Homotopiczna teoria typów

Wojciech Kołowski

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

- Teorie typów to systemy formalne, który za pomocą reguł (osądów) opisują byty zwane typami.
- Kluczową innowacją homotopicznej teorii typów jest interpretacja typów jako przestrzeni i zainspirowane nią aksjomaty rzucające światło na naturę wszechrzeczy.
- 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 formowania, wprowadzania, eliminacji, obliczania i unikalności.

Teoria typów 2 - pięć rodzajów reguł

- Reguły formowania 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 formowania, która obliczania etc.)

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

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

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

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

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

- Uwaga: w powyższych regułach $\mathcal{U}$ oznacza uniwersum, czyli typ typów, a zatem $\Gamma \vdash A : \mathcal{U}$ oznacza "A jest typem w kontekście Γ".

Teoria typów 4 - ciekawostki o regułach

- Każdy typ musi mieć regułę formowania - 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 formowania 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 $:\equiv \lambda A : \mathcal{U}.\lambda B : \mathcal{U}.\lambda x : A \times B.(\mathrm{pr}_2(x), \mathrm{pr}_1(x))$
- 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 $H : [0;1] \times X \to Y$ 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.

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.

Ścieżki 2 - interpretacja reguł

- Reguła formowania: 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. Jeżeli typ A jasno wynika z kontekstu, to będziemy pisać po prostu $a = b$ zamiast $a =_A b$.
- 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.

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: HoTTBook, podrozdział 1.12.2.

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.

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 badania nad takimi właśnie zagadnieniami doprowadziły do odkrycia homotopicznej interpretacji teorii typów.

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 $(x, y, p) = (x, x, \mathsf{refl}_x)$, 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 $(x, p) = (a, \mathsf{refl}_a)$ w typie $\Sigma x : A, a = x$. Odpowiada to indukcji po ścieżkach w wersji based.

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

Uwaga: w nagłówkach definicji i lematów pojawiają się numery, które pokazują, gdzie w HoTTBooku można znaleźć daną definicję/twierdzenie. Link do samego HoTTBooka będzie na końcu slajdów.

2.1.1 Ścieżka odwrotna

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

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

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

2.1.4 Właściwości operacji na ścieżkach

Niech $A : \mathcal{U}$ będzie typem, $x, y, z, w : A$ punktami, zaś $p : x = y, q : y = z, r : z = w$ ścieżkami. Wtedy:

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

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

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

2.3.1 Transport

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

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

Lemat 2.3.4 Aplikacja funkcji zależnej do ścieżki

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

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

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 dziurę 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

Kwaziodwrotność

$\mathsf{qinv} : \Pi A\ B : \mathcal{U}.(A \to B) \to \mathcal{U}$

$$\mathsf{qinv}(f) :\equiv \sum_{g:B\to A} g \circ f \sim \mathsf{id}_A \times f \circ g \sim \mathsf{id}_B$$

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

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

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

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:

Twierdzenie 4.1.1 Kwaziodwrotności to pętle

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

Twierdzenie to głosi, że jeżeli funkcja f jest kwaziodwrotnością, to typ $\mathrm{qinv}(f)$ 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:
  - $\mathrm{qinv}(f) \to \mathrm{isequiv}(f)$
  - $\mathrm{isequiv}(f) \to \mathrm{qinv}(f)$
  - $\Pi e_1\ e_2 : \mathrm{isequiv}(f). e_1 = e_2$
- 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.

Równoważność 1

$$\mathrm{isequiv}_1(f) :\equiv \left(\sum_{g : B \to A} f \circ g \sim \mathrm{id}_B \right) \times \left(\sum_{h : B \to A} h \circ f \sim \mathrm{id}_A \right)$$

Równoważność 2

$$\mathrm{isequiv}_2(f) :\equiv \sum_{g : B \to A} \sum_{\eta : g \circ f \sim \mathrm{id}_A} \sum_{\epsilon : f \circ g \sim \mathrm{id}_B} \prod_{x : A} \mathrm{ap}_f(\eta(x)) = \epsilon(\mathrm{ap}_f(x))$$

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

Równoważność 3

$$\mathrm{isequiv}_3(f) :\equiv \prod_{y:B} \mathrm{isContr}\left(\sum_{x:A} f(x) = y\right)$$

Równoważność 4

$\mathrm{isequiv}_4(f) :\equiv ||\mathrm{qinv}(f)||$

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.
- Dla precyzji jednak przyjmujemy $\mathrm{isequiv}(f) :\equiv \mathrm{isequiv}_1(f)$, tzn. domyślnie będziemy posługiwać się definicją z dwiema osobnymi odwrotnościami.
- 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.

Surjekcja - nieco inaczej niż w książce

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

Zanurzenie

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

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 $\mathsf{ap}_{\mathsf{ap}_f}$ 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

Ścieżki między elementami typu pustego

$\Pi x\ y : \mathbf{0}.(x = y) \simeq \mathbf{1}$

2.8 Ścieżki między elementami typu unit

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

2.5.1 Ścieżki między parami

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

2.7.2 Ścieżki między parami zależnymi

$$\Pi A : \mathcal{U}.\Pi B : A \to \mathcal{U}.\Pi w\ w' : \textstyle\sum_{x:A} B(x).$$
$$(w = w') \simeq \sum_{p:\mathrm{pr}_1(w)=\mathrm{pr}_1(w')} p_*(\mathrm{pr}_2(w)) = \mathrm{pr}_2(w')$$

Charakteryzacje ścieżek 4 - interpretacja

- Między każdymi dwoma elementami typu pustego jest dokładnie jedna ścieżka. Albo i nie, albo i dwie. Nie ma to większego znaczenia, bo jeżeli wpadnie nam w ręce element typu pustego, to i tak mamy sprzeczność.
- Między elementami **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).

2.9.2 Aplikacja homotopii do argumentu

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

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

2.9.3 Aksjomat ekstensjonalności dla funkcji

Funkcja happly jest równoważnością.

Wniosek: charakteryzacja ścieżek między funkcjami

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

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.

Reguły opisujące ścieżki między funkcjami

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

$\mathrm{happly}(\mathrm{funext}(h), x) = h(x)$

$\mathrm{funext}(\lambda x : A.\mathrm{happly}(p, x)) = p$

Ekstensjonalność 3 - interpretacja

- Reguła formowania 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.

Charakteryzacja operacji na ścieżkach między funkcjami

$$\mathsf{refl}_f = \mathsf{funext}(\lambda x : A.\mathsf{refl}_{f(x)})$$
$$p^{-1} = \mathsf{funext}(\lambda x : A.\mathsf{happly}(p, x)^{-1})$$
$$p \cdot q = \mathsf{funext}(\lambda x : A.\mathsf{happly}(p, x) \cdot \mathsf{happly}(q, x))$$

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:

Charakteryzacja transportu dla funkcji

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

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.

2.10.2 Jak przerobić ścieżkę na równoważność

$\mathrm{idtoeqv} : \Pi A\ B . A = B \to A \simeq B$
$\mathrm{idtoeqv}(p) = \mathrm{transport}^{\mathrm{id}_{\mathcal{U}}}(p)$
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.

2.10.3 Aksjomat uniwalencji

Funkcja idtoeqv jest równoważnością.

Aksjomat uniwalencji, tylko ładniej zapisany

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

Uniwalencja 2 - reguły i charakteryzacje

Reguły opisujące ścieżki między typami

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

Charakteryzacja operacji na ścieżkach między typami

$\mathsf{refl}_f = \mathsf{ua}(\mathsf{id}_A)$
$p^{-1} = \mathsf{ua}(\mathsf{idtoeqv}(p)^{-1})$
$p \cdot q = \mathsf{ua}(\mathsf{idtoeqv}(q) \circ \mathsf{idtoeqv}(p))$

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

Charakteryzacja transportu dla typów

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

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 $\mathrm{code} : A \to A \to \mathcal{U}$, której celem jest opisanie typu $x =_A y$ w bardziej ludzki sposób.
- Krok 2: deiniujemy funkcję $\mathrm{encode} : \Pi x\ y : A.x = y \to \mathrm{code}(x, y)$
- Krok 3: definiujemy funkcję $\mathrm{decode} : \Pi x\ y : A.code(x, y) \to x = y$
- Krok 4: pokazujemy, że encode i decode są swoimi odwrotnościami.
- Dzięki temu dostajemy charakteryzację postaci $\Pi x\ y : A.(x = y) \simeq \mathrm{code}(x, y)$.

Metoda encode-decode 1.5 - objaśnienie

- Uwaga: doszły mnie słuchy, że są problemy z odszyfrowaniem poniższego slajdu, więc wyjaśniam.
- $\mathbf{2}$ (pogrubiona cyfra dwa) oznacza typ dwuelementowy, znany szerzej jako typ `bool`. Jego dwa elementy to 0_2 oraz 1_2 (chude 0 i 1 z indeksem dolnym $\mathbf{2}$). Można je interpretować, odpowiednio, jako `false` i `true`, czyli prawdę i fałsz.
- $\mathbf{0}$ (pogrubiona cyfra zero) oznacza typ pusty, który nie ma żadnych elementów.
- $\mathbf{1}$ (pogrubiona cyfra jeden) oznacza typ jednoelementowy. Jego jedyny element to $*$, czyli gwiazdka.

Metoda encode-decode 2 - definicje dla bool

Scharakteryzujmy ścieżki w typie $\mathbf{2}$.

$\text{code} : \mathbf{2} \to \mathbf{2} \to \mathcal{U}$

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

$\text{encode} : \Pi(xy : \mathbf{2}).x = y \to \text{code}(x, y)$

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

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

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

Metoda encode-decode 3 - twierdzenia dla bool

encode* jest odwrotnością *decode

$\mathrm{encode}(\mathrm{decode}(c)) = c$

decode* jest odwrotnością *encode

$\mathrm{decode}(\mathrm{encode}(p)) = p$

Wniosek 1: charakteryzacja ścieżek w typie $\mathbf{2}$

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

Wniosek 2: elementy typu $\mathbf{2}$ są różne

$0_{\mathbf{2}} \neq 1_{\mathbf{2}}$

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

Twierdzenie

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 równoważności $A \simeq B$, 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

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.

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 $\mathrm{Sorted}(R, l)$ ma element, gdy lista l jest posortowana według relacji porządku R i nie ma elementu w przeciwnym razie.
- Rodzinę Perm taką, że $\mathrm{Perm}(l_1, l_2)$ 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 $\mathrm{FreeMon}(A)$ 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 konstruktorami (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

$$\frac{}{\Gamma \vdash \mathbb{I} : \mathcal{U}}$$

$$\frac{}{\Gamma \vdash 0_{\mathbb{I}} : \mathbb{I}} \qquad \frac{}{\Gamma \vdash 1_{\mathbb{I}} : \mathbb{I}} \qquad \frac{}{\Gamma \vdash \mathsf{seg} : 0_{\mathbb{I}} = 1_{\mathbb{I}}}$$

$$\frac{\Gamma \vdash A : \mathbb{I} \to \mathcal{U} \quad \Gamma \vdash a_0 : A(0_{\mathbb{I}}) \quad \Gamma \vdash a_1 : A(1_{\mathbb{I}}) \quad \Gamma \vdash p : \mathsf{seg}_*(a_0) = a_1}{\Gamma \vdash \mathsf{ind}_{\mathbb{I}}(A, a_0, a_1, p) : \Pi i : \mathbb{I}.A(i)}$$

$$\frac{\Gamma \vdash A : \mathbb{I} \to \mathcal{U} \quad \Gamma \vdash a_0 : A(0_{\mathbb{I}}) \quad \Gamma \vdash a_1 : A(1_{\mathbb{I}}) \quad \Gamma \vdash p : \mathsf{seg}_*(a_0) = a_1}{\Gamma \vdash \mathsf{ind}_{\mathbb{I}}(A, a_0, a_1, p, 0_{\mathbb{I}}) \equiv a_0 : A(0_{\mathbb{I}})}$$

$$\frac{\Gamma \vdash A : \mathbb{I} \to \mathcal{U} \quad \Gamma \vdash a_0 : A(0_{\mathbb{I}}) \quad \Gamma \vdash a_1 : A(1_{\mathbb{I}}) \quad \Gamma \vdash p : \mathsf{seg}_*(a_0) = a_1}{\Gamma \vdash \mathsf{ind}_{\mathbb{I}}(A, a_0, a_1, p, 1_{\mathbb{I}}) \equiv a_1 : A(1_{\mathbb{I}})}$$

$$\frac{\Gamma \vdash A : \mathbb{I} \to \mathcal{U} \quad \Gamma \vdash a_0 : A(0_{\mathbb{I}}) \quad \Gamma \vdash a_1 : A(1_{\mathbb{I}}) \quad \Gamma \vdash p : \mathsf{seg}_*(a_0) = a_1}{\Gamma \vdash \mathsf{wut} : \mathsf{apd}_{\mathsf{ind}_{\mathbb{I}}(A, a_0, a_1, p)}(\mathsf{seg}) = p}$$

Odcinek 2 - uwagi

- Dla wyższych typów tak jak i dla zwykłych mamy 5 rodzajów reguł.
- Reguły formowania i eliminacji pozostają bez zmian.
- Reguły wprowadzania mogą robić ścieżki między punktami definiowanego typu.
- Reguły obliczania dla punktów również nie ulegają zmianom. Musimy jednak zadbać o to, żeby napisać reguły obliczania dla ścieżek. Robimy to za pomocą funkcji ap (dla rekursora) i apd (dla reguły indukcji). Ponieważ apd został zdefiniowany wewnątrz naszej teorii, a reguły żyją na metapoziomie, to reguły obliczania dla ścieżek nie są definicyjne (tzn. $\equiv$), tylko same są ścieżkami i wobec tego musimy je jakoś nazwać.
- Wobec tego definiując za pomocą równań i dopasowania do wzorca w przypadku ścieżek będziemy używać hybrydowej notacji $:=$

Odcinek 3 - o co tak naprawdę chodzi w indukcji

- Jeżeli reguła eliminacji dla odcinka wydaje ci się mistyczna, to posłuchaj poniższej bajki na temat indukcji.
- Każdy typ induktywny ma pewną strukturę. W przypadku odcinka są to punkty $0_{\mathbb{I}}$ i $1_{\mathbb{I}}$ wraz z łączącą je ścieżką seg.
- Rekursor (czyli upośledzona reguła eliminacji) mówi, że jeżeli znajdziemy w jakimś typie taką samą strukturę (dwa punkty a_0 i a_1 połączone ścieżką p), to możemy zrobić funkcję f, która spełnia $f(0_{\mathbb{I}}) :\equiv a_0, f(1_{\mathbb{I}}) :\equiv a_1, \mathrm{ap}_f(\mathrm{seg}) := p$.
- Reguła eliminacji z poprzedniego slajdu jest uogólnieniem powyższego opisu na typy zależne.

Odcinek 4 - właściwości

6.3.1 Odcinek jest ściągalny

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

6.3.2 Odcinek implikuje ekstensjonalność

Niech $f, g : A \to B$ będą funkcjami i niech $H : \Pi x : A.f(x) = g(x)$ będzie homotopią. Wtedy $f = g$ bez używania aksjomatu ekstensjonalności.

Odcinek 5 - dowód lematu 6.3.2

Dowód lematu 6.3.2

Przez rekursję po odcinku definiujemy nastepującą funkcję:

$p : A \to \mathbb{I} \to B$

$p(x, 0_{\mathbb{I}}) :\equiv f(x)$

$p(x, 1_{\mathbb{I}}) :\equiv g(x)$

$\mathrm{ap}_{p(x)}(\mathrm{seg}) := H(x)$

Teraz definiujemy

$q : \mathbb{I} \to (A \to B)$

$q(i) :\equiv \lambda x : A.p(x, i)$

Mamy $q(0_{\mathbb{I}}) \equiv \lambda x : A.p(x, 0_{\mathbb{I}}) \equiv \lambda x : A.f(x) \equiv f$ i analogicznie $q(1_{\mathbb{I}}) \equiv g$, a zatem $\mathrm{ap}_q(\mathrm{seg}) : f = g$

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.

Ćwiczenie: opisz rekursor dla okręgu w taki sposób, jaki opisany został rekursor dla odcinka.

Okrąg 2 - właściwości

6.2.9 Okrąg to faktycznie okrąg

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

Lemat 6.4.1

$\mathrm{loop} \neq \mathrm{refl}_{\mathrm{base}}$

Dowód lematu 6.4.1

Załóżmy, że $\mathrm{loop} = \mathrm{refl}_{\mathrm{base}}$. Weźmy dowolny 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:

$f(\mathrm{base}) :\equiv x$

$\mathrm{ap}_f(\mathrm{loop}) := p$

Mamy stąd $p = \mathrm{ap}_f(\mathrm{loop}) = \mathrm{ap}_f(\mathrm{refl}_{\mathrm{base}}) = \mathrm{refl}_{f(\mathrm{base})} = \mathrm{refl}_x$. Wobec tego każdy typ mający choć jeden punkt jest zbiorem, co jest sprzeczne (bo np. uniwersum nie jest zbiorem).

Klasyfikacja typów

- W homotopicznej teorii typów 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 właściwości

Ściągalność

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

Typ jest ściągalny gdy ma punkt centralny, który jest połączony ścieżką z każdym innym punktem tego typu. Nazwa bierze się stąd, że możemy wszystkie punkty ściągnąć do centrum. Wobec tego typ ściągalny jest równoważny typowi **1**.

Ściągalność 2 - filozofowanie

- Mogłoby się wydawać, że ściągalność jest nieciekawa, skoro typy ściągalne są równoważne (czyli równe, dzięki uniwalencji) 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 w sposób unikalny.
- Co ważne, nawet jeżeli typ jest ściągalny, to zachowuje swoje właściwości obliczeniowe.

Ściągalność 3 - przykłady

- Oczywiście $\mathbf{1}$ jest ściągalny.
- Odcinek $\mathbb{I}$ jest ściągalny. Mimo tego, jak widzieliśmy, może on posłużyć nam do udowodnienia aksjomatu ekstensjonalności.
- Ściągalny jest typ funkcji sortujących. Mimo, że funkcji między listami jest dużo, to dobrze wyrażona specyfikacja sortowania unikalnie charateryzuje jedyną słuszną funkcję sortującą.
- Dla dowolnego $a : A$ typ $\sum_{x:A} a = x$ jest ściągalny i dlatego właśnie indukcja po ścieżkach (w wersji based) działa.

Zdania

Zdanie

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

- Zdanie to typ, który może mieć co najwyżej jeden punkt. Przypomina to tradycyjną logikę: twierdzenie (zdanie) może mieć dowód albo nie.
- $\mathbf{0}$ i $\mathbf{1}$ są zdaniami. Odpowiadają one fałszowi i prawdzie.
- Dla każdego $A : \mathcal{U}$ typ $\neg A :\equiv A \to \mathbf{0}$ jest zdaniem.
- Typ $\mathrm{isequiv}(f)$ jest zdaniem, ale typ $\mathrm{qinv}(f)$ może NIE być zdaniem i dlatego właśnie mieliśmy problem.
- O ile wszystkie zdania fałszywe są równe i podobnie wszystkie zdania prawdziwe są równe, to Prop nie jest tym samym co $\mathbf{2}$, gdyż w ogólności nie jesteśmy w stanie rozstrzygnąć, czy mamy do czynienia ze zdaniem prawdziwym, czy fałszywym.

Zbiory

Zbiór

$\text{isSet}(A) :\equiv \prod_{x,y:A} \prod_{p,q:x=y} p = q$

$\text{Set} :\equiv \sum_{A:\mathcal{U}} isSet(A)$

- Intuicyjnie: zbiór to typ, w którym może być dużo punktów, ale nie są one połączone ścieżkami (tzn. wszystkie ścieżki są trywialne).
- 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 pomocą zwykłych typów induktywnych jest zbiorem, pod warunkiem że jako argumenty konstruktorów również bierze zbiory.

Grupoidy

Grupoid

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

- Grupoid to typ, który może mieć punkty i ścieżki między punktami, ale nie może mieć ścieżek między ścieżkami.
- Okrąg $\mathbb{S}^1$ jest grupoidem - widzieliśmy, że pętla loop nie jest trywialna.
- Uniwersum wszystkich zbiorów Set jest grupoidem, ponieważ typ **2** ma nietrywialną pętlę, zrobioną z negacji boolowskiej za pomocą aksjomatu uniwalencji. Nie jest ona trywialna, bo negacja nie jest identycznością.

n-typy 1 - definicja

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

n-typ

is-(-2)-Type$(A) :\equiv \sum_{c:A} \prod_{x:A} c = x$
is-$(n+1)$-Type$(A) :\equiv \prod_{x,y:A}$ is-n-Type$(x = y)$
n-Type $:\equiv \sum_{A:\mathcal{U}}$ is-n-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 (hiper)sfera $\mathbb{S}^k$ 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 hipersferę.
- 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, która nie jest równa trywialnej n-pętli. Wtedy A nie jest n-typem dla żadnego $n : \mathbb{N}$.

Logika

- Poznawszy wszystkie trzy filary homotopicznej teorii typów, rzućmy na koniec okiem na konsekwencje, jakie niosą one dla logiki.
- W teorii typów częstym sloganem jest "propositions as types", który pozwala nam kodować zdania jako typy.
- Wiemy już jednak, że nasze typy reprezentują ω-grupoidy (czyli przestrzenie). Poznaliśmy też typ zdań Prop - są to przestrzenie mogące mieć co najwyżej jeden punkt.
- Spróbujmy jakoś uporządkować ten konceptualny bałagan.

Logika klasyczna 1 - problem

Większość matematyki jest robiona klasycznie, mimo że jest to niegodziwe. Wobec tego my też chcielibyśmy w ostateczności móc zrobić trochę klasycznej matematyki. Sprawy się jednak komplikują.

3.2.2 Prawo podwójnej negacji nie zachodzi dla dowolnych typów

$\neg \prod_{A:\mathcal{U}} \neg\neg A \to A$

3.2.7 Prawo wyłączonego środka nie zachodzi dla dowolnych typów

$\neg \prod_{A:\mathcal{U}} A + \neg A$

Moje uogólnienie lematu 3.2.2

Niech $F : \mathcal{U} \to \mathrm{Prop}$ i niech $x : F(\mathbf{2})$. Wtedy

$\neg \prod_{A:\mathcal{U}} F(A) \to A$

Logika klasyczna 2 - interpretacja

- Dowód lematu 3.2.2 jest dość techniczny. Ciężko go zobrazować.
- Typ $\neg\neg A$ możemy zinterpretować jako typ A, z którego wyrzucono wszystko poza jednym punktem, o ile były w nim jakieś punkty.
- Wobec tego typ $\neg\neg A \to A$ możemy zinterpretować jako operację odzyskującą z tak obciętego typu jakiś punkt.
- Można tę operację wykonać dla każdego typu z osobna, ale nie można tego zrobić w taki sam sposób dla wszystkich typów.
- Uogólnienie lematu pokazuje, że problemem faktycznie nie jest podwójna negacja, ale utrata informacji związana z obcięciem przestrzeni.
- Jeżeli przyjrzeć się dowodowi z książki, to ostateczna konkluzja brzmi: (naiwne) prawo wyłączonego środka jest sprzeczne z aksjomatem uniwalencji.

Logika klasyczna 3 - rozwiązanie

Tak więc możemy skonkludować, że nasze przestrzenie (czyli dowolne typy) zawierają za dużo informacji, żeby robić na nich logikę. Dla zdań (czyli typów z uniwersum Prop) problem znika.

Prawo podwójnej negacji

$\mathrm{DNE} :\equiv \prod_{P:\mathrm{Prop}} \neg\neg P \to P$

Prawo wyłączonego środka

$\mathrm{LEM} :\equiv \prod_{P:\mathrm{Prop}} P + \neg P$

Powyższe definicje pokazują, jak wyglądają poprawnie sformułowane prawo podwójnej negacji oraz prawo wyłączonego środka stosujące się jedynie do zdań. Tak sformułowane aksjomaty możemy przyjąć bez popadania w sprzeczność.

Trunkacja 1 - motywacja

- Okazało się, że nasze typy zawierają za dużo informacji, aby móc służyć jako zdania. Tylko niektóre z nich się do tego nadają.
- Jednak to nie koniec problemów - nasze spójniki też nie są idealne. W tradycyjnej logice spójniki przekształcają zdania w zdania, u nas natomiast tak nie jest: suma dwóch zdań nie musi być zdaniem, np. $\mathbf{1} + \mathbf{1} = \mathbf{2}$.
- Jeszcze hardkorowiej sprawa ma się z sumą zależną: nawet jeżeli dla każdego $x : A$ typ $B(x)$ jest zdaniem, to typ $\sum_{A:\mathcal{U}} B(x)$ może być dowolnie skomplikowaną przestrzenią (bo A może być dowolnie skomplikowaną przestrzenią).
- Jak widać, nasze spójniki $+$ oraz $\sum$ są zbyt konstruktywne. Spróbujmy zatem wynaleźć takie, które zachowywałyby właściwość bycia zdaniem.

Trunkacja 2 - pomysł

- Pomysł jest prosty. Widzieliśmy już, że za pomocą wyższych typów induktywnych możemy dodawać do typów dowolne ścieżki.
- Wobec tego moglibyśmy zdefiniować spójnik $\vee$ podobnie do $+$, ale wrzucić do niego tyle ścieżek, żeby jego wynik był zdaniem.
- Będziemy jednak sprytniejsi i zdefiniujemy trunkację zdaniową (ang. propositional truncation) - operację, która zamienia każdy typ w zdanie.

Trunkacja 3 - definicja

$$\frac{\Gamma \vdash A : \mathcal{U}}{\Gamma \vdash ||A|| : \mathcal{U}}$$

$$\frac{\Gamma \vdash x : A}{\Gamma \vdash |x| : ||A||} \qquad \frac{\Gamma \vdash A : \mathcal{U} \quad \Gamma \vdash x : ||A|| \quad \Gamma \vdash y : ||A||}{\Gamma \vdash \mathrm{all\_eq} : x = y}$$

$$\frac{\Gamma \vdash A : \mathcal{U} \quad \Gamma \vdash B : \mathrm{Prop} \quad \Gamma \vdash f : A \to B}{\Gamma \vdash \mathrm{ind}_{||A||} : ||A|| \to B}$$

$$\frac{\Gamma \vdash A : \mathcal{U} \quad \Gamma \vdash B : \mathrm{Prop} \quad \Gamma \vdash f : A \to B \quad \Gamma \vdash x : A}{\Gamma \vdash \mathrm{ind}_{||A||}(|x|) \equiv f(x) : ||A||}$$

Trunkacja 4 - interpretacja

- Reguła wprowadzania $|-|$ wstrzykuje punkty A do $||A||$.
- Reguła wprowadzania all_eq sprawia, że wszystkie tak wstrzyknięte punkty są połączone ścieżkami każdy z każdym.
- Reguła eliminacji mówi nam, że jeżeli A implikuje zdanie B, to zdanie B wynika już z obciętego A.
- Reguła obliczania to tylko formalność.

Logika zdań 1 - definicje

Definition 3.7.1. We define **traditional logical notation** using truncation as follows, where P and Q denote mere propositions (or families thereof):

$$\begin{aligned}
\top &:\equiv \mathbf{1} \\
\bot &:\equiv \mathbf{0} \\
P \wedge Q &:\equiv P \times Q \\
P \Rightarrow Q &:\equiv P \to Q \\
P \Leftrightarrow Q &:\equiv P = Q \\
\neg P &:\equiv P \to \mathbf{0} \\
P \vee Q &:\equiv \| P + Q \| \\
\forall (x : A).\, P(x) &:\equiv \prod_{x:A} P(x) \\
\exists (x : A).\, P(x) &:\equiv \Big\| \sum_{x:A} P(x) \Big\|
\end{aligned}$$

The notations $\wedge$ and $\vee$ are also used in homotopy theory for the smash product and the wedge of pointed spaces, which we will introduce in Chapter 6. This technically creates a potential for conflict, but no confusion will generally arise.

Similarly, when discussing subsets as in §3.5, we may use the traditional notation for intersections, unions, and complements:

$$\begin{aligned}
\{ x : A \mid P(x) \} \cap \{ x : A \mid Q(x) \} &:\equiv \{ x : A \mid P(x) \wedge Q(x) \}, \\
\{ x : A \mid P(x) \} \cup \{ x : A \mid Q(x) \} &:\equiv \{ x : A \mid P(x) \vee Q(x) \}, \\
A \setminus \{ x : A \mid P(x) \} &:\equiv \{ x : A \mid \neg P(x) \}.
\end{aligned}$$

Logika zdań 2 - interpretacja

- Dysjunkcję definiujemy jako obciętą sumę rozłączną, zaś kwatyfikator egzystencjalny jako obciętą parę zależną. Tym sposobem uzyskujemy spójniki, za pomocą których możemy rozumować tak jak dotychczas, ale tylko gdy dowodzimy zdań.
- Gdy konstruujemy jakieś obiekty, nie możemy pozbyć się trunkacji, a więc nie możemy uzależnić naszych konstrukcji od wyboru lewego lub prawego dysjunktu ani od konkretnego punktu występującego w sumie zależnej. Dzięki temu wiemy, że dysjunkcja jest prawdziwa, ale nie wiemy, który dysjunkt jej dowodzi. Podobnie wiemy, że istnieje punkt spełniający warunek, ale nie wiemy, jaki to punkt.
- Zauważmy też, że dla zdań P i Q jeżeli $P \to Q$ oraz $Q \to P$, to $P \simeq Q$, a więc także $P = Q$. Aksjomat uniwalencji w przypadku zdań mówi nam, że ścieżki między zdaniami reprezentują równoważność logiczną.
- Pozostałe spójniki zachowują właściwość bycia zdaniem.

Zasada unikalnego wyboru 1 - rozważania

- Może się wydawać, że nasza logika jest trochę upośledzona - brak możliwości rozłożenia sumy (rozłącznej/zależnej) na przypadki podczas konstrukcji może zniechęcać do ich używania.
- Jest jednak pewna sprytna technika, która pozwala obejść to ograniczenie.
- Prezentuje się ona zazwyczaj tak: chcąc z $||A||$ skonstruować B, które może nie być zdaniem, robimy predykat $P : B \to \mathcal{U}$, taki, że typ $\sum_{x:B} P(x)$ jest ściągalny. Wtedy przy konstruowaniu funkcji $||A|| \to \sum_{x:B} P(x)$ możemy pozbyć się trunkacji.
- Po więcej zobacz rozdział 3.9 w HoTTBooku.

Zasada unikalnego wyboru 2 - przykład

Zadanie 3.19 i 3.23

Niech $P : \mathbb{N} \to \mathcal{U}$ będzie rodziną typów. Wtedy

$$\left\|\left|\sum_{n:\mathbb{N}} P(n)\right|\right\| \to \sum_{n:\mathbb{N}} P(n)$$

Dowód

Wiemy, że istnieje jakieś n spełniające P, ale nie wiemy jakie, więc nie możemy go rozpakować. Zamiast konstruować dowolne n spełniające P, możemy jednak skonstruować najmniejsze takie n, tzn. element typu $\sum_{n:\mathbb{N}} P(n) \times \prod_{m:\mathbb{N}} P(m) \to n \leqslant m$. Jest ono unikalne, a zatem cały ten typ jest zdaniem. Możemy teraz odpakować n z przesłanki i zrobić przeszukiwanie od n do 0 w poszukiwaniu minimalnego n.

Aksjomat wyboru 1 - problem

Moglibyśmy pokusić się o sformułowanie Aksjomatu Wyboru w następujący sposób.

Aksjomat "Wyboru", który tak naprawdę nic nie wybiera

Dla typu $A : \mathcal{U}$, rodziny typów $B : A \to \mathcal{U}$ oraz rodziny typów $R : \prod_{x:A} B(x) \to \mathcal{U}$ zachodzi:

$$\left(\prod_{x:A} \sum_{y:B(x)} R(x,y) \right) \to \sum_{f:\Pi x:A.B(x)} \prod_{x:A} R(x, f(x))$$

Niestety ten sposób jest upośledzony, gdyż nie jest to aksjomat, tylko twierdzenie, i to bardzo proste do udowodnienia. Mankament ten wynika z faktu, że wszystkie wybory zostały już dokonane - nasza interpretacja sumy zależnej sprawia, że żeby uzyskać konkluzję, wystarczy odpowiednio przepakować przesłankę. Ten Aksjomat "Wyboru" jest zupełnie jak wybory w Polsce.

Aksjomat wyboru 2 - trunkacja na ratunek

Jak nietrudno się domyślić, nasz problem możemy rozwiązać wciskają gdzie trzeba trunkację.

Jedyny słuszny Aksjomat Wyboru w pełnej krasie

Dla zbioru A : Set, rodziny zbiorów $B : A \to$ Set oraz relacji (czyli rodziny zdań) $R : \prod_{x:A} B(x) \to$ Prop zachodzi:

$$\left(\prod_{x:A}\left\|\sum_{y:B(x)} R(x,y)\right\|\right) \to \left\|\sum_{f:\Pi x:A.B(x)} \prod_{x:A} R(x, f(x))\right\|$$

W naszej nowej notacji logicznej możemy to zapisać tak:

$$(\forall x : A.\exists y : B(x).R(x,y)) \implies \exists f : (\Pi x : A.B(x)).\forall x : A.R(x, f(x))$$

Aksjomat wyboru 3 - interpretacja

- Dzięki zastosowaniu trunkacji w przesłance wiemy, że dla $x : A$ istnieje jakieś $y : B(x)$, ale nie wiemy jakie. To sprawia, że żaden wybór nie został jeszcze dokonany.
- Podobnie dzięki zastosowaniu trunkacji w konkluzji aksjomat mówi, że istnieje jakaś funkcja, ale nie wiadomo jaka. To sprawia, że żaden wybór nie został z góry dokonany.
- Składając to do kupy widzimy, że tym razem Aksjomat Wyboru faktycznie coś wybiera.
- Jednak to nie wszystko. Zwróć uwagę, że ten aksjomat dotyczy jedynie zbiorów, rodzin zbiorów oraz relacji, w przeciwieństwie do poprzedniego, który dotyczył typów i rodzin typów.
- Powód tego jest prosty (ale skomplikowany).

Aksjomat wyboru 4 - zaskoczenie

Lemat 3.8.5

Istnieje taki typ A i rodzina zbiorów $B : A \to \mathcal{U}$, że Aksjomat Wyboru nie zachodzi.

Dowód jest dość skomplikowany - po więcej zajrzyj do książki.

W telefgraficznym skrócie: wszystkiemu winne są nietrywialne ścieżki. Nie możemy beztrosko wybierać punktów, nie zwracając na nie uwagi. Tak jak remedium w przypadku prawa wyłączonego środka było ograniczenie się do zdań, tak w przypadku Aksjomatu Wyboru jest nim ograniczenie się do zbiorów.

Ostateczne wątpliwości

- W ramach ostatecznych wątpliwości mogą nam się nasunąć różne pytania.
- A co jeżeli Aksjomat Uniwalencji jest sprzeczny? Otóż nie jest. A jeżeli nie jest konstruktywny? Otóż jest. Świadkiem nam Kubiczna Teoria Typów. Jest to teoria typów, w której można programować za pomocą n-wymiarowych sześcianów: `https://ncatlab.org/nlab/show/cubical+type+theory`
- Jaka jest ogólna składnia/schemat definiowania wyższych typów induktywnych? Co wolno a czego nie? Otóż nie do końca wiadomo, choć czynione są postępy: `http://drops.dagstuhl.de/opus/volltexte/2018/9190/`
- Jaka jest semantyka wyższych typów induktywnych? Jakaś jest: `https://arxiv.org/abs/1705.07088`, ale za dużo nie wiadomo.
- Będzie z tej mąki chleb? Tak.

Podsumowanie

Homotopiczna teoria typów daje nam:

- Wyobraźnię w postaci interpretacji homotopicznej.
- Aksjomat Uniwalencji łatający filozoficzny smród i dający fajne sposoby rozumowania.
- Wyższe typy induktywne, które dają nam możliwość konstruktywnego rozwiązania masy problemów.
- Znacznie precyzyjniejszą analizę logiki, zarówno klasycznej jak i konstruktywnej.
- Możliwość formalizowania matematyki z prawdziwego zdarzenia, w tym teorii homotopii za pomocą metod syntetycznych.
- ... i wiele więcej.

Żart 1

Co łączy samobójcę, programistę imperatywnego, autobusiarza i homotopicznoteoriotypowca?

Żart 2

Lubią pętle.

Zadania

Zadanie 1: wykonaj wszystkie rzeczy oznaczone na slajdach jako ćwiczenie (poza ostatnią).

Zadanie 2: wykonaj ostatnią rzecz oznaczoną na slajdach jako ćwiczenie.

Bibliografia 1

- Podstawowym źródłem wiedzy jest książka https://homotopytypetheory.org/book/, zwana potocznie HoTTBookiem.
- Jakaś prezentacja: http://www.cs.nott.ac.uk/~psztxa/talks/edinburgh-13.pdf
- Przyjazne przykłady wyższych typów induktywnych dla programistów: http://www.cs.ru.nl/~herman/PUBS/HIT-programming.pdf

Bibliografia 2

- Filozoficzne rozważania na temat HoTT jako podstaw matematyki: https://www.researchgate.net/publication/280671356_Does_Homotopy_Type_Theory_Provide_a_Foundation_for_Mathematics
- Praca o podobnej tematyce: https://arxiv.org/pdf/1601.05035.pdf
- Ciekawa praca zachaczająca o filozofię matematyki, prezentująca coś, co można by nazwać pluralizmem topologicznym: https://arxiv.org/abs/1703.03007