sim \mathrm{id}_B\right) \times \left(\sum_{h:B\to A} h \circ f \sim \mathrm{id}_A\right)$$

Definition (Równoważność 2)

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

Równoważności 6 - wybór

Żeby uzyskać definicję isequiv, możemy "ulepszyć" definicję qinv. Możemy to zrobić na dwa sposoby:

- Rozdzielamy odwrotność na dwie osobne. Wtedy każda z nich ma swój osobny dowód, że jest odwrotnością i gitara gra.
- Dodajemy dodatkową ścieżkę, która zapewnia, że ścieżki dowodzące odwrotności dobrze się ze sobą zachowują.

Druga definicja zdaje się być korzystniejsza w użyciu, bo łatwiej wyjąć z niej odwrotność. Dużo łatwiej jest też dostrzec, że spełnia ona dwa pierwsze pożądane przez nas warunki. Tego, że spełnia trzeci, nie będziemy dowodzić, bo to nudne.

Równoważności 7 - więcej definicji

Definition (Równoważność 3)

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

Definition (Równoważność 4)

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

Możliwe są jeszcze dwie inne definicje równoważności, których jednak nie wybierzemy, gdyż zawierają nieznane nam pojęcia.

Równoważności 8 - interpretacja reszty definciji

- Definicja nr 4 to coś w stylu: qinv nie działa, więc ulepszymy go czarami tak, żeby jednak działał (więcej dowiemy się później).
- Definicja nr 3 odpowiada naszej definicji bijekcji v2 (dla każdego elementu przeciwdziedziny istnieje dokładnie jeden element dziedziny), ale musimy zapisać to bardziej homotopicznie.
- Jest tak dlatego, że interesują nas nie tylko punkty, ale też ścieżki na każdym możliwym poziomie.
- Definicję tę można czytać tak: dla każdego punktu przeciwdziedziny istnieje tylko jeden punkt dziedziny i jedna pętla na nim, i jedna pętla na tej pętli, i jedna pętli na tej pętli i tak dalej na każdym poziomie.
- Prościej: przeciwobraz każdego punktu przeciwdziedziny jest równoważny typowi $\mathbf{1}$.

Równoważności 9 - refleksja

- Oczywiście wszystkie 4 definicje są równoważne.
- Skąd jednak biorą się problemy? Klasyczna definicja izomorfizmu okazała się za słaba, zaś definicję bijekcji v2 również trzeba było odpowiednio stuningować.
- Powód tego jest prosty: klasyczne definicje pochodzą ze świata teorii zbiorów, w którym to świecie mamy tylko worki z kropkami. W świecie, w którym kropki mogą być połączone, definicje trzeba unowocześnić.
- Żeby lepiej zrozumieć różnicę, dokonajmy pewnego wielkiego odkrycia.

Wielkie odkrycie 1 - injekcja to surjekcja

- W ramach ciekawostki dokonajmy pewnego wesołego odkrycia: injekcja to surjekcja.
- Klasycznie f jest surjekcją, gdy $\forall y \in B. \exists x \in A. f(x) = y$
- Klasycznie f jest injekcją, gdy $\forall x\ y \in A. f(x) = f(y) \implies x = y$
- Przeformułujmy tę definicję na taką bardziej homo, wciskając tam więcej ścieżek: f jest injekcją, gdy $\Pi x\ y : A. \Pi q : f(x) = f(y). \Sigma p : x = y. \mathrm{ap}_f(p) = q$
- Klasyczną surjekcję możemy rozumieć jako 0-surjekcję, tzn. surjekcję na punktach, zaś klasyczną injekcję jako 1-surjekcję, tzn. surjekcję na ścieżkach między punktami.
- Jak więc widać, klasyczna bijekcja to surjekcja na poziomach 0 i 1. A co z wyższymi?
- Próba pogłębienia tej obserwacji prowadzi do ciekawego twierdzenia.

Wielkie odkrycie 2 - wesoła charakteryzacja równoważności

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

Definition (Surjekcja - nieco inaczej niż w książce)

f jest surjekcją, gdy $\Pi y : B.\Sigma x : A.f(x) = y$

Definition (Zanurzenie)

f jest zanurzeniem, gdy dla każdego $x, y : A$ funkcja $\mathrm{ap}_f : x = y \to f(x) = f(y)$ jest równoważnością.

Theorem (Wesoła charakteryzacja równoważności)

f jest równoważnością wtedy i tylko wtedy, gdy jest surjekcją i zanurzeniem.

Wielkie odkrycie 3 - wnioski

- Nasze twierdzenie możemy odwinąć: f jest równoważnością gdy jest surjekcją i ap_f jest surjekcją i $\mathrm{ap}_{\mathrm{ap}_f}$ jest surjekcją etc.
- Wobec tego f jest równoważnością, gdy jest surjekcją na wszystkich poziomach - na punktach, ścieżkach między punktami, ścieżkach między ścieżkami etc.
- Stąd wnioskujemy, że klasyczne definicje okazują się za słabe, gdyż dotyczą tylko punktów (surjekcja) i ścieżek (injekcja), a zatem poziomów 0 i 1, a my mamy do czynienia z potencjalnie nieskończenie wieloma poziomami.

Filozoficzna interpretacja równoważności

- Przypomnijmy, że z dowolnej definicji równoważności jesteśmy w stanie uzyskać następujące rzeczy: funkcje $f : A \to B$ i $f^{-1} : B \to A$ oraz homotopie $\eta : f^{-1} \circ f \sim \mathrm{id}_A$ i $\epsilon : f \circ f^{-1} \sim \mathrm{id}_B$
- Okazuje się, że równoważność $A \simeq B$ możemy zinterpretować jako zestaw czterech reguł opisujących typ B w terminach typu A.
- Funkcja $f : A \to B$ to reguła wprowadzania.
- Funkcja $f^{-1} : B \to A$ to reguła eliminacji.
- Homotopia η to zdaniowa reguła obliczania.
- Homotopia ϵ to zdaniowa reguła unikalności.
- Powyższe rozważania okażą się przydatne za chwilę, gdy będziemy chcieli scharakteryzować przestrzenie ścieżek dla różnych typów.

Charakteryzacje ścieżek 1 - wprowadzenie

- W klasycznej matematyce mamy twierdzenia mówiące, kiedy jakieś obiekty są równe, np.
  $(a, b) = (a', b') \iff a = a' \wedge b = b'$.
- W HoTT mamy podobnie wyglądające twierdzenia:
  $(a, b) = (a', b') \simeq a = a' \times b = b'$.
- Zgodnie jednak z interpretacją homotopiczną są one dużo ogólniejsze od swoich klasycznych przodków, gdyż charakteryzują one przestrzenie ścieżek w danym typie.

Charakteryzacje ścieżek 2 - plan

- Jakkolwiek sprawa brzmi prosto, zróżnicowanie w tej materii jest spore.
- Za chwilę zobaczymy charakteryzację ścieżek w typach banalnych oraz w typach negatywnych (czyli takich, w których kluczowa jest reguła eliminacji).
- Później zobaczymy, że niektórych pożądanych charakteryzacji nie da się udowodnić i załatamy je aksjomatami.
- Następnie zobaczymy sposób pozwalający charakteryzować ścieżki w typach pozytywnych (czyli takich, w których kluczowe są reguły wprowadzania).
- HoTT pozwala nam jednak zdefiniować typy, dla których charakteryzacja ścieżek jest (częściowo) otwartym problemem badawczym, np. n-wymiarowe kule, torusy, podwieszenia i inne skomplikowane przestrzenie.

Charakteryzacje ścieżek 3 - typy banalne i negatywne

Theorem (Ścieżki między elementami typu pustego)

$\Pi x\ y : \mathbf{0}.(x = y) \simeq \mathbf{0}$

Theorem (2.8 Ścieżki między elementami typu unit)

$\Pi x\ y : \mathbf{1}.(x = y) \simeq \mathbf{1}$

Theorem (2.5.1 Ścieżki między parami)

$\Pi A\ B : \mathcal{U}.\Pi a\ a' : A.\Pi b\ b' : B.((a, b) = (a', b')) \simeq a = a' \times b = b'$

Theorem (2.7.2 Ścieżki między parami zależnymi)

$$\Pi A : \mathcal{U}.\Pi B : A \to \mathcal{U}.\Pi w\ w' : \sum_{x:A} B(x).$$
$$(w = w') \simeq \sum_{p:pr_1(w)=pr_1(w')} p_*(pr_2(w)) = pr_2(w')$$

Charakteryzacje ścieżek 4 - interpretacja

- Nie ma ścieżek między punktami typu $\mathbf{0}$. Jest to dość oczywiste, bo ścieżki muszą być między punktami, a punktów nie ma.
- Między elementami $\mathbf{1}$ jest dokładnie jedna ścieżka.
- Ścieżki między parami to pary ścieżek.
- Ścieżki między parami zależnymi to zależne pary ścieżek.

Ekstensjonalność 1 - aksjomat ekstensjonalności

Jeżeli dwie funkcje są równe, to są też homotopiczne (czyli ekstensjonalnie równe).

Definition (2.9.2)

$$\mathsf{happly} : \Pi A\ B : \mathcal{U}.\Pi f\ g : A \to B.f = g \to \Pi x : A.f(x) = g(x)$$
$$\mathsf{happly}(\mathsf{refl}_f) = \lambda x : A.\mathsf{refl}_{f(x)}$$

Jednak implikacji w drugą stronę (ani tym bardziej równoważności) nie da się pokazać. Wobec tego wprowadzamy aksjomat:

Definition (2.9.3 Aksjomat ekstensjonalności dla funkcji)

Funkcja happly jest równoważnością.

Corollary (Ładny wzorek)

$$(f = g) \simeq \Pi x : A.f(x) = g(x)$$

Ekstensjonalność 2 - rozbicie na reguły

Zauważmy, że charakteryzacje ścieżek możemy rozbić na reguły przypominające reguły opisujące typy, w których te ścieżki żyją. Jest to prawdą nie tylko dla aksjomatu ekstensjonalności, ale też np. dla twierdzeń charakteryzujących ścieżki między parami.

Corollary (Reguły opisujące ścieżki między funkcjami)

$$funext : \Pi A\ B : \mathcal{U}. \Pi f\ g : A \to B. (\Pi x : A. f(x) = g(x)) \to f = g$$
$$happly(funext(h), x) = h(x)$$
$$funext(\lambda x : A. happly(p, x)) = p$$

Ekstensjonalność 3 - interpretacja

- Reguła formacji dla $f = g$ pochodzi bezpośrednio z induktywnej definicji ścieżek.
- Reguła wprowadzania to funext: żeby zrobić ścieżkę $f = g$, wystarczy nam homotopia.
- Reguła eliminacji to happly: jeżeli mamy ścieżkę $f = g$, to możemy uzyskać ścieżkę $f(x) = g(x)$ dla dowolnego x należącego do dziedziny.
- Reguła obliczania mówi, że jeżeli zaaplikujemy do $x : A$ ścieżkę zrobioną z homotopii $h : \Pi x : A.f(x) = g(x)$ za pomocą funext, to dostaniemy $h(x)$.
- Reguła unikalności mówi, że każda ścieżka między funkcjami pochodzi od ekstensjonalności zaaplikowanej do odpowiedniej homotopii.

Ekstensjonalność 4 - charakteryzacja operacji

Możemy scharakteryzować nie tylko ścieżki między funkcjami, ale także operacje na tych ścieżkach.

Theorem (Charakteryzacja operacji na ścieżkach między funkcjami)

$$refl_f = funext(\lambda x : A.refl_{f(x)})$$
$$p^{-1} = funext(\lambda x : A.happly(p, x)^{-1})$$
$$p \cdot q = funext(\lambda x : A.happly(p, x) \cdot happly(q, x))$$

Intuicja jest prosta:

- Funkcja bierze argument i zwraca wynik.
- Ścieżka między funkcjami to funkcja biorąca argument i zwracająca ścieżkę między wynikami.
- Operacja na ścieżkach między funkcjami pochodzi na mocy ekstensjonalności od funkcji biorącej argument i wykonującej operację na ścieżkach między wynikami.

Ekstensjonalność 5 - charakteryzacja transportu

Możemy też scharakteryzować transport w rodzinach typów postaci $\lambda x : X.A(x) \to B(x)$.

Dla typu $X : \mathcal{U}$, rodzin typów $A, B : X \to \mathcal{U}$, elementów $x_1, x_2 : X$, funkcji $f : A(x_1) \to B(x_1)$ oraz ścieżki $p : x_1 = x_2$ mamy:

Theorem (Charakteryzacja transportu dla funkcji)

$$transport^{\lambda x:X.A(x)\to B(x)}(p, f) =$$
$$\lambda a : A(x_2).transport^B(p, f(transport^A(p^{-1}, a)))$$

Interpretacja twierdzenia jest łatwa: chcemy zrobić funkcję typu $A(x_2) \to B(x_2)$. Bierzemy więc element $a : A(x_2)$, transportujemy go ścieżką p w tył do typu $A(x_1)$, używamy funkcji f by dostać element typu $B(x_1)$ i transportujemy go wzdłuż p do typu $B(x_2)$.

Uniwalencja 1 - aksjomat uniwalencji

Jeżeli dwa typy są równe, to są też równoważne.

Definition (2.10.2)

$\mathrm{idtoeqv} : \Pi A\ B. A = B \to A \simeq B$
$\mathrm{idtoeqv}(p) = \mathrm{transport}^{\mathrm{id}_{\mathcal{U}}}(p)$
Słownie: idtoeqv to specjalny przypadek transportu. Trzeba jeszcze przez indukcję po ścieżkach pokazać, że jest to równoważność.

Podobnie jak w przypadku ekstensjonalności dla funkcji, w drugą stronę implikacji pokazać się nie da i stąd aksjomat.

Definition (2.10.3 Aksjomat uniwalencji)

Funkcja idtoeqv jest równoważnością.

Corollary (Ładne wzorki)

$(A = B) \simeq (A \simeq B)$ *lub równoważnie* $(A = B) = (A \simeq B)$

Uniwalencja 2 - reguły i charakteryzacje

Corollary (Reguły opisujące ścieżki między typami)

$ua : \Pi A\ B : \mathcal{U}.(A \simeq B) \to A = B$
$idtoeqv(ua(e)) = e$
$ua(idtoeqv(p)) = p$

Theorem (Charakteryzacja operacji na ścieżkach między funkcjami)

$refl_f = ua(id_A)$
$p^{-1} = ua(idtoeqv(p)^{-1})$
$p \cdot q = ua(idtoeqv(q) \circ idtoeqv(p))$

Dla rodziny typów $B : A \to \mathcal{U}$, punktów $x, y : A$, ścieżki $p : x = y$ i elementu $u : B(x)$ mamy:

Theorem (Charakteryzacja transportu dla typów)

$transport^{\lambda X:\mathcal{U}.X}(p, f) = idtoeqv(ap_B(p))(u)$

Uniwalencja 3 - przykład filozoficzny

- Rozważmy dwa poniższe typy (tak naprawdę powinniśmy też podać reguły eliminacji i obliczania, ale nie są one istotne dla przykładu).
- Niech $\mathbb{N} :\equiv 0 \mid S\ \mathbb{N}$ i niech $\mathbb{N}' :\equiv 0' \mid S'\ \mathbb{N}'$
- Rodzi się pytanie: czy $\mathbb{N}$ i $\mathbb{N}'$ to to samo, czy coś innego?
- Odpowiedź klasyczna: istnieje oczywista bijekcja $\mathbb{N} \cong \mathbb{N}'$. 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 $\mathrm{ua}(e) : \mathbb{N} = \mathbb{N}'$.

Uniwalencja 4 - przykład praktyczny

- Aksjomat uniwalencji nie tylko usuwa nieprzyjemny filozoficzny smrodek, ale daje nam też nowe sposoby rozumowania.
- Definicja: $f : B \to C$ jest monomorfizmem gdy dla dowolnych $g, h : A \to B$ jeżeli $f \circ g = f \circ h$ to $g = h$.
- Twierdzenie: każda równoważność jest monomorfizmem.
- Dowód klasyczny: każda równoważność ma odwrotność. Użyj jej.
- Dowód HoTTowy: na mocy uniwalencji każda równoważność pochodzi od jakiejś ścieżki. Na mocy indukcji po ścieżkach możemy założyć, że ścieżka ta jest trywialna, a zatem nasza równoważność jest identycznością. Wtedy nasze założenie zamienia się na $\mathrm{id}_B \circ g = \mathrm{id}_B \circ h$ i oblicza się do $g = h$, co mieliśmy pokazać.

Filozoficzna interpretacja charakteryzacji i aksjomatów

- Na mocy naszej interpretacji równoważności nasze charakteryzacje opisują przestrzenie ścieżek tak dokładnie, jakby były one osobnymi typami zdefiniowanymi za pomocą reguł.
- W przypadkach, w których nie jesteśmy w stanie udowodnić charakteryzacji, dajemy sobie protezę w postaci odpowiednich aksjomatów.
- Tak więc aksjomat ekstensjonalności dla funkcji możemy postrzegać jako charakteryzację ścieżek między funkcjami za pomocą reguł.
- Podobnie aksjomat uniwalencji możemy postrzegać jako aksjomat ekstensjonalności dla uniwersum, czyli charakteryzację ścieżek między typami za pomocą reguł.

Metoda encode-decode 1 - wstęp

- Ogólna metoda pozwalająca scharakteryzować ścieżki (niektórych) typów (głównie pozytywnych) nosi nazwę encode-decode.
- Metoda składa się z czterech kroków.
- Krok 1: definiujemy rodzinę typów code : $A \to A \to \mathcal{U}$, której celem jest opisanie typu $x =_A y$ w bardziej ludzki sposób.
- Krok 2: deiniujemy funkcję encode : $\Pi x\ y : A.x = y \to \text{code}(x, y)$
- Krok 3: definiujemy funkcję decode : $\Pi x\ y : A.code(x, y) \to x = y$
- Krok 4: pokazujemy, że encode i decode są swoimi odwrotnościami.
- Dzięki temu dostajemy charakteryzację postaci $\Pi x\ y : A.(x = y) \simeq \text{code}(x, y)$.

Metoda encode-decode 2 - definicje dla bool

Scharakteryzujmy ścieżki w typie **2**.

Definition (code)

$\mathrm{code}(0_{\mathbf{2}}, 0_{\mathbf{2}}) :\equiv \mathbf{1}$
$\mathrm{code}(1_{\mathbf{2}}, 1_{\mathbf{2}}) :\equiv \mathbf{1}$
$\mathrm{code}(\_, \_) :\equiv \mathbf{0}$

Definition (encode)

$\mathrm{encode}(\mathrm{refl}_{0_{\mathbf{2}}}) :\equiv *$
$\mathrm{encode}(\mathrm{refl}_{1_{\mathbf{2}}}) :\equiv *$

Definition (decode)

$\mathrm{decode}_{0_{\mathbf{2}},0_{\mathbf{2}}}(*) :\equiv \mathrm{refl}_{0_{\mathbf{2}}}$
$\mathrm{decode}_{1_{\mathbf{2}},1_{\mathbf{2}}}(*) :\equiv \mathrm{refl}_{1_{\mathbf{2}}}$
$\mathrm{decode}_{0_{\mathbf{2}},1_{\mathbf{2}}}(x) :\equiv \mathrm{ind}_{\mathbf{0}}(\lambda\_.0_{\mathbf{2}} = 1_{\mathbf{2}}, x)$ (czyli sprzeczność)
$\mathrm{decode}_{1_{\mathbf{2}},0_{\mathbf{2}}}(x) :\equiv$ też sprzeczność

Metoda encode-decode 3 - twierdzenia dla bool

Theorem (encode-decode)

$encode(decode(c)) = c$

Theorem (decode-encode)

$decode(encode(p)) = p$

Corollary

$\Pi x\ y : \mathbf{2}.(x = y) \simeq code(x, y)$

Corollary

$0_2 \neq 1_2$

Ćwiczenie: udowodnij.

Metoda encode-decode 4 - interpretacja dla bool

- Podobnie jak poprzednio, naszą charakteryzację możemy zinterpretować regułowo.
- decode jest tutaj regułą wprowadzania. Dokładnie opisuje ona, jak zrobić każdą z 4 potencjalnie możliwych ścieżek, np. jeżeli chcesz zrobić ścieżkę $0_2 = 0_2$ to daj mi $* : \mathbf{1}$, a jeżeli chcesz zrobić ścieżkę $0_2 = 1_2$, to daj mi $x : \mathbf{0}$.
- encode to reguła eliminacji. Dla faktycznie równych argumentów nie mówi nam ona nic ciekawego. Dla różnych argumentów daje nam ona natomiast sprzeczność.
- Twierdzenie decode-encode to reguła obliczania, a twierdzenie encode-decode to reguła unikalności.
- Reguły obliczania i unikalności są mało ciekawe, a dodatkowo mogą się różnić w zależności od sposobu, w jaki udowodniono twierdzenie. W naszym przypadku akurat jest tylko jeden słuszny sposób, ale w przypadku bardziej skomplikowanych typów niekoniecznie.

Ścieżki między ścieżkami 1 - twierdzenie i przykłady

Theorem

Jeżeli $f : A \to B$ jest równoważnością, to dla dowolnych $x, y : A$ funkcja $\mathsf{ap}_f : x = y \to f(x) = f(y)$ też jest równoważnością.

Dowód.

Odwrotnością ap_f jest oczywiście $\mathsf{ap}_{f^{-1}}$. □

- Paths $p = q$, where $p, q : w =_{A \times B} w'$, are equivalent to pairs of paths

$$\mathsf{ap}_{\mathsf{pr}_1} p =_{\mathsf{pr}_1 w =_A \mathsf{pr}_1 w'} \mathsf{ap}_{\mathsf{pr}_1} q \quad \text{and} \quad \mathsf{ap}_{\mathsf{pr}_2} p =_{\mathsf{pr}_2 w =_B \mathsf{pr}_2 w'} \mathsf{ap}_{\mathsf{pr}_2} q.$$

- Paths $p = q$, where $p, q : f =_{\prod_{(x:A)} B(x)} g$, are equivalent to homotopies

$$\prod_{x:A} (\mathsf{happly}(p)(x) =_{f(x)=g(x)} \mathsf{happly}(q)(x)).$$

Ścieżki między ścieżkami 2 - interpretacja

- Z powyższego twierdzenia płyną daleko idące wnioski.
- Jeżeli mamy charakteryzację typu A za pomocą równowazności $A \simeq B$, to mamy też charakteryzację ścieżek w A (na dowolnym poziomie) za pomocą ścieżek w B.
- Jeżeli mamy charakteryzację ścieżek między punktami w A, to mamy charakteryzację dowolnych ścieżek w A.
- Tak więc ścieżki między ścieżkami między parami to pary ścieżek między ścieżkami między komponentami par.
- Ścieżki między ścieżkami między funkcjami to homotopie na happly.

Strukturalizm 1 - ścieżki między półgrupami

Definition (Półgrupa)

$$\text{Semigroup} :\equiv \sum_{A:\mathcal{U}} \sum_{m:A\to A\to A} \prod_{x\ y\ z:A} m(x, m(y,z)) = m(m(x,y),z)$$

- Półgrupa to typ wraz z działaniem binarnym, które jest łączne.
- Z charakteryzacji ścieżek dla par zależnych, funkcji oraz z aksjomatu uniwalencji wynika, że typ ścieżek między półgrupami jest równoważny typowi równoważności na ich nośnikach, które zachowują działanie m.
- Tak więc ścieżka między półgrupami to homomorficzna równoważność, czyli, w klasycznym rozumieniu, izomorfizm półgrup.

Strukturalizm 2 - filozofia w praktyce

- Strukturalizm to filozofia matematyki, która twierdzi, że teorie matematyczne opisują strukturę obiektów matematycznych.
- Strukturę obiektu można rozumieć jako związki łączące go z innymi obiektami.
- Wobec tego obiekty matematyczne nie posiadają żadnych wewnętrznych właściwości i są zdeterminowane przez swoją strukturę.
- W praktyce znaczy to na przykład, że ze strkturalistycznego punktu widzenia izomorficzne półgrupy są identyczne.
- HoTT świetnie realizuje filozoficzne założenia strukturalizmu - jak się przekonaliśmy, izomorficzne półgrupy faktycznie są równe, czyli połączone ścieżką w typie półgrup.

Klasyfikacja typów

- W HoTT poza punktami mamy też różne mniej lub bardziej skomplikowane ścieżki.
- Wobec tego mądrym pomysłem wydaje się klasyfikowanie typów ze względu na złożoność ścieżek, jakie w nich występują.
- n-typ, to (w przybliżeniu) typ, w którym wszystkie ścieżki powyżej n-tego poziomu są trywialne. n-typy bywają też nazywane typami n-obciętymi (ang. n-truncated).
- Dualnym pojęciem jest pojęcie n-spójności (ang. n-connectedness). Typ n-spójny to taki, w którym wszystkie ścieżki poniżej n-tego poziomu są trywialne.
- My zajmiemy się tylko typami n-obciętymi. Zanim jednak omówimy je w ogólności, zobaczmy kilka pierwszych poziomów.

Ściągalność 1 - definicja i intuicja

Typ jest ściągalny to taki, który ma punkt centralny, który jest połączony ścieżką z każdym innym punktem tego typu.

Definition (Ściągalność)

$$\mathrm{isContr}(A) :\equiv \sum_{c:A} \prod_{x:A} c = x$$

Ściągalność 2 - właściwości

- $\mathrm{isContr}(A) \to A = \mathbf{1}$
- Jeżeli dla każdego $x : A$ typ $B(x)$ jest ściągalny, to $\sum_{x:A} B(x) \simeq A$
- Jeżeli typ A jest ściągalny, to $\sum_{x:A} B(x) \simeq B(c)$, gdzie c jest centrum ściągalności typu A.

Ściągalność 3 - filozofowanie

- Mogłoby się wydawać, że ściągalność jest nieciekawa, skoro typy ściągalne są równoważne (czyli równe) typowi $\mathbf{1}$.
- Kluczowy jest jednak fakt, że typy, które same w sobie są wysoce nietrywialne i z pewnością nie są ściągalne, złożone do kupy mogą dać typ, który jest ściągalny.
- Jest tak dlatego, że wzajemne zależności między obiektami mogą wykluczyć wszystkie kombinacje poza jedną.
- Najprostszym przykładem tego rodzaju jest sparowanie jakiegoś obiektu ze specyfikacją, która charakteryzuje go unikalnie.
- Co ważne, nawet jeżeli typ jest ściągalny, to zachowuje swoje właściwości obliczeniowe.

Ściągalność 4 - przykłady

- Oczywiście $\mathbf{1}$ jest ściągalny.
- Ściągalny jest typ funkcji sortujących. Mimo, że funkcji między listami oraz algorytmów sortujących jest dużo, to dobrze wyrażona specyfikacja sortowania charateryzuje je unikalnie.
- Dla dowolnego $a : A$ typ $\sum_{x:A} a = x$ jest ściągalny i dlatego właśnie indukcja po ścieżkach działa.

Zdania 1 - definicja i intuicja

Definition (Zdanie)

$\mathrm{isProp}(A) :\equiv \prod_{x,y:A} x = y$
$\mathrm{Prop} :\equiv \sum_{A:\mathcal{U}} \mathrm{isProp}(A)$

Zdania to typy, które mogą mieć co najwyżej jeden element. Liczy się zatem istnienie elementu, a nie jego specyfika. Sytuacja ta przypomina klasyczne rozumienie logiki, gdzie liczy się przeprowadzenie dowodu, a nie jego postać.

Zdania 2 - przykłady

- $\mathbf{0}$ i $\mathbf{1}$ są zdaniami. Odpowiadają one fałszowi i prawdzie.
- Typ isequiv(f) jest zdaniem.
- Typ qinv(f) może nie być zdaniem i dlatego właśnie mieliśmy problem.
- Produkty, funkcje i funkcje zależne zachowują bycie zdaniem, ale suma i suma zależna NIE zachowują bycia zdaniem.

Zbiory 1 - definicja i intuicja

Definition (Zbiór)

isSet$(A) :\equiv \prod_{x,y:A} \prod_{p,q:x=y} p = q$
Set $:\equiv \sum_{A:\mathcal{U}} isSet(A)$

Poetycko mówiąc: zbiór to worek z kropkami. Wszystkie ścieżki między kropkami są trywialne.

Zbiory 2 - przykłady

- Każde zdanie jest zbiorem, więc w szczególności $\mathbf{0}$ i $\mathbf{1}$.
- Na mocy naszej charakteryzacji typ $\mathbf{2}$ jest zbiorem.
- Typ $\mathbb{N}$ jest zbiorem (ćwiczenie: udowodnij za pomocą metody encode-decode).
- Wszystko, co jesteśmy w stanie zdefiniować za typów induktywnych (takich jak np. w Coqu) jest zbiorem, pod warunkiem że jako argumenty konstruktorów również bierze zbiory.
- W szczególności, suma (rozłączna) zbiorów jest zbiorem.

Grupoidy

Grupoid to typ żyjący na poziomie 1. Może on mieć punkty i ścieżki między punktami, ale nie ścieżki między ścieżkami.

Definition

$\mathrm{isGrpd}(A) :\equiv \prod_{x,y:A} \prod_{p,q:x=y} \prod r, s : p = qr = s$
$\mathrm{Grpd} :\equiv \sum_{A:\mathcal{U}} \mathrm{isGrpd}(A)$

Przykłady:

- Uniwersum wszystkich zbiorów Set jest grupoidem.
- Okrąg $\mathbb{S}^1$ jest grupoidem.

n-typy 1 - definicja

Na mocy pewnych zaszłości historycznych numerowanie n-typów zaczyna się od -2, a nie od 0.

Definition (n-typ)

$$\text{is-}(-2)\text{-Type}(A) :\equiv \sum_{c:A} \prod_{x:A} c = x$$
$$\text{is-}(n+1)\text{-Type}(A) :\equiv \prod_{x,y:A} \text{is-}n\text{-Type}(x = y)$$
$$n\text{-Type} :\equiv \sum_{A:\mathcal{U}} \text{is-}n\text{-Type}(A)$$

Widać, że:

- -2-typy to typy ściągalne.
- -1-typy to zdania.
- 0-typy to zbiory.
- 1-typy to grupoidy.

n-typy 2 - właściwości

- Jeżeli A jest n-typem, to jest też $(n+1)$-typem. Intuicyjnie: skoro A ma trywialną strukturę powyżej poziomu n, to ma też trywialną strukturę powyżej poziomu $(n+1)$
- Jeżeli $A : \mathcal{U}$ jest n-typem i $B : A \to \mathcal{U}$ jest rodziną n-typów, to $\sum_{x:A} B(x)$ oraz $\prod_{x:A} B(x)$ także są n-typami.
- W szczególności jeżeli A i B są n-typami, to $A \times B$ oraz $A \to B$ także są n-typami.
- Jeżeli A i B są n-typami dla $n \geqslant 0$, to $A + B$ jest n-typem.
- Uwaga: jeżeli A i B są -2-typami, to $A + B$ jest 0-typem. Podobnie jeżeli A i B są -1-typami, to $A + B$ może być -2-typem, -1-typem lub 0-typem.
- Typ is-n-Type jest zdaniem.

n-typy 3 - (kontr)przykłady

- Typ $\mathbf{1}$ jest n-typem dla każdego n.
- Typ n-Type jest $(n+1)$-typem.
- Jednak ważniejsze jest to, jakie typy nie są n-typami.
- n-te uniwersum $\mathcal{U}_n$ nie jest n-typem.
- W klasycznej teorii homotopii k-wymiarowa kula $\mathbb{S}^k$ nie jest n-typem dla żadnego $k \geqslant 2$. W HoTTbooku nie ma na to dowodu.
- Jeżeli powyższe jest prawdą, to żadne uniwersum nie jest n-typem, bo zawiera kulę.
- Konkretny (kontr)przykład (8.8.6): Niech $A :\equiv \prod_{n:\mathbb{N}} B(n)$, gdzie $B : \mathbb{N} \to \mathcal{U}$ jest taką rodziną typów, że typ $B(n)$ zawiera n-pętlę, które nie jest równa n-pętli trywialnej. Wtedy A nie jest n-typem dla żadnego $n : \mathbb{N}$.

Typy induktywne 1 - przypomnienie

- Typ induktywny to typ wygenerowany w sposób “wolny” przez kolekcję konstruktorów.
- Typ $\mathbf{0}$ jest generowany przez brak konstruktorów.
- Typ $\mathbf{1}$ jest generowany przez konstruktor $* : \mathbf{1}$
- Typ $\mathbf{2}$ jest generowany przez konstruktory $0_{\mathbf{2}} : \mathbf{2}$ oraz $1_{\mathbf{2}} : \mathbf{2}$.
- Typ $\mathbb{N}$ jest generowany przez konstruktory 0 oraz $\mathsf{succ} : \mathbb{N} \to \mathbb{N}$
- List to parametryczna rodzina typów generowana przez konstruktory $\mathsf{Nil} : \Pi A : \mathcal{U}.\mathsf{List}(A)$ oraz $\mathsf{Cons} : \Pi A : \mathcal{U}.A \to \mathsf{List}(A) \to \mathsf{List}(A)$
- $=$ to indeksowana rodzina typów generowana tak jak było napisane na wcześniejszych slajdach.

Typy induktywne 2 - więcej przykładów

Za pomocą typów induktywnych da się zdefiniować dużo różnych rzeczy. Ćwiczenie: spróbuj zdefiniować:

- Rodzinę Tree drzew trzymających elementy danego typu wyłącznie w liściach (drzewo może być puste).
- Rodzinę Sorted taką, że $\text{Sorted}(R, l)$ ma element, gdy lista l jest posortowana według relacji porządku R i nie ma elementu w przeciwnym razie.
- Rodzinę Perm taką, że $\text{Perm}(l_1, l_2)$ ma element, gdy l_1 jest permutacją l_2 i nie ma elementu, gdy l_1 nie jest permutacją l_2.
- Rodzinę FreeMon taką, że typ $\text{FreeMon}(A)$ jest wolnym monoidem na typie A. Uwaga: podchwytliwe.

Wyższe typy induktywne 1 - motywacja

Typy induktywne nie są jednak wszechmocne. Niektórych rzeczy zdefiniować się nie da:

- Nie da się zdefiniować typów ilorazowych, np. nie da się zdefiniować liczb wymiernych $\mathbb{Q}$ jako par $\mathbb{Z} \times \mathbb{Z}$ podzielonych przez relację równoważności $\sim$ zdefiniowaną jako $(a, b) \sim (a', b') :\equiv ab' = a'b$.
- Nie da się zdefiniować rodziny FreeGrp, która reprezentuje grupę wolną na danym typie.
- W ogólności, nie da się zdefiniować niczego, co wymaga utożsamienia ze sobą dwóch elementów danego typu. Wynika to z faktu, że konstruktory typów induktywnych są injektywne.

Wyższe typy induktywne 2 - sposób

- Wyższe typy induktywne generalizują typy induktywne w następujący sposób.
- Pozwalamy sobie na to, że poza zwykłymi konstruktyorami (które od teraz będziemy nazywać konstruktorami punktów) możemy robić też konstruktory ścieżek, które wkładają do typu nowe ścieżki.
- Ponieważ ścieżki to równość, możemy dzięki temu utożsamiać ze sobą różne punkty i w ten sposób zdefiniować rzeczy z powyższego slajdu.
- Ale otwierają się przed nami również inne możliwości: możemy bezpośrednio definiować przestrzenie topologiczne takie jak odcinek albo okrąg, wesołe konstrukcje logiczne takie jak n-trunkacja, która przekształca dany typ w n-typ, oraz rzeczy przydatne w teorii kategorii, np. kogranice.

Wyższe typy induktywne 3 - filozofia

- Przypomnijmy, że w typach zdefiniowanych przez wyższą indukcję mogą być rzeczy, których tam nie włożyliśmy. Np. jeżeli wrzucimy do typu jakąś ścieżkę p, to pojawi się w nim także ścieżka p^{-1}.
- A priori nie wiadomo, jak się zachowują włożone przez nas ścieżki. Mogą one być trywialne albo i nie.
- Mimo, że możemy do typu dodawać ścieżki, to wciąż definiujemy pojedynczy typ - nie redefiniujemy typu $=$. Definiowany przez nas typ dostanie swoją własną regułę indukcji, a reguła indukcji po ścieżkach pozostanie taka jak była - nowe ścieżki nie mają na nią wpływu.
- Wymiar konstruktora ma nikły wpływ na strukturę ścieżek: jeżeli definiując B dodamy konstruktor punktów typu $A \to B$, to wszystkie ścieżki z A zostają wstrzyknięte do B.

Odcinek 1 - definicja

Odcinek 2 - właściwości

Theorem (Lemat 6.3.1 Odcinek jest ściągalny)

$\mathit{isContr}(\mathbb{I})$

Okrąg 1 - definicja

$$\frac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash \mathbb{S}^1 : \mathcal{U}_i}\ \mathbb{S}^1\text{-FORM} \qquad \frac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash \mathsf{base} : \mathbb{S}^1}\ \mathbb{S}^1\text{-INTRO}_1 \qquad \frac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash \mathsf{loop} : \mathsf{base} =_{\mathbb{S}^1} \mathsf{base}}\ \mathbb{S}^1\text{-INTRO}_2$$

$$\frac{\Gamma, x{:}\mathbb{S}^1 \vdash C : \mathcal{U}_i \qquad \Gamma \vdash b : C[\mathsf{base}/x] \qquad \Gamma \vdash \ell : b =^{C}_{\mathsf{loop}} b \qquad \Gamma \vdash p : \mathbb{S}^1}{\Gamma \vdash \mathsf{ind}_{\mathbb{S}^1}(x.C, b, \ell, p) : C[p/x]}\ \mathbb{S}^1\text{-ELIM}$$

$$\frac{\Gamma, x{:}\mathbb{S}^1 \vdash C : \mathcal{U}_i \qquad \Gamma \vdash b : C[\mathsf{base}/x] \qquad \Gamma \vdash \ell : b =^{C}_{\mathsf{loop}} b}{\Gamma \vdash \mathsf{ind}_{\mathbb{S}^1}(x.C, b, \ell, \mathsf{base}) \equiv b : C[\mathsf{base}/x]}\ \mathbb{S}^1\text{-COMP}_1$$

$$\frac{\Gamma, x{:}\mathbb{S}^1 \vdash C : \mathcal{U}_i \qquad \Gamma \vdash b : C[\mathsf{base}/x] \qquad \Gamma \vdash \ell : b =^{C}_{\mathsf{loop}} b}{\Gamma \vdash \mathbb{S}^1\text{-}\mathsf{loopcomp} : \mathsf{apd}_{(\lambda y.\, \mathsf{ind}_{\mathbb{S}^1}(x.C,b,\ell,y))}(\mathsf{loop}) = \ell}\ \mathbb{S}^1\text{-COMP}_2$$

In $\mathsf{ind}_{\mathbb{S}^1}$, x is bound in C. The notation $b =^{C}_{\mathsf{loop}} b$ for dependent paths was introduced in §6.2.

Okrąg 2 - właściwości

Theorem (Lemat 6.2.9 Okrąg to faktycznie okrąg)

$(\mathbb{S}^1 \to A) \simeq \sum_{x:A} x = x$

Theorem (Lemat 6.4.1)

$loop \neq refl_{base}$

Dowód.

Załóżmy, że loop $=$ refl$_{\text{base}}$. Weźmy dowolne typ A z punktem $x : A$ i pętlą $p : x = x$ (są takie). Wtedy na mocy rekursora dla $\mathbb{S}^1$ możemy zdefiniować funkcję $f : \mathbb{S}^1 \to A$ następująco:
$f(\text{base}) :\equiv x$
$\text{ap}_f(\text{loop}) := p$
Mamy stąd $p = \text{ap}_f(\text{loop}) = \text{ap}_f(\text{refl}_{\text{base}}) = \text{refl}_{f(\text{base})} = \text{refl}_x$.
Wobec tego każdy typ mający choć jeden punkt jest zbiorem, co jest sprzeczne (bo np. uniwersum nie jest zbiorem). □

HIT

- todo

HIT

- todo

3.7 Trunkacja

- todo

3.4 Logika klasyczna i intuicjonistyczna

- Świadczy to o tym, że są "zbyt konstruktywne", tzn. zawierają w sobie za dużo informacji. Stoi to w sprzeczności z tradycyjną logiką, gdzie spójniki logiczne przekształcają zdania w zdania.

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)

$$\prod(A:\mathcal{U})(B:A\to\mathcal{U})(R:\Pi x:A, B\ x\to\mathcal{U}),$$
$$\left(\prod x:A,\sum y:B\ x, R\ x\ y\right)\to$$
$$\sum f:(\Pi x:A, B\ x), \Pi x:A, R\ x\ (f\ x)$$

Dowód.

Na tablicy. □

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

3.9 Zasada unikalnego wyboru

•

wut

-

Bibliografia

- Podstawowym źródłem wiedzy jest książka https://homotopytypetheory.org/book/
- Jakaś prezentacja: http://www.cs.nott.ac.uk/~psztxa/talks/edinburgh-13.pdf
- Filozoficzne wynurzenia: https://www.researchgate.net/publication/280671356_Does_Homotopy_Type_Theory_Provide_a_Foundation_for_Mathematics
- Wesoły papiur o trunkacji i topologicznych rzeczach: https://arxiv.org/pdf/1610.03346.pdf