Volume VI: Philosophy

Introduction to Formal Philosophy

Namdak Tonpa

2026 · Groupoid Infinity
VI

Зміст

1 Свідомість

Формальна філософія якщо й має чимось займатися, то лише кодуванням різних моделей свідомості (різного ступеня фрічества, чому фрічество взагалі допускається скажу пізніше). Не здаватимуся політкоректним, більшість філософів, які вивчаються в контексті свого предмета, я вважаю душевно хворими людьми (і не бачу нічого в цьому поганого). Якщо ви можете уявити будь-яку модель на MLTT і більш сучасних типових системах, і можете розповісти, як ця модель кодує якийсь феномен, ви вже стаєте личинкою формального міні-філософа. Зазвичай що складніша модель, то більше вписувалося її як твір розглядатимуть інші формальні художники.

Багато хто критично ставиться до сучасних моделей AI, тому що вони надто прості, щоб повірити, що там може зародитись якась автономна свідомість. Щоб дати можливість системі зародитися і здобути якусь свободу, ми повинні надати цій системі якийсь простір, і глибина цього простору повинна перебувати в мові цієї системи. У MLTT таку глибину, яка закриває навіть гьоделівські питання, надає ієрархія всесвітів. Причому вона виникає незалежно від наших примх, типи зобов'язані десь перебувати, тому ми в мові виділяємо контейнер для типів U_n, і кажемо, що цей тип містить усі типи (наочно це демонструє індукція-рекурсія, в інших випадках потрібно вірити тайпчекеру, що всі формейшин рули ядра живуть в U_n. Природно постає питання межі послідовності U_i прагне до U_ω. А далі послідовність недоступних кардиналів $\mathsf{U}_\omega : \mathsf{U}_{\omega+1}$. Всесвіт Махло є щось на зразок такої згортки цієї послідовності. Така глибина дає певний спокій, що глибшого простору для мови ми не запропонуємо для нашої моделі свідомості, тому що ми просто не знаємо про це нічого. Інше відображення цієї глибини можна знайти в теорії інфініті категорій, теорії інфініті топосів та їх фізичним моделям ізоморфізмів, різних версіям теорії струн. Подих такого простору типів з відкритим дном і контрактибл типом у вершині конуса — це та мандала де знаходяться всі малюнки всіх формальних філософів.

Тепер про інформаційний тракт свідомості на нижніх рівнях, які вже можна помацати у вигляді AI. Якщо припустити, що ландшафт моделей всіх можливих мереж описується різними видами комплексів (симпліціальними, клітинними), а їх інваріанти задаються гомологіями і гомотопічними типами, то така глибина теж цілком сумісна з поточними методами, а гомологічна алгебра вже застосовується в мережевій інженерії. Такий простір вимагає застосування методів топології алгебри і створює нову глибину де може зародитися мислення. Якщо стисло, то тут ідея така, що є якийсь генератор свідомості, який постійно будує сам різні топології мереж, сам їх навчає, і сам веде реєстр цього поля мереж, яке вбудовується в сам простір, як типи вбудовуються у всесвіт.

Третя більша частина, яка зараз відсутня в моделях свідомості, це фізична комутативна математика, виняткові і класичні групи Лі, просто тому, що ми себе виявили в цьому просторі, і очевидно, це якось пов'язано з мисленням. Як і що тут вбудовувати мені незрозуміло, але здається, що тут спливає щось на кшталт чакр або рівнів буття якоїсь мета-істоти, яка задає уречевлення всієї мандали у видимому нам світі, який інкрустований іншими різноманіттями як прикрасами основного простору.

2 Розшарування Хопфа

Візьмемо людський мозок. Нехай кожен нейрон — це вершина симпліціального комплексу (тріангульованого простору), навіть не комплексу, а симпліціального множини (теж що і комплекс, але з інформацією про орієнтацію, тут орієнтація — це категорна дуальність, перевертання n-стрілок), так як різниця потенціалів передається по дендрону від нейрона до нейрона у певному напрямку. Беремо радіоактивні ізотопи (найкраще взяти помічений ЛСД), пропускаємо через енцефалічний бар'єр і будуємо симпліційний комплекс. Чому ми взяли симпліційні множини, а не спрямовані графи, тому що групи нейронів утворюють згустки, в яких енергія зв'язку настільки сильна, що можна говорити про компактність клітин на n-рівнях. Якщо спробувати згенерувати рендомний мозок, з урахуванням статистичних даних, ми отримаємо симпліційний комплекс розмірності 3 (три) загалом. Якщо ж ми візьмемо очікування розмірності за реальними конкретними мізками, ми отримаємо кількість вимірів комплексу рівним приблизно 8 (восьми). Такі многовиди відомі як Калабі Яу, а простір в якому живуть всі фізичні симетрії стандартної моделі міститься в групі E8, яка в гомотопічній інтерпритації розрізається на 4 розрашування Хопфа. Моделювання нейромереж багатовимірними комплексами це маст хев сучасного теоретичного AI, як у симуляції (медичній), так і в прикладному моделюванні. У комп'ютер віжині вже, до речі. Критерій Сохацького: якщо комплекс нейромережі має розмірність менше 8, чекати на самозароджене AI там безглуздо. У процесі навчання ми зможемо спостерігати зміну комплексу у реальному часі та можливе навіть підвищення розмірностей.

Взагалі, теорія симпліціальних множин має багато ізоморфізмів: теорія інфініті категорій (відразу кілька моделей, квазікатегорії, про них піде мова в наступних постах), теорія струн, і т.д. Забезпечення відкритого нескінченного глобулярного (n-розмірного) когерентного (аналог композиції на n-рівнях) простору — чиста геометрія. В геометрію ми виходимо завжди, якщо щось узагальнюємо на нескінченності. Наприклад у теоретичній інформатиці, а саме теорії типів — у нас є два розділи: теорія типів та їх поліноміальні функтори (звичайні індуктивні типи) з одного боку, і, з іншого боку — гомотопічна теорія типів (де є глобулярні рівності, завдяки викинутому ета-правилу Id типа) та їх вищих індуктивних типів (або CW-комплексів, тому що будь-який CW-комплекс можна виразити через HIT і навпаки).

3 Формалізація буддизму

Зараз я дам вам відчути смак математичної формальної філософії по-справжньому! А то вам може здатися, що це канал з формальної математики, а не формальної філософії. Я ж вважаю, що якщо формальна філософія не спирається на формальну математику, то гріш ціна такій формальній філософії.

```
module buddhism where
import path
```

Сьогодні ми будемо формалізувати поняття недвоїстості в буддизмі, яке пов'язане одразу з багатьма концепціями на рівнях Сутри, Тантри та Дзогчена: поняттям взаємозалежного виникнення та поняттям порожнечі всіх феноменів (Сутра Праджняпараміти). Класичний приклад із розчленовуванням тіла ставить питання, коли тіло перестає бути людиною-істотою, якщо від нього почати відрубувати шматки м'яса (ми буддисти любимо і лілеєм такі уявні образи-експерименти) або іншими словами, щоб відрізнити тіло від не-тіла, нам потрібен двомісний предикат (родина типів), функція, яка може ідентифікувати конректні два еклемпляри тіла. Практично йдеться про ідентифікацію двох об'єктів, тобто про звичайний тип-рівность Мартіна-Льофа.

За фреймворк візьмемо концепти Готтлоба Фреге, згідно з визначенням, концепт - це предикат над об'єктом або, іншими словами, Пі-тип Мартіна-Льофа, індексований тип, сім'я типів, тривіальне розшарування тощо. Де об'єкт x з o належить концепту, якщо сам концепт, параметризований цим об'єктом, населений p(o) : U (де p : concept o).

```
concept (o: U): U
   = o -> U
```

Концепт p повинен надавати приклад чи контрприклад розрізнення, тобто щоб визначити тіло це чи не тіло ще, поки ми його розчленовуємо, нам потрібно як мінімум два шматки: тіло і не тіло як приклади ідентифікації. Таким чином, недвоїстість може бути представлена як рівність між усіма розшаруваннями (предекатами над об'єктами).

```
nondual (o: U) (p: concept o): U
  = (x y: o) -> Path U (p x) (p y)
```

Отже, недвоїстість усуває різницю між прикладами і контрприкладами на примордіальному рівні мандали MLTT, тобто ідентифікує всі концепти. Сама ж ідентифікація класів об'єктів, які належать різним концептам — це умова, що стискає всі об'єкти в точку, або стягуваний простір, вершина конуса мандали MLTT, або, іншими словами, порожнеча всіх феноменів виражена як тип логічної одиниці, який містить лише один елемент.

```
allpaths (o: U): U
  = (x y: o) -> Path o x y
```

Формулювання буддійської теореми недвоїстості, яка поширюється всі типи учнів (тупих, середніх і тямущих), може звучати так: недвоїстість концепту є спосіб ідентифікації його об'єктів. Сформулюємо цю саму теорему в інший бік: спосіб ідентифікації об'єктів задає предикат неподвоїстості концептів. Туди - ((p: concept o) -> nondual o p) -> allpaths o, Сюди - allpaths o -> ((p: concept o) -> nondual o p). І доведемо її! Як видно з сигнатур нам лише треба побудувати функцію транспорту між двома просторами шляхів: (p x) =U (p y) і x =o y. Скористаємося приведенням шляху до стрілки (coerce) та конгруентності (cong) з базової бібліотеки.

```
forward (o:U): ((p: concept o) -> nondual o p) -> allpaths o
  = \(nd: (p: concept o) -> nondual o p) (a b: o) ->
      coerce (Path o a a) (Path o a b) (nd (\(z:o)->Path o a z) a b) (refl o a)

 backward (o:U): allpaths o -> ((p: concept o) -> nondual o p)
  = \(all: allpaths o)(p: concept o)(x y: o) -> cong o U p x y (all x y)
```

Як бачите, теоремка про порожнечу всіх феноменів вийшла на кілька рядків, які демонструють: 1) основи формальної філософії та швидке занурення в область математичної філософії; 2) гарний приклад до першого розділу HoTT на простір шляхів та модуль path.

4 Хроматична теорія гомотопій

Поговоримо про хроматичну теорію гомотопій. Я маю на увазі, що ви трохи знайомі з терією категорій і топологією.

Отже, передусім, припустимо, ви вірите в те, що намагатися класифікувати топологічні простори гомотопічним типом не марна витівка. Це практично неможливо, проте ціль ця шляхетна. Щоб полегшити собі завдання, ми будемо розглядати базові простори, які можуть бути створені приєднанням клітин. Вони іноді називаються CW-комплексами чи клітинними комплексами. Коли я говорю "склеювання клітин я маю на увазі конструювання пушауту для конуса $D^n \leftarrow S^{n-1} \rightarrow X$, де X — деякий простір, D^n — n-диск, а S^{n-1} — його межа. Буквально - склеювання диска на його кордоні.

Виявляється, що будь-який гарний простір (компактно згенерований хаусдорфово) може бути побудований таким чином, починаючи з простих точок, хоча вам, можливо, доведеться приєднати нескінченну кількість осередків. Іншими словами, якщо єдиними будівельними блоками, які ми маємо, є осередки, такі як D^n, ми можемо, аж до гомотопії, створювати найкрасивіші топологічні простори (наприклад, усі ті, що з'являються у таких додатках, як диференціальна геометрія).

Отже, щоб зрозуміти, які нові простори ми можемо побудувати з існуючого простору X приєднанням осередків, достатньо знати всі способи, якими сфера може безперервно відображатися сама в собі (бо ми прикріплюємо осередки з використанням сфер відображення). Ви, напевно, вже знаєте, що безліч способів відображення сфери в інший топологічний простір X називається гомотопічними групами X. Тому, якщо ми можемо обчислити гомотопічні групи, ми можемо класифікувати (гарні) простори з точністю до гомотопій. Звичайно, ви також можете знати, що обчислювати гомотопічні групи топологічних просторів дійсно дуже, дуже складно. Зокрема, найпростіше питання, яке ви можете поставити — це які простори можна отримати склейками будь-яких сфер (S^n до S^m для будь-яких n і m). Це буде набір груп із двома індексами n і m, один для розмірності сфери домену та один для розмірності сфери кодомена. Знання цього могло стати непоганим стартом.

Погані новини: ми не знаємо ці групи, і ніколи можливо не дізнаємось. Ми знаємо безліч їх, і ми хороші в обчисленні груп для фіксованих n і m, якщо добре постараємося, але несхоже щоб там був певний очевидний патерн для генерації всіх таких груп. Ну добре, ми можемо принаймні спробувати, і сподіватися, що ми побачимо прикольні штуки дорогою вирішення цього завдання. Коли це все починалося, не було відомо, наскільки це буде складним (сходить до Пуанкаре), тому ми просунулися досить далеко, перш ніж зрозуміли, що справа погана. Зрештою, хроматична гомотопічна теорія є спробою розбити вищеописані гомотопічні групи сфер на будівельні блоки, які легше зрозуміти і з якими легше працювати. Слово "хроматичний"відноситься до складових довжин хвиль, в які "розкладається"біле світло.

Сподіваюся, ви знаєте, що для сфери S^n існує відображення "ступеня p яке обертає сферу S^n навколо себе p разів. Уявімо це відображення як $p : S^n \to S^n$. Це в точності те p, що ви побачите, коли згадаєте, що n-та гомотопічна група сфери S^n — це цілі числа $\mathbb{Z}$. Зауважте, що це відображення генерує ціле сімейство відображень $S^n \to S^n$, задане ітеруванням p . Тобто. $p^k : S^n \to S^n$ для будь-якого k. Таким чином у нас обчислилися деякі гомотопічні групи, але вони не такі вже й цікаві. Одну річ, яку ми можемо зробити — це причепити клітину уздовж цього відображення (або будь-якої його ітерації), щоб отримати новий простір, який я запишу як $V(0)$, або $S^n \bmod p$. Зауважте, що пушаут, який визначає причеплену клітину, зробив вихідне відображення p гомотопним нулю (або гомотопічно тривіальним). Зауважте, що $V(0)$ це лише (n+1)-сфера приклеєна до n-сфери, існує включення нижньої сфери S^n в $V(0)$, назвемо це i, і відображення $V(0) \to S^{n+1}$, яке стягує нижню сферу, назвемо це q. Таким чином, якби я мав інше відображення $f : \Sigma V(0) \to V(0)$, де Σ — надбудова (суспензія), тоді я міг би зліва закомпозити це з i, а потім праворуч закомпозити з q

$(i \cdot f \cdot q)$, щоб отримати нове відображення $S^{n+1} \to S^{n+1}$, яке було б свого роду породженим за допомогою f. Зверніть увагу, що роль, яку відіграє надбудова, полягає у збільшенні розмірності нижньої сфери $V(0)$. Інакше ми мали б відображення $S^{n} \to S^{n+1}$, і будь-яке таке відображення було б гомотопічно тривіальним (знецінюючи наші дії).

Причина, через яку ми це все робимо, полягає в тому, щоб просто знайти ДЕЯКІ елементи гомотопічних груп сфер. Загалом, нам знадобилося чимало часу, щоб знайти БУДЬ-ЯКІ елементи гомотопічних груп сфер, а тим більше спробувати обчислити ВСІ їх. Таким чином, це була велика справа, коли люди як Адамс і Тода, змогли показати що, ТАК, є відображення $\Sigma V(0) \to V(0)$ (тут упускаються деталі, насправді вам потрібна більше ніж одна надбудова). Більше того, це відображення, назвемо його $A : \Sigma V(0) \to V(0)$, може бути итерировано нескінченне число разів, не стаючи при цьому гомотопічно тривіальним. І щоразу, коли ми ітеруємо, ми збільшуємо розмірність. Отже, у нас є вся родина відображень сфер, що виходять з (ітеруючих) A. Під ітеруванням, я маю на увазі, що у мене є відображення $A : \Sigma V(0) \to V(0)$, тому я можу отримати відображення $\Sigma A : \Sigma\Sigma V(0) \to \Sigma V(0)$, а потім інше відображення $\Sigma\Sigma A : \Sigma\Sigma\Sigma V(0) \to \Sigma\Sigma V(0)$, і так до нескінченності, де розмір доменної сфери буде ставати все більше і більше. Якщо це спрацювало одного разу, то чому б не зробити це знову? Так само, як ми затюніли p раніше, давайте візьмемо коядро відображення A (яке насправді називається корозшаровуванням). Іншими словами, стягуйте все, що потрапляє в A, даючи нам точну послідовність просторів $V(0) \to V(0) \to V(0)/A$.

Давайте перейменуємо $V(0)/A = V(1)$. Тепер вам просто потрібно повірити, що $V(1)$ виходить шляхом приєднання осередків (змішуючи склеювання, які ми зробили для побудови $V(0)$), і тому все ще існують відображення з нижньої сфери i: $S^{k} \to V(1)$ і фактор групи до верхньої сфери $q : V(1) \to S^{k}$. Сміт показав, що існує інше відображення $B : \Sigma V(1) \to V(1)$, яке ми можемо ітерувати стільки разів, скільки ми захочемо, і ми отримаємо ІНШЕ велике сімейство відображень між сферами. Отже, у нас є два великі сімейства відображень сфер (всіх можливих вимірів!), Я назву їхню A-родину і B-сімейство. Виявляється, якщо ви знову "затюніте"B, ви отримаєте новий простір, назвемо його $V(2)$, і ви можете повторити цей процес ще раз. І справді, існує карта $C : \Sigma V(2) \to V(2)$, що дає нам інше сімейство, я назву його C-сімейством гомотопічних груп сфер. Тому тут виникають два природні питання: 1) чи можна повторювати це нескінченно? 2) чи отримаємо ми всі гомотопічні групи сфер?

У певному сенсі перлина хроматичної гомотопічної теорії — це позитивна відповідь на ці два питання (знову ж таки, упускаючи багато деталей). Це насправді є зміст теорем Нільпотентності та Періодичності Девіннаца, Хопкінса та Сміта. У своїй основі "хроматична"ідея полягає в тому, що ці сімейства, які ми отримали A, B і C, є початком стратифікації гомотопічних груп сфер, і існує нескінченний список цих сімейств. Таким чином, це різновид глобальної структурної теореми для цих розсипаних та заплутаних груп. Ми ще не знаємо їх усіх, але знаємо, як вони організовані. І враховуючи, наскільки вони складні, це справді велика гра!

Ще раз трохи про це все, що є ідеєю простих ідеалів, локалізації та K-теорій Морави. Також тут буде трохи алгебраїчної геометрії. Ви можете пам'ятати, що якщо ви хочете дізнатися про пучку на $\mathsf{Spec}(\mathbb{Z})$, достатньо знати про нього в кожному простому числі, тобто кожної "локалізації"цілих чисел $\mathbb{Z}$. Таким чином, хроматична стратифікація говорить про те, що на відміну від алгебраїчних різноманітності, інформація про які може бути зібрана з цілих простих чисел, інформація про комплекси CW може бути зібрана з цих так званих хроматичних шарів. Насправді, ці хроматичні шари не просто стратифікують гомотопічні групи сфер, вони стратифікують всю категорію кінцевих CW-комплексів. Оскільки схема може бути розбита на її p-локальні частини для кожного простого p, простір може бути розбитий на його хроматичні частини. Це зміст Thick Subcategory Theorem, яка є частиною теореми про Нільпотентність та Періодичність (наслідком). У ній говориться, що, знаходячи цю стратифікацію лише з сфер, ми фактично стратифікували всі кінцеві комплекси клітин. Тут під кінцевим клітинним комплексом, я маю на увазі простір, побудований шляхом виконання кінцевого числа приклеювань осередків (починаючи з порожнечі). Більше того, для тих з вас, хто знає, що це означає, існують теорії когомологій, які говорять про те, на якому СЛО є цей простір. І ці теорії когомологій є так званими K-теоріями Морави $\mathsf{K}(\mathfrak{n})$ для натуральних n. Тому кінцевий клітинний комплекс має "тип який є натуральним числом, і це число говорить мені, до якого хроматичного шару належить кінцевий клітинний комплекс. Це може бути обчислено, тому що це просто перший n, для якого $\mathsf{K}(\mathfrak{n})$ когомології вашого простору відмінні від нуля.

$\mathsf{K}(0)$ відповідає ступеню відображення p, $\mathsf{K}(1)$ відповідає відображенню A, $\mathsf{K}(2)$ відповідає відображенню B і т.д. Ці відповідності я не розглядатиму тут детально. Але, в будь-якому випадку, з'являється така, дійсно красива, картина, де ці $\mathsf{K}(\mathfrak{n})$ говорять нам про те, як розділити категорію кінцевих клітинних комплексів на дрібніші, зручніші шари. Фактично, подібно до того, як ви можете локалізувати схему або пучок для простого p, ви можете локалізувати будь-який простір $\mathsf{K}(\mathfrak{n})$ для будь-якого n. І якщо ви знаєте його $\mathsf{K}(\mathfrak{n})$ локалізацію для кожного n, тоді ви можете зібрати цей простір по шматках, і ви будете знати все про цей простір. Це невелика частина того, що люди мають на увазі, коли кажуть, що теорія Морави — це "прості числа"(стабільної) гомотопічної категорії. По суті, вони — "місця в яких ми можемо локалізувати, щоб отримати локальну інформацію, яку ми хочемо зібрати в глобальну інформацію.

Я повинен додати, що в основному все, що я знаю про це, я дізнався з "помаранчевої книги або "Нільпотентність та періодичність у Стабільній теорії гомотопії"Дугласа Равенела. Це справді красива, приголомшлива книга, і я всіляко її рекомендую.

Джонатан Бірдслей

5 Геометрія в модальній HoTT

Подяка за підтримку

Як завжди перед початком звітом по конференції хочу висловити подяку усім, хто підтримує мене на тернистому шляху докторантури. В першу чергу це моя родина та друзі, які надихають мене, а також представники академії які допомогають мені у цьому. Окрім моїх покровителів з КПІ, перш за все — це звичайно усі спікери та учасники конференції "Геометрія в модальнії Гомотопічній теорії типів". Також хотів би подякувати усім моїм клієнтам, які з розумінням ставляться до моєї математичої діяльності та не відволікають мене по дрібницям.

Передісторія

Як ви знаєте конструктивна теорія типів HoTT відкриває дорогу в CW-комплекси та алгебраїчну топологію, однак значний пласт проблем вона не вирішує, наприклад теоерема Брауера про нерухому точку не виводиться у цій теорії, те саме стосується таких речей як етальні відображення, на яких будується фундамент сучасної диференціальної геометрії. Модальна теорія HoTT у пов'язаних топосах зародилася як продукт дослідження Урсом Шрайбером Картанової супергеометрії у застосуванні до сучасної теоретичної фізики та було детально розроблена як розширення HoTT Майком Шульманом (cohesivett). Після дисертації Фелікса Черубіні стало зрозуміло, що HoTT у зв'язаних топосах може вирішити останні проблеми формалізації математики і ця конференція перший івент цього штибу.

Еґберт Райке

Еґберт захистив свій мастер тезис у Андрія Бауера, а дисертацію у Стів Еводі. Але найбільше Еґберт вплинув на мене своїм курсом по HoTT. Усього є декілька курсів: по-перше це сам підручник з HoTT, далі курс Валерія Ісаєва, курс Роберта Харпера (15-819), та один з найбільш просвітлюючих курсів це HoTT курс Еґберта, про який я писав на каналі Групоїд Інфініті. Його курс найбільше вплинув на мій власний курс HoTT.

Також Еґберт відомий тим, що у співробітництві з Ульріком Бушхольцом формалізував квартерніонні фібрації Хопфа. На цьому воркшопі Еґберт представив зразу три доповіді. Перша було про рефлексивні підсвествіти та модальності, де показав, що транкейшин рівні гомотопічних типів є рефлексивними підвсесвітами, їх універсальні властивості, необхідні умови доступності для підвсесвітів, еквівалентність визначень локальних відображень, локальних просторів, та замкненість підвсесвітів відносно еквівалентностей. Друга доповідь була присвячена відокремленим просторам (простір L-відокремлений, якщо усі його id-типи L-локальні, де L — рефлективний підвсесвіт). Тут розлядалось чи завжди підвсесвіт L-відокремлених типів є рефлексивним підвсесвітом. Третя доповідь була про

Модальний спуск та рефлексивні системи факторизації, де були дані визначенння ортогональних систем факторизації (OFS), стабільних OFS, та n-етальні відображення.

Майк Шульман

Переоцінити вклад Майкла Шульмана в розвиток HoTT важко. Фактично він основним автором ncatlab Cafe, співзасновником ncatlab, та головним контрибутором в cohesivett, теорію яка відкрила двері в математику нескінченних околів, та інших модальностей, яка були ним розроблена з подачі Урса Шрайбера. На ції конференції Майк Шульман представив дві доповіді. Перша була присвячена основам теорії звязанних топосів (про яку вперше можно прочитати у Вільяма Лавіра). В першій доповіді були дані основні визначення комонадичної флет-модальносты (бемоль) згідно з принципами MLTT (формації, інтро, елімінатори, бета та ета рівності), яка за якою власне і схована корекурсія обчислення нескінченно малих величин, або більше відомих в математиці як епсилон-околи. Це база cogesivett, розробка якої розпочалася в далекому 2011 році. Також було обговорення імплементації бемоль модальносты в Agda (flat-бренч на гітхабі). Друга доповідь була присвячена семантиці вищих молальностей, та за допомогою бемоль-модальносты була накінець-то конструктивно доведена теорема Брауера про нерухому точку (яка як відомо не доводиться у чистій HoTT), чим було закрито кон'юнктуру Еводі, у залі стояли гучні аплодисменти. Раджу усім послухати цю доповідь, хто скептично ставився до конструктивної математики.

Урс Шрайбер

Урс один з засновників сучасної математичної енциклопедії ncatlab, якою користуються усі сучасні математики, не тільки в області інфініті категорій, HoTT, а також прикладної та теоретичної фізики, яка для Урса є головним мотиватором. Вперше ідея запропонувати cohesive топоси Лавіра для моделювання вищих геометорій та логік виходила саме від Урса, а Майк Шульман виконова усю необхідну роботу по формалізації (cohesivett). Як бачимо колаборація Майка та Урса виходять далеко за межі адмінітративної роботи по управлінню математичної екнциклопедії ncatlab, якою ми з вами зачитуємося до 5-ї години ранку.

Додовання модальностей як системи операторів до HoTT дає змогу дуже елегатно доводити теореми вищої геметрії, диференціальної топології, диференціальної гемометрії, супергеметрії. В доповіді Урса показані засади Картанової геометрії, формальної Картанової геометрії та Картанової Супергеометрії. Головна мотивація для Урса — це побудова формальної М-теорії (спільна робота з Хішамом Саті в Нью-Йоркському Університеті в Абу Дабі). Урс побудував вежу модальностей які хитро вилаштовуються в діагональ спряжень, і чи має ця вежа кінець відкрите питання в новоспеченій модальній HoTT.

Нижній рядок — Empty та Unit, другий рядок модальностей застосовується в диференціальній топології (ретопологізація та топологізація) — шейп модаліті, флет (бемоль) модаліті, та шарп (дієз) моділіьті, третій рядок застосовується у диференціальній геометрії (Im та Re модальності), та верній рядок — фізика (бозонні та ферміонні модальності), далі вежа насичується та стабілізується (начебто?). Синім кольором зображені мотивні локалізації афінного відрізка.

Як на мене це сама езотерична теорія сучасної формальної математики та теоретичної фізики яка дає безмежне натхнення по формальнізації будь яких теорії (M-теорія, теорія струн, супергеометрія, інші версії гомотопічних теорій). Без сумніву Урса можна назвати батьком модальної гомотопічної теорії типів, а також батьком ncatlab!

Фелікс Черубіні

Фелік Черубіні перший учень Урса Шрайбера який детально розробив у своїй дисертації Im модальності, які застосовуються для моделювання етальних відображень, які у свою чергу дозволяють доводити теореми про нескінченнін околи (диференціальна геометрія). Сама дисертація Фелікса відрила у мене друге дихання та породила нову лінію досліджень. Власне цей воркшоп присвячений геометрій відбувся завдяки зусиллям Фелікса, як головного організатора заходу.

Ульрік Башхольтц

У Ульріка багато регалій, але якби мене попросили його описати одним досягненням, то я би сказав, що Ульрік єдина людина у світі яка змогла формалізувати кватерніонне розшарування Хопфа (мова Lean). Також я замовив у Ульріка code review інтерналізації кубічної теорії у мову Lean), так як Ульрік адепт Lean.

Доповідь Ульріка була присвячена некомутативній теорії когомологій, а саме пучкам (gerbe, як інструмент гокомологій степені 2), крученням (twist), лєнтам (band). І усе це в модальній HoTT. Дивіться його підручник з симетрій Symmetry Book[1].

Пітер Арндт

Пітер Арндт представив абстраткну мотивну теорія гомотопій — яка є одночасно його PhD тезисом. Доповідь почалася зі зведеної теорії когомологій як функтора з оберненої категорії топологічних просторів з точками в абелеву категорію: Top*op -> AbZ. Далі були розглянуті схеми на базою K: K-Alg -> Set. Потім ми розглянули глядкі схеми та мотивні простори з топологією Нісневича/Заріського, мотивні спектри та мотиви HZ-Mod (шестикутник). Потім ми замінили мотивні простори на представимі інфініті декартово

[1] `https://github.com/UniMath/SymmetryBook`

замкнені категорії та визначили таким чином цілком абстрактну мотивну теорію гомотопій. Були розглянуть стабільні та нестабільні результати та побудови та наведені достатні приклади. Показана можливість пабудови раціонального оператора Адамса.

Спонтанне інтерв'ю з авторами кубіків

Нагадаю, що кубіками я називаю системи доведення теорем, які побудовані на базі кубічної теорії типів, єдиної теорії типів, у якій можно довести аксіому унівалентності Воєводського. 5 із шести кубічних тайпчекері були написані в CMU цими хлопцями (три версії на мові хаскель: cubicaltt, cubicaltt/hcomptrans, yacctt, одна на Standard ML — RedPRL, та одна на OCaml). Не взяти інтерв'ю в авторів цих пруверів було би великою необачністю. Я зробив презентацію нашої бібліотеки Groupoid Infinity — усі були у захваті.

Андерс Мортберг

Не знажаючи на те, що cubicaltt більше не розроблюється, Андерс жартує, що він ідеальний, тому нема далі що покращувати. Наймінімальніший синтаксис кубічної теорії який поміщається на 12-рядках BNF нотації!

Карло Анджіулі, Джон Стерлінг

Джон Стерлінг пропрацював деякий час в індустрії програмного забепечення, тому його системи відрізняються дорослістю та сккладністю. Карло забезпечує математичну підтримку, але усі вони программісти та математики, важно виділити що головніше!

Еван Кавалло

Еван був представлений мені як спеціаліст з вищих індуктивних типів (CW-комплексів), або HIT-чарівник! Останні публікації стосовно теорії HIT та формалізація тайпчекінга HIT в декартовій кубічній теорії — заслуги Евана.

Стів Еводі

Стіва Еводі можна назвати батьком гомотопічної теорії типів (HoTT), саме на запрошення Стіва Володимир Воєводський почав інтенсивно займатися HoTT, а також завдяки Стіву відбувся проект HoTT. Як хранитель та модератор Стів спостерігав за конференцією та був адвайзором багатьох людей присутніх на конференції.

6 Lean конференція

По-перше хочу подякувати усім, хто мене підтримує, надихає та прощає мені багато чого. Попасти на таку подію — це справжній подарунок та задоволення від усвідомлення, що коло однодумців більше, ніж ти очікував.

Хенк Барендрегт

Назвичайно був вражений теплим сердечним прийомом та щирим і відкритим інтерв'ю Хенка Барендрехта, перша частина якого опублікована в інстаграмі, а друга на ютубі та фейсбуці. Хенк Барендрехт відомий нам інженерам як автор лямбда кубу, цілої системи формальних моделей які поєднують та класифікують усі лямбда числення в залежності від різного набору чотирьох формул $\star : \star, \star : \square, \square : \square, \square : \star$. Фактично, усі типизовані мови програмування, включаючи PTS (CoC, System P_{ω}, або чиста система), яка є ядром усіх пруверів, потрапляють у лямбда куб. До цього Хенк Барендрехт також повністю формалізував та довів майже усі теореми про нетипизоване лямбда числення яке теж є основою багатьох мов з динамічною типизацією. Його український учень Андрій Полонський, який повернувся в Україну під час Майдану, довів спростування кон'юнктури Хенка Барендрехта, яке лягло в основу сучасних ідей оптимальної бета-редукції, яка була реалізована в роботах Майя Віктора (мова Kind з залежними індуктивними типами, написана на Rust, містить оптимальний евалуатор бета-редукцій та здійснює екстракт в GPU).

Хенк зараз досліджує свідомість за допомогою формальних методів та медитації, він є сертифікованим учителем Віпаш'яни в стилі Махасі, його учителями були Кобун Чіно Роші та Фра Меттавіхарі. Найближчі його ретріти в 2019 році в Греції та Італії. Хенк також пише статті на тему формальної філософії, формалізації буддійських систем та формалізації свідомості. Можливо варто спробувати написати огляд робіт Хенка для нашого каналу Формальної Філософії, якщо кількість підписників, скажімо, подвоїться.

Джеремі Авігад

Перша — Джеремі Авігада по факторизації поліноміальних функторів. В секції питань його запитали чи дійсно дійсні числа закодовані як фактор-типи коіндуктивних послідовностей цифр є найкрасивішою моделлю дійсних чисел і Джеремі відповів, що так.

```
class qpf (F : Type u → Type u) [functor F] :=
      (P : pfunctor.{u})
      (abs : Π {α}, P.apply α → F α)
      (repr : Π {α}, F α → P.apply α)
      (abs_repr :∀ {α} (x: F α), abs(reprx)=x)
      (abs_map : ∀ {α β} (f : α → β) (p : P.apply α),
                 abs (f <$> p) =f <$> abs p)
```

Рейд Бартон

Друга — Ріда Бартона по формалізації модель-категорій Квілена та структурі моделі Квілена на топологічному просторі. У цій роботі розглянутий трансфінітний випадок, проведена велика класифікація топологічних просторів (найбільша на моїй пам'яті), достатньо пророблена теорія категорій та фундаментальний групоїд, категорія корозшарувань, факторизація Брауна.

```
class model_category (M : Type u) extends category.{u v} M :=
      (complete : has_limits M)
      (cocomplete : has_colimits M)
      (W C F : morphism_class M)
      (h : is_model_category W C F)

structure is_model_category (W C F : morphism_class M) : Prop :=
          (weq : is_weak_equivalences W)
          (caf : is_wfs C (F ∩ W))
          (acf : is_wfs (C ∩ W) F)

def Meas : Type (u+1) := bundled measurable_space
def Top  : Type (u+1) := bundled topological_space

def Borel : Top ⇒ Meas :=
    concrete_functor @measure_theory.borel
                     @measure_theory.measurable_of_continuous
```

Джесі Хан

Третя — доведення незалежності континуум гіпотези від Джесі Хана. Джесі Хана також цікавить транспорт з топоса в класичну топологію.

```
theorem independence_of_CH :
        (¬ ZFC ⊢' continuum_hypothesis)
     ∧ (¬ ZFC ⊢' ~ continuum_hypothesis) :=
begin
  have := CH_consistent,
  have := neg_CH_consistent, repeat{auto_cases},
  apply @independence_of_exhibit_models L_ZFC ZFC
        ZFC_consistent continuum_hypothesis,
  show Model ZFC,
  exact this_w_1,
  show Model ZFC,
  exact this_w,
  repeat{assumption}
end
```

7 Модельні категорії Квілена

PhD Деніела Квілена була присвячена диференціальним рівнянням, але відразу після цього він перевівся в MIT і почав працювати в алгебраїній топології, під впливом Дена Кана. Через три роки він видає Шпрінгеровські лекції з математики "Гомотопічна алгебра яка назавжди трансформувала алегбраїчну топологію від вивчення топологічних просторів з точністю до гомотопій до загального інструменту, що застосовується в інших галузях математики.

Модельні категорії вперше були успішно застосовані Воєводським на підтвердження кон'юнктури Мілнора (для 2) і потім мотивної кон'юнктури Блоха-Като (для n). Для доказу для 2 була побудована зручна гомотопічна стабільна категорія узагальнених схем. Інфініті категорії Джояля, досить добре досліджені Лур'є, є прямим узагальненням модельних категорій.

До часу, коли Квіллен написав "Гомотопічну алгебру вже було деяке уявлення про те, як має виглядати теорія гомотопій. Починаємо ми з категорії C та колекції морфізмів W — слабкими еквівалентностями. Завдання вправи інвертувати W морфізму щоб отримати гомотопічну категорію. Хотілося б мати спосіб, щоб можна було конструтувати похідні функтори. Для топологічного простору X, його апроксимації LX і слабкої еквівалентності LX -> X це означає, що ми повинні замінити X на LX. Це аналогічно до заміни модуля або ланцюгового комплексу на проективну резольвенту. Подвійним чином, для симпліційної множини K, Кан комплексу RK, і слабкої еквівлентності K -> RK ми повинні замінити K на RK. У цьому випадку це аналогічно до заміни ланцюгового комплексу ін'єктивною резольвентою.

```
modelStructure (C: category): U
  = (fibrations: fib C)
  * (cofibrations: cofib C)
  * (weakEqivalences: weak C)
  * unit
```

Таким чином Квілену потрібно було окрім поняття слабкої еквівалентності ще й поняття розшарованого (RK) та корозшарованого (LX) об'єктів. Ключовий інстайт з топології тут наступний, в неабелевих ситуаціях об'єкти не надають достатньої структури поняття точної послідовності. Тому стало зрозуміло, що для відновлення структури необхідно ще два класи морфізмів: розшарування та корозшарування на додаток до слабких еквівалентностей, яким ми повинні інвертувати для розбудови гомотопічної категорії. Природно ці три колекції морфізом повинні задовольняти набору умов, званих аксіомами модельних категорій: 1) наявність малих лімітів і колимітів, 2) правило 3-для-2, 3) правило ректрактів, 4) правило підйому, 5) правило факторизації.

Цікавою властивістю модельних категорій є те, що дуальні до них категорії перевертають розшарування та корозшарування, таким чином реалізуючи дуальність Екманна-Хілтона. Розшарування та корозшарування пов'язані, тому взаємовизначені. Корозшарування є морфізми, що мають властивість лівого гомотопічного підйому по відношенню до ациклічних розшарування і розшарування є морфизми, що мають властивість правого гомотопічного підйому по відношенню до ациклічних кофібрацій.

Основним застосуванням модельних категорій у роботі Квілена було присвячено категоріям топологічних просторів. Для топологічних просторів існує дві модельні категорії: Квілена (1967) та Строма (1972). Перша як розшарований використовує розшарування Серра, а як корозшаровування морфізму які мають лівий гомотопічний підйом по відношенню до ациклічних розшарування Серра, еквівалентно це ретракти відповідних CW-комплексів, а як слабка еквівалентність виступає слабка гомотопічна. Друга модель Строма як розшарування використовуються розшарування Гуревича, як корозшарування стандартні корозшаровування, і як слабка еквівалентність — сильна гомотопічна еквівалентність.

```
quillen67
  : modelStructure Top
  = ( serreFibrations ,
      retractsCW ,
      weakHomotopyEquivalence )
```

```
strom1972
  : modelStructure Top
  = ( hurewiczFibrations ,
      cofibrations ,
      strongHomotopyEquivalence )
```

Найпростіші модельні категорії можна побудувати для категорії множин, де кількість ізоморфних моделей зростає до дев'яти. Наведемо деякі конфігурації модельних категорій для категорії множин:

```
set0: modelStructure Set = (all,all,bijections)
set1: modelStructure Set = (bijections,all,all)
set2: modelStructure Set = (all,bijections,all)
set3: modelStructure Set = (surjections,injections,all)
set4: modelStructure Set = (injections,surjections,all)
```

У контексті модельних категорій визначаються сполучення Квілена, лівий і правий функтори Квілена, Квілен еквівалентності, лівий і правий похідні функтори, розширення Ріді (оскільки в загальному випадку ліміти та коліміти не існують у гомотопічних категоріях визначених на модельних категоріях, то модельні категорії Наприклад є категорії C, і Ріді категорії J то J -> C має всю необхідну структуру для існування гомотопічних (ко-)лімітів.

Для переходу від модельних категорій до інфініті категорій [або (∞,1)-категорій] необхідно перейти до категорій де морфізми утворюють не множини, а симпліційні множини. Потім можна переходити до локалізації.

```
simplicial
  : modelStructure sSet
  = ( kanComplexes ,
      monos ,
      simplicialBijections )
```

Але для нас, для програмістів найцікавішими є модельні категорії симпліціальних множин та модельні категорії кубічних множин, саме в цьому сеттингу написано CCHM пейпер 2016 року, де показано модельну структуру категорії кубічних множин.

```
cubical
  : modelStructure cSet
  = ( kanComplexes ,
      monos ,
      geometricRealisation )
```

де cSet = [$\square^{op}$,Set], а $\square$ — категорія збагачена структурою алгебри де Моргана.

8 Модальна гомотопічна теорія

Тут представлена рецензія на книгу Девіда Корфілда "Модальна Гомотопочна Теорія Типів. Перспектива нової логіки для філософії".

Модальна гомотопічна теорія типів — це сучасне поєднання модальної теорії типів (для вираження або класичних модальних логік висловлювань K, S4, S5 і т.д. або сучасних (ко)-монадичних інтерпретаторів які реалізують відповідні модальності) з та у застосуванні до гомотопічної теорії типів. Це поєднання мотивовано багатьма дослідженнями та відкритими проблемами одна з яких це неможливість реалізації доведення Брауера про нерухому точку у чистій HoTT. До цього модальності в теоретичній інформатиці моделювалося за допомогою семантик Кріпке, тому перехід до категорної моделі монад або роботі в 2-теорії типів, дав змогу формалізації інфінітіземальних околів та понять многовидів у алгебраїчній геометрії.

Крім метатеоретичної частини у зв'язаних топосах та геометричних морфізмах між ними, сформульованої Вільямом Ловером, автор показує також основи залежної теорії типів, та її застосування до філософії та формальної природньої людської мови. На мою думку потужна логічна основа та формальна модель NLP — це запорука успішної системи обробки природньої мови гомосапієнсів побудованої на стохастичних моделях.

Найприємніше в книзі є те, що значна її частина приділяється історії виникнення модальної HoTT як результату пошуку формальної моделі для калібрувальної інваріантністі (теоретична фізика) Урсом Шрайбером. Відповідь на цей запит спочатку була представлена Майклом Шульманом як cohesivett, а пізніше, на Конференції по диференціальній геометорії в модальній HoTT влаштованій Феліксом Велленом, на якій мені довелось побувати, і саме конструктивне доведення теореми Брауера про нерухому точку в модальній HoTT. Детальний звіт про конференцію.

Модальності важливі не тільки для застосувані у формалізації часу в природній мові, та геометричного поняття коіндуктивного околу, або дійсних чисел, модальності також формалізують поняття процесу, як фізичного так і інформатико-теоретичного. Позаяк числення процесів цілком поглинається модальною теорією типів у зв'язаних топосах, модальна HoTT у свою чергу не тільки забезпечує формалізацію теоретичної фізики, але і відкриває двері у формальну філософію різних модальностей, або інтерпретацій.

Зараз $\flat$-модальність уже вбудована в Агду, та існують уже прототипи мультимодальних тайпчекерів. Раджу цю книжку усім математикам, усім програмістам, та усім філософам у якості підручника з формальної філософії.

9 Метафілософія

Коротко про сучасні філософії. Якщо визначити філософію предикативно, це наука вивчає певний набір питань, типу: як жити добре, чи реальний всесвіт та інші питання гносеологічних категорій, свобода волі, етика, математика, музика, поезія, література, - усе це взагалі-то питання чи мовні набори, що цікавлять сучасних філософів.

У цій нотатці ми спробуємо побудувати формальну систему і на її прикладі показати інтерпретацію трьох ліній передачі сучасної філософії: європейської чи континентальної філософії — школи, яка задала початок глобальній інтерпретації світу та реконструкції мов, у тому числі й математики; східної філософії як приклад особливої школи, на прикладі якої ми будуватимемо модель; та аналітичної чи англосаксонської філософії, що формалізується сучасною математикою.

Європейська філософія

На наш погляд, головне питання європейської філософії - це Good Life. Як жити, як жити добре самому, у соціумі, які цілі можуть стояти перед індивідом та видом, баланс етики та етика балансу. Європейська філософія народила геометорію, психоаналіз, навчила людей не боятися свободи, трансформувати агресію, бути більш зрілою істотою, і під вінець свого розвитку поставила питання про мову та мовну гру, як основний інструмент рефлексуючої свідомості.

Мова перестала мати ґрунт, вона стала просто візерунками, семантика яких втрачена, філософія стала формою літературного мистецтва.

Представники континентальної філософії: Арістотель, Платон, Кант, Декарт, Ніцше, Фрейд, Юнг, Юм, Хайдегер, Адорно, Хабермас, Делез.

Тибетська філософия

У східній філософії центральним питанням є визволення себе і інших, в першу чергу від різних форм страждання. Ця філософія має чітку систему, яка нерозривно пов’язана з тілесними та розумовими практиками, і вижила протягом тисячоліть у законсерованому гірському плато. Тут також порушуються питання етики та свободи волі, але основний наголос робиться на інтелектуальних та неконцептуальних вправах, що ведуть до безпосереднього переживання простору.

Деякі формулювання східної філософії, такі як недвоїстість всіх феноменів піддаються формалізації в гомотопічній теорії типів (використовуючи методи аналітичної філософії), що спонукало до подальших досліджень у галузі формалізації езотеричних теорій.

Представники східної (тибетської) філософії: Атіша, Нагарджуна, Бхававівека, Камалашила, Шантаракшита, Арьядева, Буддхапаліта, Чандракірті, Цонкапа, Міпам, Лонгченпа.

Аналітична філософія

Аналітична філософія народжена в математиці, рання аналітична філософія починається напевно з Лейбніца, Ньютона та Ейлера. Пізня аналітична філософія починається з Фреге і далі за списком: Рассел, Уайтхед, Дедекінд, Пеано, Гільберт, Фон-Нейман, Каррі, Акерманн, Карнап, Сколем, Пост, Гедель, Черч, Берньє, Тюрінг, Кліні, Россер, Мак-Лейн Ловір, Гротендік, Скотт, Джояль, Терньє, Мартін-Льоф, Мілнер, Жирар, Плоткін, Рейнольдс, Бакус, Барр, Барендрехт, Лер'є, Силі, Кокан, Х'юет, Ламбек, Воєводський, Еводі, Шульман, Шрайбер.

Якщо описати двома словами головне питання аналітичної філософії — це мова простору. Побудова мови, яка дасть формальний фундамент не лише математики та роздумів, а й самої філософії.

Мова простору (абстрактний нонсенс)

Формальні підстави мови роздумів, математики (усієї) та фізики (всесвіту). Оскільки математика вміщає всі теорії, то мова математики є дуже обмеженою у порівнявнні з математичними теоріями та моделями, хоча оперує всіма, зокрема найабстрактнішими математиками. Всі інші моделі нижчих рівнів записуються на цій мові програмування.

Мовні фреймворки (теорії)

Мовні фреймворки для менш формальних (з парадоксами) та нечітких (стохастичних) систем. Мовні фреймворки це теорії та тактики доведення, або системні декомпозиції які допомагають аналізувати об'єкти формально та обчислювально.

Конкретні математичні моделі

Прикладна філософія Використання мовних фреймворків для опису конкретних феноменів. Конкретні феномени тут представляються як конкретні моделі певних теорій, побудованих на своїх мовних фреймворках.

Обчислювальні моделі (прикладна математика)

Останній онтологічний рівень сучасної формальної філософії — це конкретні обчислення на конкретних моделях, які є практичними дослідженнями.

10 Прикладна математика

Цю статтю я вирішив написати аби пролити світло на предметноорієнтованість математичних дисциплін через призму власного досвіду. Мені не пощастило (як я про це думав колись) у тому, що я опинився в світі, де математика уже існувала задовго до мене, проте значною мірою доля щастя була в тім, що комп'ютери змінювали математику на моїх очах.

При виборі фалькутету ПМ я керувався елітарним критерієм максимального конкурсу на місце, однак (як зрозумів згодом) стратегія немає такого значення, як має воля до всезбагнення. Саме вона мені дозволила пережити перші розчарування в моєму освітньому шляху і я почав усвідомлено мімікрувати під систему освіти та предмети західного аналітичного духу філософії, як і належить математикам.

Інститут формальної математики

Зараз, коли у мене є своя лабораторія, яка стабільно приносить плоди, як у вигляді публікацій так і у вигляді артефактів, я по-іншому уявляю кафедральну структуру факультету:

— КФ-033: Кафедра формальної філософії
— КМ-111: Кафедра чистої математики
— КМ-113: Кафедра прикладної математики
— КА-121: Кафедра мовного забезпечення
— КВ-123: Кафедра теоретичної інформатики

КФ-033 Формальна філософія (магістр)

Перш за все, всі основи математики (УДК 510) я би формальним чином засунув на спеціальну новостворену кафедру формальної аналітичної математичної філософії, яка би читала курси по математичній логіці та філософії для всіх інших кафедр та факультетів як базовий провайдер. Саме так це зроблено в CMU. Місце аналітичної філософії розкривається в курсі історії філософії.

— Теорія категорій (екзамен) КФ-033-01
— Формальні логіки (екзамен) КФ-033-02
— Категоріальна логіка (екзамен) КФ-033-03
— Лямбда-числення (екзамен) КФ-033-04
— Модальні логіки (екзамен) КФ-033-05
— Теорія типів Мартіна-Льофа (екзамен) КФ-033-06
— Формалізація штучних мов КФ-0133-07
— Формалізація людських мов КФ-033-08

Ці курси є основою для усіх кафедр інституту прикладної математики. Для інших спеціальностей друго наувого-освітнього рівня потрібно набрати мінімум три повноцінних курси (36 кредитів), які читаються на кафедрі філософії (КФ-033).

КФ-033 Формальна філософія (доктор)

— Мультимодальна гомотопічна теорія типів КФ-033-09
— Гіперграфові моделі фізики КФ-033-10

КМ-111 Чиста математика (магістр)

Кафедра чистої математики, яка на мій погляд обов'язково потрібна на факультеті прикладної математики, аби прикладна математика володіла усіма математичними інструментами які потрібно вміти використовувати для поліедральних та нелінійних оптимізацій. Цей провайдер повинен постачати курси, які могли би конкурувати з російськими школами алгебраїчної топології та геометрії та курсами університету Карнегі-Мелона.

— Алгебра (екзамен) КМ-111-01
— Геометрія (екзамен) КМ-111-02
— Теорія гомотопій (екзамен) КМ-111-03
— Теорія чисел КМ-111-04
— Теорія схем Гротендіка КМ-111-05
— p-адичний аналіз КМ-111-06
— Теорія Ходжа. Мотивне інтегрування (екзамен) КМ-111-07
— Алгебраїчна топологія (екзамен) КМ-111-08
— Диференціальна геометрія (екзамен) КМ-111-09
— Гомотопічна теорія типів КМ-111-10
— Теорія топосів КМ-111-11
— Модельні категорії Квілена КМ-111-12

Ці курси потрібні в основному для спеціалістів за спеціальностями 111 (чиста математика) та 113 (прикладна математика).

КМ-111 Чиста математика (доктор)

Деякі спеціальні теми для створення мотиваційної конкуренції з НАН України:

— Сімпліціальна гомотопічна теорія КМ-111-13
— Локальна гомотопічна теорія (екзамен) КМ-111-14
— K-теорія КМ-111-15
— Теорія інфініті-категорій КМ-111-16

КМ-113 Прикладна математика (магістр)

На початкових курсах своєї подорожі я керувався головним чином спадком радянської школи, тому як доступу до Інтернету, а тим більше до Шпрінгер видань не було, я зміг скласти на той час достатньо повну бібліотеку усього класифікатора УДК 51, яку мені вдалося відновити тільки відносно недавно. В умовах тотального розчарування в рядянській педагогічній школі я вимушений був сфокусуватися значним чином на програмуванні, а не на математиці. Вже тоді мені стало зрозуміло, що запорука якісного середовища математика полягає в якійсній основі. Хоча прикладна математика традиційно не фокусується на засобах програмування та операційних системах, це направлення мене цікавило з самого першого дня першого курсу університету і допомогло пережити складні части наукової зневіри.

Курси другого освітнього рівня, які читаються на кафедрі прикладної математики є обов'язковими для спеціальностей 111 (чиста математика) та 113 (прикладна математика).

— Дискретна математика (екзамен) КМ-113-01
— Математична статистика (екзамен) КМ-113-02
— Формальні логіки (екзамен) КМ-113-03
— Комплексний аналіз КМ-113-04
— Чисельні методи (екзамен) КМ-113-05
— Математичний аналіз (екзамен) КМ-113-06
— Функціональний аналіз (екзамен) КМ-113-07
— Звичайні диференційні рівняння (екзамен) КМ-113-08
— Диф. рівняння в частинних похідних (екзамен) КМ-113-09
— Рівняння математичної фізики КМ-113-10

Рестроспективний погляд на мою дисоціативну освіту в минулому дозволяє зараз мені реігтегрувати цей досвід і виділити найсуттєвіші його частини. В дусі аналітичної філософії корені логіки, теорії типів та лямбда числення як основи для математики та програмування слід шукати в роботах Расела, Уайтгеда, Пеано, Черча, Карі. Пізніше варно збагнути основні технічні питання математики: робота з універсальними властивостями (теоремами), поняття нескінченності та принципи її декомпозиції (індексація натуральними числами), різні аксіоматичні геометрії, поняття дійсного числа, граничного переходу, класичного аналізу нескінченно-малих Ейлера та Лейбніца. Звичайні диференційні рівняння відкриють багато схованок у вигляді тензорного числення та лінійної алгебри, а рівняння в частинних похідних відкриють ворота у функціональний аналіз, гармонічний аналіз, алгебри Лі та покажуть трохи нелінійної фізики. Окрім рівнянь теоретичної фізики уся база прикладної маметатики зводиться до диференційних рівнянь, а усі інші аспекти математики, як аналіз чи тензорне числення уже випливають як необхідний апарат. З академічної точки зору традиційно виклад матеріалу ведеться з математичних теорій, які не містять залежностей і далі усладнюючи алгебраїчні структури переходять то теорій верхнього рівня. Важливо завжди бачити повну картину та намагатись утримати головну мотивацію математики — обчислення.

КМ-113 Прикладна математика (доктор)

Прикладна математика зараз сприймається як все, що я можу порахувати на комп'ютері, а особливо задачі які опимально обчислюють на границі сучасної точності за початковими та граничними умовами. За цей час спектр математичних моделей для мене розширився та доповнився новими дисциплінами. Наступні 10 років я провів більше займаючись формальною математикою та філософією і лише мріяв вернутися в часи базової вищої освіти повністю переглянувши усі математичні засоби та їх взаємодію. Для виконання своєї дисертаційної роботи мені довелося взяти такі додаткові курси на кафедрі чистої математики:

— Алгебраїчна топологія (екзамен) КМ-111-08
— Диференціальна геометрія (екзамен) КМ-111-09
— Гомотопічна теорія типів КМ-111-10
— Теорія топосів КМ-111-11

Як би я не хотів вирватися з лап прикладної математики в межі чистої та формальної математики чи формальної філософії думки завжди повертаються у прикладну математику. Прикладна математика в моєму розумінні це чиста математика сформульована формальними методами з метою точної (в таких випадках як логіка, чи символьні обчислення) чи наближеної калькуляції. Наближене обчислення у свою чергу передбачає додатковий апарат у вигляді теорем та теорій, який доводить збереження властивостей та корекність моделі при наближених обчисленнях. Таким чином чистий математик, який вирішив застосувати певну математичну теорію та порахувати в рамках її обчислювальної моделі певний клас формул, вже перетворюється на прикладного математика.

Хоча найабстрактніша чиста математика не оперує конкретними моделями, а лише їх сигнатурами і метамоделями сигнатур (та вище), у ній теж є обчислення — це точні обчислення логіки (доведення теорем), в даному випадку гомотопічної теорії типів, особливого виду логіки який підходять як основна мова для усієї математики. Цей предмет (який вивчає основну мову математики) як і усі логіки можна віднести до прикланої математики (логічні числення), а також і до теоретичної інформатики (яка необхідна для побудови математичної мови-інструменту). Також можна класифікувати цей предмет як формальну філософію, тому що для прикладного формального філософа ця мова є єдиним необхідним інструментом, на відміну відміну від інших прикладних математиків, які зазвичай використовують додатково потужні прикладні теорії (алгтоп та дифгеом) або їх частини.

Оскільки (математичний) твір написаний на (формальній) мові потребує форми, з давна математики використовували літературний жанр у її якості. Це певного роду напів-формальна напів-людська мова, базис якої зводиться до кванторів: "для всіх x, таких що ..."та "існує таке x, що ...". Математика як одна з філософських дисциплін (серед яких також музика, образотворче мистецтво, література, этика) це певна форма поезії яка працює виключно з простором на абсолютному рівні. Без проміжків, без узагальнень і апроксимацій нелінійних динамічних систем, лише абсолютна фібраційна тотальна логіка простору. Такі (математичні) вірші з давніх давен цінувалися (філософами) високо, а їх створення супроводжується не тільки майже езотеричною традицією але і інтегровано в те, що ми зараз називаємо науковим методом пізнання реальності (з метою отримання максимального задоволення максимальний проміжок часу). Велике мистецтво в таких творах показати сутність сприйняття та дати авторський коментар (відповідної глибини), хоча самі теореми теж є самостійним об'єктом творчості.

З появою формальної верифікованої математики (1968) літературна форма математики розширилися та доповнилася тим, що вивчає теоретична інформатика — програмним продуктом, який можна скачати та поставити, у якому є своє обчислювальне середовище, своя мова програмування, своя базова бібліотека та приклади програм (теореми). Але на відміну іншої філософської дисципліни, ігрової індустрії, культура верифікованої математики не набула таких масштабів стандартизації як ISO стандарти з урегулювання складних інформаційних систем включно з Інтернетом. Можна сказати, що сучасні чисті математики обмежують себе чисто філософськими інструментами — системами доведення теорем (Agda, Lean) і тішуться з того; у той час, як прикладі математики зосереджені або в математичних спеціалізованих пакетах Mathematica, Mapple, Mathlab, GAP або з підручних засобів на Github будують MVP моделі та оформляють це у вигляді Jupyter блокноту. Культура блокнотів достатньо непогане поєднання літературного жанру з образотворчим мистецтвом візуалізації даних (як це маніфестується у системі Processing). І чисті і прикладні сучасні математики використовують TeX як де-факто стандарт при класичній літературній публікації (творів мистецтва), а самі програмні продукти або MVP моделі є лише артефактами такої публікації. Лише ті твори (праці) знаходять певну стандартизацію, коли доходять до офіційних пакетів для публічних пакетних менежерів певних мов програмування (як Owlbarn). Певну культуру в сенсі цифрової публікації здобули формальні операційні системи та формальні мови програмування, вийшовши з теоретичної інформатики, однак культура прикладної математики досі не дозволяє говорити про замкнене середовище (наприклад Mirage юнікернел) у якому міститься достатня кількість мовних засобів для створення мов та систем (Menhir), для символьних обчислень (Axiom), для логік та систем доведення теорем (Coq), для програмування (OCaml), для тензорного числення (SPIRAL, Futhark), для паралельних та узгоджених систем (Multicore, LING), для імітаційного моделювання (Simulink) для публікації літературних математичних творів (TeX), для блокнотів (Jupyter). Екосистеми теоретичної інформатики французької та британської шкіл (OCaml, Haskell) які мають здатність вистояти перед викликами уніфікованого середовища прикладного математика.

— Видавнича система для верстки та графіків, яка нагадує TeX
— Лінійна алгебра на BLAS та LAPACK
— Тензорні обчислення на GPU/AVX
— Система вводу-виводу PCIe на NVMe 2.0
— CAS система для символьної математики
— Віконний менеджер та система рендерінгу на GPU
— Симулятор нейромереж у форматі ONNX
— Інтерфейс прикладного програмування

KB-123 Теоретична інформатика (магістр)

Якщо коротко описати мій шлях після здобуття ступеня магістра то я би це визначив як 10 років укорінення та здобуття впевненості в професії програміста. За цей час вдалося зрозуміти вимоги до властивостей середовищ виконання на виробництві. Теоретична інформатика (комп'ютерна інженерія або комп'ютерні науки) дозволила мені глибше зазирнути у різні школи моделювання обчислень, що дало змогу спробувати виконати декілька вправ у написанні середовищ виконання. Паралельно гроші на життя приносила робота пов'язана з розвитком більше прикладних бібліотек.

Курси другого наувого-освітнього рівня, що читаються на кафедрі теоретичної інформатики є обов'язковими для спеціальностей 113 (прикладна математика), 121 (мовне забезпечення), 123 (теоретична інформатика).

— Лямбда-числення (екзамен) KB-123-01
— Теорія графів (екзамен) KB-123-02
— Теорія мов програмування (курсова, екзамен) KB-123-03
— Теорія операційних системи (курсова, екзамен) KB-123-04
— Теорія масового обслугов. (курсова, екзамен) KB-123-05
— Теорія розподілених систем KB-123-06
— Системи реального часу та телеметрія (РТС) KB-123-07
— Інформаційно-пошукові системи (ІПС) (екзамен) KB-123-08
— Моделювання складних систем (МСС) KB-123-09
— Комп'ютерні сховища даних KB-123-10
— Комп'ютерні обчислення KB-123-11
— Комп'ютерна графіка KB-123-12
— Комп'ютерні мережі KB-123-13
— Веб-програмування (екзамен) KB-123-14
— Функціональне програмування (екзамен) KB-123-15

Фундаментальні поняття комп'ютерних наук, такі як обчислювальність, алгоритмічна розв'язність, тотальність, консистентність, категоріальна алгебраїчність, розширилися за цей час в моєму розумінні математичною лінгвістикою. Визріла впевненість в необхідності зафіксувати особливий набір мовних засобів, в рамках наступного 10-річного дослідження в області теоретичної інформатики, що дозволили би побудувати фундамент для моделей які вивчає прикладна математика. Зараз для мене теоретична інформатика і є математичною лінгвістикою, завдання якої конструювати формальні мови з необхідними властивостями для математичних обслатей. Цей продукт знайшов форму монографії та включає наступні мовні артефакти:

— Інтерпретатор віртуальної машини
— Компілятор класу System F_ω
— Гомотопічна система
— Система процесів та черг віртуальної машини без надлишкового копіювання

Хоча вступ доціє роботи вимагає повного занурення в основи роботи процесора та пошуку таких сніпетів які ефективно навантажуватимуть конвеєри фізичного процесору. Технологія побудови інтерпретаторів які разом з виконуваними программами повністю поміщаються у L1 кеш процесора може бути такою ж ефективною як повністю наперед скомпільований оптимізований код.

Що стосується початкової мови для програмування, то я рекомендую в програму для 10-11 класів ввести LISP у формі Racket, а також глибоко зануритися в роботи авторів цього мовного середовища.

11 Абелеві категорії

Абелеві категорії — це збагачене поняття категорії Сандерса-Маклейна поняттями нульового об'єкту, що одночасно ініціальний та термінальний, властивостями існування всіх добутків та кодобутків, ядер та коядер, а також, що всі мономорфізми і епіморфізми є ядрами і коядрами відповідно (тобто норомальними).

```
def isAbelian (C: precategory): U₁
 := Σ (zero:    hasZeroObject C)
      (prod:    hasAllProducts C)
      (coprod:  hasAllCoproducts C)
      (ker:     hasAllKernels C zero)
      (coker:   hasAllCokernels C zero)
      (monicsAreKernels:
         Π (A S: C.C.ob) (k: C.C.hom S A),
         Σ (B: C.C.ob) (f: C.C.hom A B),
         isKernel C zero A B S f k)
      (epicsAreCoKernels:
         Π (B S: C.C.ob) (k: C.C.hom B S),
         Σ (A: C.C.ob) (f: C.C.hom A B),
         isCokernel C zero A B S f k), U
```

Мотивація

Ось п'ять коротких формальних застосувань математичного апарату абелевих категорій:

Гомологічна алгебра: абелеві категорії забезпечують основу для гомологічної алгебри, яка є розділом алгебри, що вивчає властивості груп гомології та когомології. Теорія похідних функторів, яка є фундаментальним інструментом гомологічної алгебри, базується на понятті абелевої категорії.

Алгебраїчна геометрія: абелеві категорії використовуються в алгебраїчній геометрії для вивчення когомологій пучка, що є потужним інструментом для розуміння геометричних властивостей алгебраїчних многовидів. Зокрема, категорія пучків абелевих груп на топологічному просторі є абелевою категорією.

Теорія представлень: абелеві категорії природно виникають у вивченні теорії представлень, яка є розділом математики, який має справу з алгебраїчними структурами, що виникають під час вивчення симетрії. Категорія модулів над кільцем, наприклад, є абелевою категорією.

Топологічна квантова теорія поля: абелеві категорії відіграють центральну роль у вивченні топологічної квантової теорії поля, яка є розділом математичної фізики, що має справу з математичною структурою топологічних просторів. У цьому контексті абелеві категорії виникають як категорії граничних умов для певних типів теорій топологічного поля.

Теорія категорій: Нарешті, абелеві категорії є важливим об'єктом дослідження в теорії категорій, яка є розділом математики, який вивчає властивості категорій та їхні зв'язки. Зокрема, абелеві категорії забезпечують природне середовище для вивчення властивостей адитивних функторів, які є фундаментальним інструментом у теорії прохідних категорій та функторів. Тут можна порадити роботу румунських математиків Бакура і Деляну «Вступ в теорію категорій та функторів».

Джерела

Поняття абелевої категорії вперше було введено математиком Александром Гротендіком у його основоположній статті «Sur quelques points d'algèbre homologique» (Про деякі аспекти гомологічної алгебри), опублікованій у 1957 році. Гротендік представив абелеві категорії як спосіб об'єднання вивчення гомологічної алгебри в різних математичних дисциплінах, таких як алгебраїчна геометрія, алгебраїчна топологія та теорія представлень.

У цій статті Гротендік визначив абелеву категорію як категорію, яка задовольняє ряду аксіом, включаючи існування ядер і коядер, теореми ізоморфізму та існування точних послідовностей. Він показав, що багато категорій алгебраїчних об'єктів, таких як категорії модулів над кільцем або пучків абелевих груп на топологічному просторі, є абелевими категоріями.

З тих пір концепція абелевих категорій стала фундаментальним інструментом у багатьох областях математики, включаючи алгебраїчну геометрію, теорію представлень, гомологічну алгебру та теорію категорій. Він також застосовувався в таких галузях фізики, як топологічна квантова теорія поля.

Визначення

При формальній побудові абелевих категорій потрібно розділяти типову сигнатурну інформацію абстрактної абелевої категорії та її інстанціацій, таких як категорія абелевих груп, модулів над кільцем, тощо.

Всі, хто хоче побачити сучасні абелеві категорії в кубічній Агді, може подивитися магістерську роботу 2021 року Девіда Еліндера «Дослідження абелевих категорій і унівалентній теорії типів».

```
module abelian where
import lib/mathematics/categories/category
import lib/mathematics/homotopy/truncation

def zeroObject(C: precategory) (X: C.C.ob): U₁
 := Σ (bot: isInitial C X) (top: isTerminal C X), U

def hasZeroObject (C: precategory) : U₁
 := Σ (ob: C.C.ob) (zero: zeroObject C ob), unit

def hasAllProducts (C: precategory) : U₁
 := Σ (product: C.C.ob -> C.C.ob -> C.C.ob)
      ($\pi_1$: Π (A B : C.C.ob), C.C.hom (product A B) A)
      ($\pi_2$: Π (A B : C.C.ob), C.C.hom (product A B) B), U

def hasAllCoproducts (C: precategory) : U₁
 := Σ (coproduct: C.C.ob -> C.C.ob -> C.C.ob)
      ($\sigma_1$: Π (A B : C.C.ob), C.C.hom A (coproduct A B))
      ($\sigma_2$: Π (A B : C.C.ob), C.C.hom B (coproduct A B)), U

def isMonic (P: precategory) (Y Z : P.C.ob) (f : P.C.hom Y Z) : U
 := Π (X : P.C.ob) (g1 g2 : P.C.hom X Y),
    Path (P.C.hom X Z) (P.P.∘ X Y Z g1 f) (P.P.∘ X Y Z g2 f)
 -> Path (P.C.hom X Y) g1 g2

def isEpic (P : precategory) (X Y : P.C.ob) (f : P.C.hom X Y) : U
 := Π (Z : P.C.ob) (g1 g2 : P.C.hom Y Z),
    Path (P.C.hom X Z) (P.P.∘ X Y Z f g1) (P.P.∘ X Y Z f g2)
 -> Path (P.C.hom Y Z) g1 g2

def kernel (C: precategory) (zero: hasZeroObject C)
    (A B S: C.C.ob) (f: C.C.hom A B) : U₁
 := Σ (k: C.C.hom S A) (monic: isMonic C S A k), unit

def cokernel (C: precategory) (zero: hasZeroObject C)
    (A B S: C.C.ob) (f: C.C.hom A B)  : U₁
 := Σ (k: C.C.hom B S) (epic: isEpic C B S k), unit

def isKernel (C: precategory) (zero: hasZeroObject C)
    (A B S: C.C.ob) (f: C.C.hom A B) (k: C.C.hom S A) : U₁
 := Σ (ker: kernel C zero A B S f), Path (C.C.hom S A) ker.k k

def isCokernel (C: precategory) (zero: hasZeroObject C)
```

```
    (A B S: C.C.ob) (f: C.C.hom A B) (k: C.C.hom B S) : U₁
 := Σ (coker: cokernel C zero A B S f), Path (C.C.hom B S) coker.k k

def hasKernel (C: precategory) (zero: hasZeroObject C)
    (A B: C.C.ob) (f: C.C.hom A B) : U₁
 := ‖_‖$_{−1}$ (Σ (monic: isMonic C A B f), unit)

def hasCokernel (C: precategory) (zero: hasZeroObject C)
    (A B: C.C.ob) (f: C.C.hom A B) : U₁
 := ‖_‖$_{−1}$ (Σ (epic: isEpic C A B f), unit)

def hasAllKernels (C : precategory) (zero: hasZeroObject C) : U₁
 := Σ (A B : C.C.ob) (f : C.C.hom A B), hasKernel C zero A B f

def hasAllCokernels (C : precategory) (zero: hasZeroObject C) : U₁
 := Σ (A B : C.C.ob) (f : C.C.hom A B), hasCokernel C zero A B f
```

Абелеві категорії забезпечують природну основу для вивчення гомологічної алгебри, яка є розділом алгебри, що має справу з алгебраїчними властивостями груп гомологій та когомологій. Зокрема, абелеві категорії створюють сеттінг, де можна визначити поняття похідних категорій і похідних функторів.

Основна ідея похідних категорій полягає в тому, щоб ввести нову категорію, яка побудована з абелевої категорії шляхом «інвертування» певних морфізмів, майже так само, як будується поле часток на області цілісності. Похідна категорія абелевої категорії фіксує «правильне» поняття гомологічних і когомологічних груп і забезпечує потужний інструмент для вивчення алгебраїчних властивостей цих груп.

Похідні функтори є фундаментальним інструментом гомологічної алгебри, і їх можна визначити за допомогою концепції похідної категорії. Основна ідея похідних функторів полягає в тому, щоб взяти функтор, який визначено в абелевій категорії, і «підняти» його до функтора, який визначений у похідній категорії. Похідний функтор потім використовується для обчислення вищих груп гомології та когомології об'єктів в абелевій категорії.

Використання похідних категорій і функторів зробило революцію у вивченні гомологічної алгебри, і це призвело до багатьох важливих застосувань в алгебраїчній геометрії, топології та математичній фізиці. Наприклад, похідні категорі використовувалися для доведення фундаментальних результатів алгебраїчної геометрії, таких як знаменита теорема Гротендіка-Рімана-Роха. Вони також використовувалися для вивчення дзеркальної симетрії в теорії суперструн.

12 Мова простору

Ця стаття присвячена цілком CCHM верифікатору гомотопічної системи типів з двома рівностями, відомої як HTS система Воєводського або система 2LTT Анненкова-Капріотті-Крауса-Саттлера. Крім рівності на претипах, Андерс містить як примітив стек де Рама Черубіні-Шрайбера, що робить його придатним для програмування когомологій та синтетичної диференціальної геометрії.

```
type exp =
  | EPre of Z.t | EKan of Z.t | EVar of name | EHole
  | EPi of exp * (name * exp) | ELam of exp * (name * exp) | EApp of exp * exp
  | ESig of exp * (name *exp) | EPair of tag*exp*exp | EFst of exp | ESnd of exp
  | EId of exp | ERef of exp | EJ of exp | EField of exp * string
  | EPathP of exp | EPLam of exp | EAppFormula of exp * exp
  | EI | EDir of dir | EAnd of exp * exp | EOr of exp * exp | ENeg of exp
  | ETransp of exp * exp | EHComp of exp * exp * exp * exp
  | EPartial of exp | EPartialP of exp * exp | ESystem of exp System.t
  | ESub of exp * exp * exp | EInc of exp * exp | EOuc of exp
  | EGlue of exp | EGlueElem of exp * exp * exp | EUnglue of exp
  | EEmpty | EIndEmpty of exp
  | EUnit | EStar | EIndUnit of exp
  | EBool | EFalse | ETrue | EIndBool of exp
  | EW of exp * (name * exp) | ESup of exp * exp | EIndW of exp * exp * exp
  | EIm of exp | EInf of exp | EIndIm of exp * exp | EJoin of exp
```

Космос або структура всесвітів

Почати статтю хочу з влаштування тайп чекера. По суті тайп чекер це функція над мовою виразів exp. Розглянемо пристрій функцій тайп чекера рядково для кожного притимітива з дерева exp.

```
type exp = | EPre of Z.t | EKan of Z.t | EVar of name | EHole
```

Система HTS (або 2LTT) характеризується наявністю двох ієрархій предикативних всесвітів U_i для фібраційних типів і V_j для претипів, де живе гомотопічний багатовимірний відрізок. Також в ядрі тайп чекерів зазвичай знаходяться конструктори для змінних і дірок зручних для процесу вилучення доказів. Рівняння цих примітивів будуть дані у цьому параграфі.

Сам тайпчекер влаштований таким чином (стандартна практика, дивіться наприклад, тайпчекери Mini-TT або cubicaltt, що вже стали класичними), що для процесу бета-нормалізації (евалуації) або інтерпретування виразу використовується внутрішнє уявлення, оптимізоване для потреб ефективних обчислень.

```
type value = | VKan of Z.t | VPre of Z.t | Var of name * value | VHole
```

Таким чином сигнатура тайпчекера виглядає так:

```
and check ctx (e0: exp) (t0: value)
  = traceCheck e0 t0; try match e0, t0 with
```

Алгоритм класичний і звучить так: для типового виразу та його екземпляра ми беремо екземпляр типового виразу виводимо його тип і порівнюємо із заданим типовим виразом. Якщо вони збігаються все добре, якщо ні — то помилка типізації. Далі йде список патерн-матчінг рівнянь всіх функцій на деревах мовних виразів expr:

```
check:
  | EHole, v -> traceHole v ctx
```

```
  | e, VPre u -> begin match infer ctx e with
    | VKan v | VPre v ->
      if ieq u v then ()
      else raise (Ineq (VPre u, VPre v))
    | t -> raise (Ineq (VPre u, t)) end
  | e, t -> eqNf (infer ctx e) t

conv:
  | VKan u, VKan v -> ieq u v | VPre u, VPre v -> ieq u v
  | Var (u, _), Var (v, _) -> u = v

eval:
  | EPre u -> VPre u | EKan u -> VKan u
  | EVar x -> getRho ctx x | EHole -> VHole

infer:
  | VPre n -> VPre (Z.succ n) | VKan n -> VKan (Z.succ n)
  | EPre u -> VPre (Z.succ u) | EKan u -> VKan (Z.succ u)
  | EVar x -> lookup x ctx

inferV:
  | Var (_, t) -> t | VPre n -> VPre (Z.succ n)
  | VKan n -> VKan (Z.succ n)
```

```
act:
  | Var (i, VI) -> actVar rho i
  | Var (x, t) -> Var (x, act rho t) | VHole -> VHole
  | VKan u -> VKan u | VPre u -> VPre u

check:
  | e, t -> eqNf (infer ctx e) t

and getRho ctx x = match Env.find_opt x ctx with
  | Some (_, _, Value v) -> v
  | Some (_, _, Exp e) -> eval e ctx
  | None -> raise (VariableNotFound x)

and eqNf v1 v2 : unit = traceEqNF v1 v2;
  if conv v1 v2 then () else raise (Ineq (v1, v2))

and lookup (x : name) (ctx : ctx) = match Env.find_opt x ctx with
  | Some (_, Value v, _) -> v
  | Some (_, Exp e, _) -> eval e ctx
  | None -> raise (VariableNotFound x)
```

Тут описується так звана база рекурсії та робота з контекстом верифікатора (getRho, lookup).

Π-тип

Першим математичним прувером взагалі та першим прувером на фібраційному типі вважається [мною] AUTOMATH де Брейна. Перша повна формальна система CoC та лямбда-куб були детально розроблені Барендрехтом. Проте батьком формальної математики прийнято вважати Мартіна-Лефа. Його типова система досі формує обчислювальну основу сучасних пруверів. Традиційно MLTT складається з $\Pi, \Sigma, 0, 1, 2, W, \mathrm{Id}$ типів. При розробці верифікатора Anders ми керувалися мотивацією мінімалістичності, тому відкинули варіант імплементації загальної схеми індуктивних типів і вищих ідуктивних типів (HIT), а вирішили реалізувати як індуктивне ядро класичні W-типи системи MLTT.

Досніповий код верифікатора для П-типу:

```
eval:
  | EPi (a, (p, b)) -> let t = eval a ctx in
    VPi (t, (fresh p, closByVal ctx p t b))
  | ELam (a,(p, b)) -> let t = eval a ctx in
    VLam (t,(fresh p, closByVal ctx p t b))
  | EApp (f, x) -> app (eval f ctx, eval x ctx)

infer:
  | EPi (a, (p, b)) -> inferTele ctx p a b

inferV:
  | VPi (t, (x, f)) -> imax (inferV t) (inferV (f (Var (x, t))))
  | VLam (t, (x, f)) -> VPi (t, (x, fun x -> inferV (f x)))
  | VApp (f, x) -> begin match inferV f with
    | VPi (_, (_, g)) -> g x
    | v -> raise (ExpectedPi v) end

act:
  | VLam (t, (x, g)) -> VLam (act rho t, (x, g >> act rho))
  | VPi (t, (x, g)) -> VPi (act rho t, (x, g >> act rho))
  | VApp (f, x) -> app (act rho f, act rho x)

app:
  | f, x -> VApp (f, x)

conv:
  | VPi (a,(p,f)), VPi (b,(_,g)) ->
    let x = Var (p,a) in conv a b && conv (f x) (g x)
  | VLam (a,(p,f)), VLam (b,(_,g))
  | VApp (f,a), VApp (g,b) -> conv f g && conv a b

check:
  | ELam (a, (p, b)), VPi (t, (_, g)) ->
    ignore (extSet (infer ctx a)); eqNf (eval a ctx) t;
    let x = Var (p, t) in let ctx' =
    upLocal ctx p t x in check ctx' b (g x)

and inferTele ctx p a b =
    ignore (extSet (infer ctx a));
    let t = eval a ctx in let x = Var (p, t) in
    let ctx' = upLocal ctx p t x in
    let v = infer ctx' b in imax (infer ctx a) v

and inferLam ctx p a e =
    ignore (extSet (infer ctx a)); let t = eval a ctx in
    ignore (infer (upLocal ctx p t (Var (p, t))) e);
    VPi (t, (p, fun x -> inferV (eval e (upLocal ctx p t x))))
```

Π-тип тестується інтерналізацією, а за наявності гомотопічної рівності ще й функціональною екстенсіональністю.

```
def Pi (A : U) (B : A → U) : U := Π (x : A), B x
def lambda (A: U) (B: A → U) (b: Pi A B) : Pi A B := $\lambda$ (x : A), b x
def lam (A B: U) (f: A → B) : A → B := $\lambda$ (x : A), f x
def apply (A: U) (B: A → U) (f: Pi A B) (a: A) : B a := f a
def app (A B: U) (f: A → B) (x: A): B := f x

def Π–$\beta$ (A : U) (B : A → U) (a : A) (f : Pi A B)
  : Path (B a) (apply A B (lambda A B f) a) (f a) := idp (B a) (f a)
def Π–$\eta$ (A : U) (B : A → U) (a : A) (f : Pi A B)
  : Path (Pi A B) f ($\lambda$ (x : A), f x) := idp (Pi A B) f

def funext–form (A B: U) (f g: A → B): U := Path (A → B) f g
def funext (A B: U) (f g: A → B) (p: Π (x: A), Path B (f x) (g x))
  : funext–form A B f g := <i> $\lambda$ (a: A), p a @ i

def happly (A B: U) (f g : A → B) (p: funext–form A B f g) (x : A)
  : Path B (f x) (g x) := cong (A → B) B ($\lambda$ (h: A → B), app A B h x) f g p

def funext–$\beta$ (A B: U) (f g: A → B) (p: Π (x: A), Path B (f x) (g x))
  : Π (x: A), Path B (f x) (g x) := $\lambda$ (x: A), happly A B f g (funext A B f g p) x

def funext–$\eta$ (A B: U) (f g: A → B) (p: Path (A → B) f g)
  : Path (Path (A → B) f g) (funext A B f g (happly A B f g p)) p
 := idp (Path (A → B) f g) p
```

Σ-тип

Як ви вже могли помітити система типів прувера порізана на модулі, кожен з яких реалізує певний тип системи типів, а саме 5 правил Мартіна-Лефа: 1) Правило сигнатури або формації, що поселяє тип у певний всесвіт, 2) Конструктори за допомогою яких створюються елементи типу , 3) Елімінатори та/або Індуктори за допомогою яких доводять теореми про тип, 4) Обчислювальне рівняння, що гарантують процес обчислень (бета-правило);

При цьому для додавання типу в систему достатньо додати рівняння патерн-мачингу в набір функцій з яких складається тайп чекер: infer, inferV, app, check, act, conv, eval.

Досніповий код верифікатора для Σ-типу:

```
infer:
  | ESig (a, (p, b)) -> inferTele ctx p a b
  | EFst e -> fst (extSigG (infer ctx e))
  | ESnd e -> let (_, (_, g)) = extSigG (infer ctx e) in g (vfst (eval e ctx))
  | EField (e, p) -> inferField ctx p e

inferV:
  | VFst e -> fst (extSigG (inferV e))
  | VSnd e -> let (_, (_, g)) = extSigG (inferV e) in g (vfst e)

eval:
  | ESig (a, (p, b)) -> let t = eval a ctx in VSig (t, (fresh p, closByVal ctx p t b))
  | EPair (r, e1, e2) -> VPair (r, eval e1 ctx, eval e2 ctx)
  | EFst e -> vfst (eval e ctx)
  | EField (e, p) -> evalField p (eval e ctx)

check:
  | EPair (r, e1, e2), VSig (t, (p, g)) ->
    ignore (extSet (inferV t)); check ctx e1 t;
    check ctx e2 (g (eval e1 ctx)); begin match p with
    | Name (v, _) -> r := Some v
    | Irrefutable -> () end

act:
  | VSig (t, (x, g)) -> VSig (act rho t, (x, g >> act rho))
  | VPair (r, u, v) -> VPair (r, act rho u, act rho v)
  | VFst k -> vfst (act rho k) | VSnd k -> vsnd (act rho k)

conv:
  | VFst x, VFst y | VSnd x, VSnd y -> conv x y
  | VPair (_, a, b), VPair (_, c, d) -> conv a c && conv b d
  | VPair (_, a, b), v | v, VPair (_, a, b) -> conv (vfst v) a && conv (vsnd v) b

and inferField ctx p e = snd (getField p (eval e ctx) (infer ctx e))

let rec getField p v = function
  | VSig (t, (q, g)) ->
    if matchIdent p q then (vfst v, t)
    else getField p (vsnd v) (g (vfst v))
  | t -> raise (ExpectedSig t)

let vfst : value -> value = function | VPair (_, u, _) -> u | v -> VFst v
let vsnd : value -> value = function | VPair (_, _, u) -> u | v -> VSnd v
```

Наш Σ-тип розширений додатковим елімінатором, який дає доступ до іменованого тілесного прямо в процесі нормалізації. Цей механізм дозволяє реалізувати базовий механізм записів-кортежів з іменованими полями, за винятком успадкування та розширення рекордів. Елімінатори .1 та .2 (з cubicaltt) теж працюють.

```
def Sigma (A : U) (B : A → U) : U := summa (x: A), B x
def prod (A B : U) : U := summa (_ : A), B
def pair (A: U) (B: A → U) (a: A) (b: B a) : Sigma A B := (a, b)
def pr₁ (A: U) (B: A → U) (x: Sigma A B) : A := x.1
def pr₂ (A: U) (B: A → U) (x: Sigma A B) : B (pr₁ A B x) := x.2

def Sigma-rec (A: U) (B: A -> U) (C: U) (g: Π (x: A), B(x) -> C)
    (p: Σ (x: A), B x): C := g p.1 p.2

def Sigma-ind (A : U) (B : A -> U) (C : Π (s: Σ (x: A), B x), U)
    (g: Π (x: A) (y: B x), C (x,y)) (p: Σ (x: A), B x) : C p := g p.1 p.2

def ac (A B: U) (R: A -> B -> U) (g: Π (x: A), Σ (y: B), R x y)
  : Σ (f: A -> B), Π (x: A), R x (f x) := (λ(i:A),(g i).1,λ(j:A),(g j).2)

def total (A:U) (B C : A -> U) (f : Π (x:A), B x -> C x) (w: Σ(x: A), B x)
  : Σ (x: A), C x := (w.1,f (w.1) (w.2))

def funDepTr (A: U) (P: A -> U) (a0 a1: A) (p: PathP (<_>A) a0 a1)
    (u0: P a0) (u1: P a1)
  : PathP (<_>U) (PathP (<i> P (p @ i)) u0 u1) (PathP (<_>P a1)
            (hcomp (P a1) 0 (λ (k : I), []) (transp (<i> P (p @ i)) 0 u0)) u1)
 := <j> PathP (<i> P (p @ j \\/ i))
            (comp (λ(i:I), P (p @ j /\\ i)) -j (λ(k: I), [(j =0) -> u0 ])
            (inc (P a0) -j u0)) u1

def pathSig0 (A: U) (P: A -> U) (t u: Σ (x: A), P x) (p: PathP (<_>A) t.1 u.1)
  : PathP (<_>U) (PathP (<i>P (p @ i)) t.2 u.2) (PathP (<_>P u.1)
          (hcomp (P u.1) 0 (λ(k:I),[]) (transp (<i> P (p @ i)) 0 t.2)) u.2)
 := funDepTr A P t.1 u.1 p t.2 u.2
```

0-тип

0-тип є тип-брехня, логічний нуль $\nvdash$, порожнечу, Empty, Void або $\bot$. Використовується для доказу протиріч, містить лише правила формації та індуктор.

Досніповий код верифікатора для 0-типу:

```
eval:
  | EEmpty -> VEmpty
  | EIndEmpty e -> VIndEmpty (eval e ctx)

inferV:
  | VEmpty -> VKan Z.zero
  | VIndEmpty t -> implv VEmpty t

act:
  | VEmpty -> VEmpty
  | VIndEmpty v -> VIndEmpty (act rho v)

conv:
  | VEmpty, VEmpty -> true
  | VIndEmpty u, VIndEmpty v -> conv u v

infer:
  | EEmpty | EUnit
  | EIndEmpty e -> ignore (extSet (infer ctx e)); implv VEmpty (eval e ctx)
```

1-тип

1-тип являє собою логічну одиницю $\nVdash$, тип-істину в інтуїціоністській логіці, Unit або $\top$. Має єдиний конструктор $\star$.

Досніповий код верифікатора для 1-типу:

```
eval:
  | EUnit -> VUnit
  | EStar -> VStar
  | EIndUnit e -> VIndUnit (eval e ctx)

app:
  | VApp (VIndUnit _, x), VStar -> x

inferV:
  | VUnit -> VKan Z.zero
  | VStar -> VUnit
  | VIndUnit t -> recUnit t

act:
  | VUnit -> VUnit
  | VStar -> VStar
  | VIndUnit v -> VIndUnit (act rho v)

conv:
  | VUnit, VUnit -> true
  | VStar, VStar -> true
  | VIndUnit u, VIndUnit v -> conv u v

infer:
  | EStar -> VUnit
  | EIndUnit e -> inferInd false ctx VUnit e recUnit

and recUnit t = let x = freshName "x" in
  implv (app (t, VStar)) (VPi (VUnit, (x, fun x -> app (t, x))))
```

2-тип

2-тип є логічною двійкою $\nvDash$, булевим типом Bool або 0-мірною гомотопічною (без метрики) сферою. Має два конструктори false=0_2 та true=1_2.

Досніповий код верифікатора для 2-типу:

```
eval:
  | EBool -> VBool
  | EFalse -> VFalse
  | ETrue -> VTrue
  | EIndBool e -> VIndBool (eval e ctx)

app:
  | VApp (VApp (VIndBool _, a), _), VFalse -> a
  | VApp (VApp (VIndBool _, _), b), VTrue -> b

inferV:
  | VBool -> VKan Z.zero
  | VFalse | VTrue -> VBool
  | VIndBool t -> recBool t

act:
  | VBool -> VBool
  | VFalse -> VFalse
  | VTrue -> VTrue
  | VIndBool v -> VIndBool (act rho v)

conv:
  | VBool, VBool -> true
  | VFalse, VFalse -> true
  | VTrue, VTrue -> true
  | VIndBool u, VIndBool v -> conv u v

infer:
  | EBool -> VKan Z.zero
  | EFalse | ETrue -> VBool
  | EIndBool e -> inferInd false ctx VBool e recBool

and recBool t = let x = freshName "x" in
  implv (app (t, VFalse)) (implv (app (t, VTrue))
    (VPi (VBool, (x, fun x -> app (t, x)))))
```

W-тип

W-тип призначений для кодування добре визначених дерев. W-тип визначається конструкторах індуктивного дерева, а гілки умови виражені функцією другий залежної компоненти. За допомогою гомотопічної рівності, W-типів, а також типів 0,1,2 виразна індукція натуральних чисел.

Досніповий код верифікатора для W-типу:

```
eval:
  | EW (a, (p, b)) -> let t = eval a ctx in W (t, (fresh p, closByVal ctx p t b))
  | ESup (a, b) -> VSup (eval a ctx, eval b ctx)
  | EIndW (a, b, c) -> VIndW (eval a ctx, eval b ctx, eval c ctx)

app:
  | VApp (VIndW (a, b, c), g), VApp (VApp (VSup (_, _), x), f) ->
    app (app (app (g, x), f),
      VLam (app (b, x), (freshName "b", fun y ->
        app (VApp (VIndW (a, b, c), g), app (f, y)))))

inferV:
  | VSup (a, b) -> inferSup a b
  | VIndW (a, b, c) -> inferIndW a b c

and wtype a b = W (a, (freshName "x", fun x -> app (b, x)))

and inferSup a b = let t = wtype a b in let x = freshName "x" in
  VPi (a, (x, fun x -> implv (implv (app (b, x)) t) t))

and inferIndW a b c = let t = wtype a b in
  implv (VPi (a, (freshName "x", fun x ->
    VPi (implv (app (b, x)) t, (freshName "f", fun f ->
      implv (VPi (app (b, x), (freshName "b", fun b -> app (c, (app (f, b))))))
        (app (c, VApp (VApp (VSup (a, b), x), f))))))))
    (VPi (t, (freshName "w", fun w -> app (c, w))))

act:
  | W (t, (x, g)) -> W (act rho t, (x, g >> act rho))
  | VSup (a, b) -> VSup (act rho a, act rho b)
  | VIndW (a, b, c) -> VIndW (act rho a, act rho b, act rho c)
```

```
conv:
  | VSup (a1,b1), VSup (a2,b2) -> conv a1 a2 && conv b1 b2
  | VIndW (a1,b1,c1), VIndW (a2,b2,c2) -> conv a1 a2 && conv b1 b2 && conv c1 c2

infer:
  | ESup (a, b) -> let t = eval a ctx in ignore (extSet (infer ctx a));
    let (t', (p, g)) = extPiG (infer ctx b) in eqNf t t';
    ignore (extSet (g (Var (p, t))));
    inferSup t (eval b ctx)
  | EIndW (a, b, c) -> let t = eval a ctx in ignore (extSet (infer ctx a));
    let (t', (p, g)) = extPiG (infer ctx b) in
    eqNf t t'; ignore (extSet (g (Var (p, t))));
    let (w', (q, h)) = extPiG (infer ctx c) in
    eqNf (wtype t (eval b ctx)) w';
    ignore (extSet (h (Var (q, w'))));
    inferIndW t (eval b ctx) (eval c ctx)

and inferIndW a b c = let t = wtype a b in
    implv (VPi (a, (freshName "x", fun x ->
      VPi (implv (app (b, x)) t, (freshName "f", fun f ->
        implv (VPi (app (b, x), (freshName "b", fun b -> app (c, (app (f, b))))))
          (app (c, VApp (VApp (VSup (a, b), x), f))))))))
      (VPi (t, (freshName "w", fun w -> app (c, w))))
```

Після того, як ми визначили W типи в ядрі, для реалізації принципу індукції нам знадобиться транспорт у фібраційному шляху Path.

```
def ind^W-β (A : U) (B : A → U) (C : (W (x : A), B x) → U) (g : Π
(x : A)
     (f : B x → (W (x : A), B x)), (Π (b : B x), C (f b)) → C (sup A B x f))
     (a : A) (f : B a → (W (x : A), B x))
  : PathP (<_> C (sup A B a f))
            (ind^W A B C g (sup A B a f)) (g a f (λ (b : B a), ind^W A B C g (f b)))
 := <_> g a f (λ (b : B a), ind^W A B C g (f b))

def trans-W (A : I → U) (B : Π (i : I), A i → U)
     (a : A 0) (f : B 0 a → (W (x : A 0), B 0 x))
  : W (x : A 1), B 1 x
 := sup (A 1) (B 1) (transp (<i> A i) 0 a)
          (transp (<i> B i (transFill (A 0)
                   (A 1) (A j) a @ i) → (W (x : A i), B i x)) 0 f)

def trans-W' (A : I → U) (B : Π (i : I), A i → U)
     (a : A 0) (f : B 0 a → (W (x : A 0), B 0 x))
  : W (x : A 1), B 1 x
 := transp (<i> W (x : A i), B i x) 0 (sup (A 0) (B 0) a f)

def trans-W-is-correct (A : I → U) (B : Π (i : I), A i → U)
     (a : A 0) (f : B 0 a → (W (x : A 0), B 0 x))
  : Path (W (x : A 1), B 1 x) (trans-W A B a f) (trans-W' A B a f)
 := <_> trans-W A B a f

def hcomp-W' (A : U) (B : A → U) (r : I) (a : I → Partial A r)
     (f : Π (i : I), PartialP [(r = 1) → B (a i 1=1) → (W (x : A), B x)] r)
     (a₀ : A[r ↦ a 0]) (f₀ : (B (ouc a₀) → (W (x : A), B x)) [r ↦ f 0])
  : W (x : A), B x
 := hcomp (W (x : A), B x) r
            (λ (i : I), [(r =1) → sup A B (a i 1=1) (f i 1=1)])
            (sup A B (ouc a₀) (ouc f₀))
```

Path-тип

Нарешті багатовимірний Path тип є та гомотопічна кубічна гетерогенна рівність за допомогою якої можна побудувати групоїди (дивіться базову бібліотеку Андерса).

```
eval:
  | EPathP e -> VPathP (eval e ctx)
  | EPLam e -> VPLam (eval e ctx)

check:
  | EPLam (ELam (EI, (i, e))), VApp (VApp (VPathP p, u0), u1) ->
    let v = Var (i, VI) in let ctx' = upLocal ctx i VI v in
    let v0 = eval e (upLocal ctx i VI vzero) in
    let v1 = eval e (upLocal ctx i VI vone) in
    check ctx' e (appFormula p v); eqNf v0 u0; eqNf v1 u1

inferV:
  | VPLam (VLam (VI, (_, g))) -> let t = VLam (VI, (freshName "ι", g >>
inferV)) in
    VApp (VApp (VPathP (VPLam t), g vzero), g vone)
  | VAppFormula (f, x)       -> let (p, _, _) = extPathP (inferV f) in appFormula p x
  | VPathP p -> let (_, _, v) = freshDim () in let t = inferV (appFormula p v) in
    let v0 = appFormula p vzero in let v1 = appFormula p vone in implv v0 (implv v1 t)

act:
  | VPLam f -> VPLam (act rho f)
  | VPathP v -> VPathP (act rho v)
  | VAppFormula (f, x)   -> appFormula (act rho f) (act rho x)


and inferPath ctx p =
  let (_, t0, t1) = extPathP (infer ctx p) in
  let k = extSet (inferV t0) in implv t0 (implv t1 (VKan k))

and appFormula v x = match v with
  | VPLam f -> app (f, x)
  | _       -> let (_, u0, u1) = extPathP (inferV v) in
    begin match x with
      | VDir Zero -> u0
      | VDir One  -> u1
      | i -> VAppFormula (v, i)
    end
```

```
conv:
  | VPLam f, VPLam g -> conv f g
  | VPLam f, v | v, VPLam f ->
    let (_, _, i) = freshDim () in conv (appFormula v i) (app (f, i))
  | VPathP a, VPathP b -> conv a b

infer:
  | EPathP p -> inferPath ctx p
  | EPLam (ELam (EI, (i, e))) ->
    let ctx' = upLocal ctx i VI (Var (i, VI)) in ignore (infer ctx' e);
    let g = fun j -> eval e (upLocal ctx i VI j) in
    let t = VLam (VI, (freshName "ι", g >> inferV)) in
    VApp (VApp (VPathP (VPLam t), g vzero), g vone)
  | EPLam _ -> raise (InferError e)
  | VAppFormula (f, x), VAppFormula (g, y) -> conv f g && conv x y
```

Маючи Π, Σ та $\mathbf{Path}$ типи та урізаний транспорт можна побудувати обчислювальну семантику MLTT-73:

```
def MLTT (A : U) : U1 := Σ
  (Π-form   : Π (B: A → U), U)
  (Π-ctor1  : Π (B: A → U), Pi A B → Pi A B)
  (Π-elim1  : Π (B: A → U), Pi A B → Pi A B)
  (Π-comp1  : Π (B: A → U) (a: A) (f: Pi A B),
                   Equ (B a) (Π-elim1 B (Π-ctor1 B f) a) (f a))
  (Π-comp2  : Π (B : A → U) (a : A) (f : Pi A B),
                   Equ (Pi A B) f (λ (x : A), f x))
  (Σ-form   : Π (B: A → U), U)
  (Σ-ctor1  : Π (B: A → U) (a: A) (b : B a) , Sigma A B)
  (Σ-elim1  : Π (B: A → U) (p: Sigma A B), A)
  (Σ-elim2  : Π (B: A → U) (p: Sigma A B), B (pr1 A B p))
  (Σ-comp1  : Π (B: A → U) (a: A) (b: B a), Equ A a (Σ-elim1 B (Σ-ctor1 B a b)))
  (Σ-comp2  : Π (B: A → U) (a: A) (b: B a), Equ (B a) b (Σ-elim2 B (a, b)))
  (Σ-comp3  : Π (B: A → U) (p: Sigma A B),
                   Equ (Sigma A B) p (pr1 A B p, pr2 A B p))
  (=-form   : Π (a: A), A → U)
  (=-ctor1  : Π (a: A), Equ A a a)
  (=-elim1  : Π (a: A) (C: D A) (d: C a a (=-ctor1 a)) (y: A) (p: Equ A a y),
             C a y p)
  (=-comp1  : Π (a: A) (C: D A) (d: C a a (=-ctor1 a)),
                   Equ (C a a (=-ctor1 a)) d (=-elim1 a C d a (=-ctor1 a))), 1

theorem internalizing (A : U) : MLTT A :=
  (Pi A, lambda A, app A, comp1 A, comp2 A,
   Sigma A, pair A, pr1 A, pr2 A, comp3 A, comp4 A, comp5 A,
   Equ A, refl A, J A, comp6 A, A)
```

13 Суперпростір

В цій статті у формі посилань на статті дається визначення суперточки, суперлінії, суперсфер та супермноговидів, їх розшаруваня та когомології. Слово «супер» означає $\mathbb{Z}_2$ градуйовані супералгебри Лі, які містять два вектори координат, з парними та непарними індексами та використовуються в теорії суперструн. Показується еволюція емерджентного суперпростору та всіх його струнних теорій (Type I, IIA, IIB, SO(32), $E_8 \times E_8$).

Супералгебри Лі

Супер-алгебри Лі як необхідний пререквізит суперсиметричних бозонно-ферміонних геометрій. Категорно, супералгебра Лі — це внутрішній об'єкт в симетричній моноїдальній категорії (SMC) $\mathbb{Z}_2$ градуйованих суперпросторів $\mathrm{sVect} = (\mathrm{Vect}_{\mathbb{Z}_2}, \otimes_k, \tau, \mathfrak{g})$. Морфізми в цих категоріях — дужки Лі, такі що: 1) $[a, b] = -(-1)\alpha\beta[b, a]$; 2) $[a, [b, c]] = [[a, b], c] + (-1)\alpha\beta[b, [a, c]]$. $\alpha\beta \in \mathbb{Z}_2, a \in V_1, b \in V_2, \mathrm{Vect}_{\mathbb{Z}_2} = V_1 \otimes V_2$.

Суперсфери та розшарування Хопфа

Крім суперточки в $\mathbb{Z}_2$ градуйованих, [точніше над градуйованими просторами які визначаються прямим декартовим добутком цілих чисел] суперсиметричних алгебрах (супералгебрах) Лі нас будуть цікавити класичні розшарування Хопфа, та суперсфери з їх використанням. При цьому форма точних послідовностей які визначають розшарування не змінюються, але змінюються їх когомології.

Так, в нас є проблема з відсутністю $\mathrm{SL}(2)$ на октаніонах, тому в супергеометрії ми використовуємо накриваючі групи спінів лоренцевих груп в сигнатурі Мінковського (9,1) для останнього розшарування Хопфа.

Супермноговиди

Окрім суперточки, суперлінії, суперсфер, нас будуть цікавити також супермноговиди довільної форми.

Когомології де Рама

За теоремою де Рама існує ізоморфізм між групами когомологій де Рама $H^k_{dR}(M)$ та групами когомологій $H^k(M; \mathbb{R})$ для будь-якого гладкого многовида. З комутативної діаграми ізоморфізмів випливає, що спектральні послідовності, що ґрунтуються на гомологічних групах сфер, можуть бути адаптовані до суперсфер після спеціалізації на конкретному полі коефіцієнтів.

Когомології Шевал'є-Ейленберга

Традиційно перед визначенням когомологій типу Шевал'є-Ейленберга, спочатку дають визначення алгебрам Шевал'є-Ейленберга $\mathrm{CE}(\mathfrak{g})$ як супералгебрам Грасмана в дуальному суперпросторі $\wedge\, \partial\mathfrak{g}*$, що має диференціал $d\mathfrak{g} := [\_, \_]* : \mathfrak{g}* \to \mathfrak{g} * \wedge \mathfrak{g}*$, що розширюється на весь дуальний простір за допомогою градуйованості Лейбніца.

Когомології BRST

У фізиці, супералгебри Шевал'є-Ейленберга $\mathrm{CE}(\mathfrak{g}, N)$ дії алгебри Лі або L-∞ алгебри групи калібрування G на простір полів N називається BRST комплексом на честь Беккі, Руе, Стора, Тютіна.

Емерджентний суперпростір

Суперточка $\mathbb{R}^{0|N}$ визначається для всіх N. Суперточка $\mathbb{R}^{0|1}$ має природнє розширення до суперлінії $\mathbb{R}^{1|1} = \mathbb{R}^{1,0|1}$. Максимальний інваріант центрального розширення суперточки $\mathbb{R}^{0|2}$ є трьохвимірна сумер-алгебра Мінковського $\mathbb{R}^{2,1|2}$.

Далі простір розвивається по сферам згідно конструкції Келі-Діксона та розшарувань Хопфа, набуваючи свого повного змісту у об'єднуючій М-теорії:

1). $\mathbb{R}^{2,1|2} \to \mathbb{R}^{2,1|2+2}$;
2). $\mathbb{R}^{3,1|4} \to \mathbb{R}^{3,1|4+4}$;
3). $\mathbb{R}^{5,1|8} \to \mathbb{R}^{5,1|8+8}$;
4). $\mathbb{R}^{9,1|16} \to \mathbb{R}^{9,1|16+16}$.

IIB $\to \mathbb{R}^{9,1|16+16} \leftarrow \mathbb{R}^{9,1|16} \to \mathbb{R}^{9,1|16+16} \leftarrow$ IIA. Максимальний інваріант центрального розширення простору Мінковського IIA типу $\mathbb{R}^{9,1|16+16}$ є $\mathbb{R}^{10,1|32}$ — 11-вимірна М-теорія з тридцятьма двома додатковими ферміонними параметрами.

14 Теорії Янга-Міллса

З точки зору алгебраїчної топології.

Електромагнетизм (Фотон)

Теорія квантової електродинаміки розвинулася в 1930—1940-х рр., де унітарна група перетворень відіграє головну роль $U(1)$. Шредінгер показав, що група $U(1)$ викликає фазовий зсув $e^{i\theta}$ в електромагнітному полі, що відповідає збереженню електричного заряду, зокрема при розповсюдженні світла.

Електромагнітне поле може бути описано як вектор потенціал $A\mu$ і тензор $F\mu\nu$. Зв'язок між групою $U(1)$ і перетвореннями Лоренца полягає в тому, що калібрувальне перетворення електромагнітного потенціалу $A\mu$ при $U(1)$ аналогічне перетворенню просторово-часових координат при перетвореннях Лоренца. В обох випадках ці перетворення гарантують, що основна фізика залишається незмінною.

Теорії Янг-Міллса

Калібровочна теорія поля — це тип теорії поля де Лангранжін, а значить і динамічна система загалом, є інваріантним відносно локальних трансформацій згідно певного гладкої сім'ї операторів (Груп Лі).

Теорія Янга-Міллса – це калібровочна квантова теорія поля, де головну роль відіграє спеціальна унітарна група $SU(n)$, або більш загально, довільна компактна група Лі.

Слабка взаємодія (W і Z бозони)

Група $\mathsf{SU}(2)$ формалізує інваріант ізоспіна при колізіях, спричиненими сильними взаємодіями. Многовид S^3 є дифеоморфізмом до групи $\mathsf{SU}(2)$, який показує, що $\mathsf{SU}(2)$ (многовид) є однозв'язним і що S^3 може бути наділений структурою компактної зв'язної групи Лі.

Сильна взаємодія

Квантова хромодинаміка є неабелевою калібровочною теорією поля на локальній (калібрувальній) групі симетрії під назвою $\mathsf{SU}(3)$. Її топологічну структуру можна зрозуміти зауваживши, що $\mathsf{SU}(3)$ діє транзитивно на одиничній сфері S^5 у $\mathbb{C}^3 \cong \mathbb{R}^6$. Стабілізатор довільної точки сфери ізоморфний до $\mathsf{SU}(2)$, яка топологічно є 3-сферою. Це показує, що SU(3) є розшаруванням над базою S^5 з розшаруванням S^3. Так як розшарування і бази просто-з'єднані, тоді просто-зв'язність $\mathsf{SU}(3)$ випливає з стандартного топологічного результату (довга точна послідовність гомотопічних груп для пучків розшарувань).

15 Формальна Йогачара

Вступна частина

У той час як буддизм Yogācāra досить добре відомий академічним дослідникам та спеціалістам буддистських студій, він все ще в основному невідомий звичайним буддистам азіатських країни, а також практикуючим буддистам та іншим неспеціалістам студентам на Заході. Чому так? Перш за все, незважаючи на величезний вплив Йогачари в період становлення буддизму махаяни в Індії школа вимерла разом із буддизмом загалом, до кінця першого тисячоліття. в Тибеті, незважаючи на свій вплив, в Школа ніколи не існувала як окрема традиція, а лише у складі Абхідхарми (Трипітака) Пімапа Рінпоче (Gateway To Knowledge). У Східній же Азії, Йогачара справді існувала як окрема традиція, але для практичних цілей майже перестала мати будь-який великий вплив після першого тисячоліття нашої ери.

Незважаючи на її остаточне зникнення як незалежної школи, Yogācāra як вчення про карму, медитацію, пізнання та теорію шляху мали силу — повний вплив на інші школи махаяни, що розвинулися в той час імпорту Йогачари до Тибету та Східної Азії, так що більшість технічної термінології, на якій засновані інші школи махаянського дискурса, була поглинена з різних напрямів Yogācāra.

Відсутність розвитку школи йогачари в Тибеті головним чином була внаслідок того, що вона була поглинена новоствореним Тибетськими доктринальними школами. у Східній Азії, з іншого боку, Йогачара певний час існували як незалежна секта, відома китайською як Weishi (лише свідомість) або Faxiang (характеристика дхарми). Але школа зрештою вимерли перед різними формами конкуренції з (1) доктринальними школами, чиї вчення вважалися більш резонансними зі східноазіатським світоглядом і (2) більш масово орієнтовані школи, такі як Школи чистої землі та медитації (чан/сон/дзен), які пропонували форму навчання та практики набагато легше сприйняті звичайним людям мирянин віруючий.

Найбільша перешкода Yogācāra для набуття широкої популярності полягала у складності її громіздкої системи точок зору, шляхів і категорії, пояснені складною технічною термінологією. це, дійсно, вимагають досить значного ступеня відданості з боку студента досягти рівня базового розуміння, достатнього для читання та розуміння священних писанн Йогачари.

Однак є деякі, хто стверджує, що ця передбачувана проблема здатності до розуміння Йогачари також може значною мірою полягати в манері презентації, що і стало основною мотивацією Тагава Шун'єі. Тобто, незважаючи на здавалося б громіздку складність Система Yogācāra, про що говорять майстри Yogācāra у багатьох кейсах — це легко впізнавані повсякденні переживання, якими ми всі ділимося. Багато моментів, на яких зосереджувалися майстри Йогачари, були речами, які ми всі сприймаємо як належне, але для якого, якщо розглянути більш детально, ми насправді немає пояснень. і в більшості випадків — я вважаю, що ми можемо додати — багато з цих питань, для яких дослідники в таких галузях, як сучасна психологія, фізіологія, хімія і фізика ще не мають відповіді.

Нас, звичайно, з дитинства вчили тому, що пам'ять зберігається десь у мозку. Якщо це правда, то з мозок складається з фізичної матерії, хіба це не так, як ми продовжуйте додавати інформацію, активність мозку має збільшуватися, щоб тренувати це? Звичайно, ні. Але тоді де вся ця концептуальна інформація, що зберігається, навіть не кажучи про інформацію, що стосується тілесних активностей.

Очевидною відповіддю на це запитання є те, що ця інформація зберігається десь у «свідомості». але якщо це так, то де це в розумі великий обсяг інформації, що зберігається? І як ми знаємо, що ми ні постійно втрачає інформацію? і якщо ми його зберігаємо, як саме ми його отримуємо, коли нам це потрібно? Для більшості відповідей, відповідь: «ну, ми точно не знаємо».

Для авторів школи Йогачара така відповідь не була прийнятною, і тому вони прагнули через свої дослідження, дослідження та коншаблонні техніки для надання деяких відповідей, а також широкий спектр суміжних, і навіть більш фундаментальних питання.

На цьому етапі слід зазначити, що мотивація для Дослідники Yogācāra були не просто творінням ранніх індійців буддійських еквівалент сучасної когнітивної або поведінкової психології.

Асанга, Васубанду (Автори Йогачари) та їхні колеги були релігійними мислителями, вимушено, через очевидні протиріччя та доктринальні складності, властиві буддійським поясненням природи людського розуму, зіставлення з процесами, які ведуть або до просвітлення, або до глибшого захоплення через незнання та страждання — пробували знайти якісь рішення, які були би раціонально сприйняті. У процесі розробки таких рішень (успадковуючи давню традицію філософії перетворення розуму, надані попередніми вченими) їм довелося зробити дуже серйозне дослідження, як саме ми знаємо речі, і як, точно, наші тіла та розум змінюються та розвиваються. Маючи справу з подібними проблемами, вони не могли не зіткнутися з тими ж проблемами, з якими зустрічаються сучасні філософи, психологи і навіть еволюційні біологи.

І саме з цієї причини Йогачара прийшла у наш час аби привернули інтереси різноманітних інтелектуалів, людей, чия робота лежить поза сферою релігійної віри, які вивчають проблеми у пізнанні, поведінці людини, розвитку особистості, тощо.

Врешті-решт, проблеми, з якими займаються йогачара — це буддійські проблеми, наскрізь, і, таким чином, зрозуміти мотивації, що стоять за працями цих мислителів, мабуть, буде корисно надати короткий огляд розвитку цих проблем.

Насіння

Я хотів би попередити, що будь-яка відповідність між п'ятьма ефектами-/зернами філософії Йогачари та п'ятьма теоріями суперструн є спекулятивною і не базується на жодній загальноприйнятій чи усталеній теорії. Однак, якби ми вели спекулятивне листування на основі роботи Девіда Фу, то отримали би такі результати.

Формальна Йогачара є мненомічним альтернативним езотеричним індексом-показчиком сучасної теоретичної математичної фізики яка базується на еквіваріантній теорії гомотопій у відповідності до тибетських термін школи Йогачари, яка збереглася в університеті Наланди в Тибеті. Терміни йогачари базуються на ранніх перекладах Сутр трипітаки, основні контрибютори Йогачари були брати Асанга та Васубандху в Абхідхарма-коші та коментареві. Обширні коментарі на Йогачару та Абхідхарму були залишені Міпамом Рінпоче в традиції Кама, який вважається еманацією Манджушрі.

У цьому випуску даються мнемонічні недоказові відповідності між поняттями йогачари та сучасної М-теорії (класифікація п'яти суперструнних теорій).

В коментарях Абхідхарма-коша-бхася про «саманартху» та «паратантру» розказується в главі 5, Термін «васана» використовується в главі 6, і термін «ніродха» в главі 7.

Терміни «саманартха» і «паратантра» вживаються у Махаяна-сутраламкара-каріці у віршах 2.9-2.11, а термін «парінішпанна» вживається у вірші 2.14. Термін «васана» не використовується в Mahāyāna-sūtrālamkāra-kārikā, але це загальноприйнятий термін у філософії Йогачари. Термін «ніродха» вживається в контексті припинення страждань у вірші 2.43.

Samanartha

rang bzhin gnas dang skor bar dgos pa

Саманартха відноситься до «взаємно підтримуючих» або «взаємопов'язаних» аспектів існування, підкреслюючи взаємозалежність усіх явищ.

Той самий порядок (санскр. samanartha) — відповідає теорії суперструн типу IIB, яка є теорією суперсиметричних струн, які мають властивість самодуальності. Властивість самодуальності означає, що той самий тип струни може трансформуватися в інший тип струни за певних умов, вказуючи на те, що всі частинки та сили у Всесвіті виникають із тих самих базових мод коливань струн.

Paratantra

gnas su sdud pa'i gros pa'i sems dang byin rlabs pa

Паратантра термін відноситься до «залежної» або «відносної» природи всіх явищ, які розглядаються як такі, що виникають в залежності від різних причин і умов.

Створено людиною (санскр. paratantra) — відповідає теорії суперструн типу IIA, яка також є теорією суперсиметричних струн. У цій теорії властивості та поведінка частинок і сил залежать від контексту, в якому вони виникають, подібно до ідеї паратантри у філософії Йогачари.

Parinispanna

mngon par byed pa

Паріністанна термін стосується «повністю усвідомленої» або «просвітленої» природи явищ, які розглядаються як за своєю суттю чисті та вільні від оман его та дуалістичного мислення.

Непереверщеність (санскр. parinispanna) — відповідає гетеротичній теорії струн E_8xE_8, яка описує остаточну реальність Всесвіту за допомогою математичних і груп симетрії, які лежать в основі поведінки струн. Група симетрії E_8xE_8 є найбільшою можливою групою симетрії, яка може описати поведінку струн.

Vasana

bar do dga' ba

Васана термін стосується «звичних тенденцій» або «кармічних відбитків», які формують наше сприйняття та досвід, що призводить до прихильності та страждань.

Дозрівання (санскр. васана) — відповідає теорії гетеротичних струн SO(32), яка є теорією суперсиметричних струн, що враховує історію взаємодій і попередні стани частинок і сил, подібно до ідеї васани у філософії Йогачара.

Nirodha

rnam par 'das pa

Цей термін стосується «припинення» або «знищення» страждання та причин страждання, що є кінцевою метою практики Йогачари.

Нірвана (санскр. nirodha) — відповідає теорії суперструн типу I, яка містить відкриті і закриті струни, що описує остаточну природу реальності через розуміння та споглядання математичних принципів, які керують поведінкою струн. Цю теорію можна розглядати як більш просту та фундаментальну версію теорій суперсиметричних струн, подібну до ідеї ніродхи у філософії Йогачари.

Формальна частина

Думав про локальну гомотопічну теорію статтю написати ще і про хроматичну теорію гомотопій та індексовані p-адичними числами K-теорії Морави.

Ну і ще про модальні категорії. Так само, як похідні категорії є категорною семантикою когомологічних типів, так само локальна теорія Жардіна і теорії Морави є категорною семантикою гомотопічних типів. Вони є основними інструментами, що пропонують градуальний спуск і локалізацію (ітеративний процес), у випадку когомології, похідні категорії — інформацію про виколоті многовиди, векторні розшарування та інтегрування, а у випадку гомотопій, модальні категорії та локальна гомотопічна теорія — інформацію про топологію простору та його гомотопічний тип, що теж обчислюється ітеративно.

Ādarśa-jñāna

mdangs su gso ba shes rab gzhan po

Найабстрактніша категорія, яка охоплює будь-які категорії логіки, у тому числі топосо-теоретичні з локалізацією відкритих підпросторів в підкласифікаторі об'єктів є категорія комутативних діаграм — воістину найчистіший та найбстрактніший простір мислення, у якому написані (або закодовані) усі математичні формули у сучасній (абстрактній) математиці. Це — найабстрактніший чистий листок, табула раса, відкритий гіперкуб, де вершини — це топоси, а стрілки — геометричні морфізми (пара спряжених функторів між топосами), простір для математичної творчості, необмежений конкретними числами чи груповими представленнями. Саме на цьому «листку» працює мислення математика. Це — перший рівень абстрактності.

Інфініті категорії, групоїди, топоси, стеки. Поняття інфініті категорій та інфініті групоїдів є основним та найбільшохоплюючим в теорії категорій. Це такі багаторівневі категорії категорії, де всі композиції у всіх всесвітах є когерентно-узгодженими. Категоріальний аналіз включає у себе насупні стадії-конструктції від поняття самої категорії до понятя спряженої трійки: 1) Категорії (C); 2) Функтори (F); 3) Природні перетворення (N); 4) Спряжені пари (P); 5) Спряжені еквівалентності (E); 6) Спряжені трійки (T); Загальна вежа:

$$\mathsf{C} \to \mathsf{F} \to \mathsf{N} \to \mathsf{P} \to \mathsf{E} \to \mathsf{T}.$$

Категорії діаграм. Нехай C — мала категорія, а D — будь-яка категорія. Функтор $\mathsf{F} : \mathsf{C} \to \mathsf{D}$ можна розглядати як діаграму в D з індексом C. Категорія діаграм в D з індексом C, позначена D^{C}, має як об'єкти всі функтори від C до D, а як морфізми всі природні перетворення між цими функторами. Тобто, для двох діаграм $\mathsf{F}, \mathsf{G} : \mathsf{C} \to \mathsf{D}$ морфізм від F до G є природним перетворенням $\alpha : \mathsf{F} \to \mathsf{G}$, яке пов'язує кожному об'єкту c у C морфізм $\alpha_{\mathsf{c}} : \mathsf{F}(\mathsf{c}) \to \mathsf{G}(\mathsf{c})$ у D таким чином, що для кожного морфізму $\mathsf{f} : \mathsf{c} \to \mathsf{c}'$ у C діаграма комутує:

$$\begin{array}{ccc} \mathsf{F}(\mathsf{C}) & \xrightarrow{\alpha_{\mathsf{c}}} & \mathsf{G}(\mathsf{C}) \\ {\scriptstyle \mathsf{F}(\mathsf{f})}\downarrow & & \downarrow{\scriptstyle \mathsf{G}(\mathsf{f})} \\ \mathsf{F}(\mathsf{C}') & \xrightarrow{\alpha_{\mathsf{c}'}} & \mathsf{G}(\mathsf{C}') \end{array}$$

Локальні декартово-замкнені категорії. Категорії зі скінченними лімітами, де слайс-категорії по будь-якому об'єкту є декартово-замкненими називаються LCCC, та моделюють фібраційне лямбда-числення як внутрішню мову таких категорій.

Симетричні моноїдальні категорії. Моноїдальні категорії мають додаткову структуру яка складається з: 1) тензорного добутку $\otimes : \mathsf{M} \times \mathsf{M} \to \mathsf{M}$;

2) об'єкту одиниці; 3) природнього-перетворення «асоціатора» $a_{x,y,z} : (x \otimes y) \otimes z \to x \o times(y \otimes z)$; 4) правила трикутника та п'ятикутника.

Сплетена моноїдальна категорія — це категорія з додатковим природнім перетворенням яке називається «сплетіння» $B_{x,y} : x \otimes y \to y \otimes x$, таким, що комутують дві діаграми:

Слетена моноїдальна каегорія, сплетення якої задовільняє рівняння $B_{x,y} = B^{-1}_{y,x}$ для всіх об'єктів x, y називається симетричною моноїдальною категорією.

Samatā-jñāna

dus pa shes rab gzhan po

Потім з цього листка ми ітеративно інстанціюємо спочатку абстрактні алгебраїчні інструменти, такі як прикладні теорії категорій орисані вище, або шість йог Гротендіка, симетричні моноїдальні категорії, модальні спряження (трійки), тощо. Це — другий рівень «абстрактності».

Абелеві категорії — це збагачене поняття категорії Сандерса-Маклейна поняттями нульового об'єкту, що одночасно ініціальний та термінальний, властивостями існування всіх добутків та кодобутків, ядер та коядер, а також, що всі мономорфізми і епіморфізми є ядрами і коядрами відповідно (тобто норомальними).

Похідні категорії. Похідна категорія — це конструкція в гомологічній алгебрі, яка пов'язує з даною категорією комплексів нову категорію, в якій ідентифікуються квазіізоморфні комплекси. Зокрема, задана категорія C комплексів над комутативним кільцем R, будується нова категорія, позначена D(C), об'єкти якої такі ж, як об'єкти C, але в якій морфізми визначені більш гнучким способом. Зокрема, морфізми в D(C) — це класи еквівалентності діаграм морфізмів у C, які називаються квазіізоморфізмами, які задовольняють певним аксіомам. Ці аксіоми включають існування певних виділених трикутників і сумісність із прямими сумами.

Модельні категорії. Категорія моделі Квіллена — це категорія, оснащена певною структурою, яка дозволяє створити гомотопічну теорію об'єктів у цій категорії. Більш конкретно, це категорія з класом слабких еквівалентностей, класом розшарувань і класом корозшарувань, які задовольняють набір аксіом, відомих як аксіоми категорії моделі Квіллена. Слабкі еквівалентності в категорії моделі Квіллена є класом відображень, які індукують ізоморфізми на певних гомотопічних групах. Розшарування є класом відображень, які задовольняють певну властивість підняття щодо корозшарувань, тоді як корозшарування є класом відображень, які задовольняють певну властивість підняття щодо розшарувань.

Категорії спектрів. Спектр — це послідовність точкових просторів X_0, X_1, X_2, ... разом із відображеннями $\sigma_n : X_n \to \Sigma X_{n+1}$, де Σ позначає суспензію простору, що задовольняє деяким аксіомам, які гарантують, що σ_n узагальнюють звичайні карти суспензій. Морфізм спектрів $f : X \to Y$ є послідовністю відображень $f_n : X_n \to Y_n$, які комутують із відображеннями суспензій в тому сенсі, що комутує діаграма:

$$\begin{tikzcd}
X_n \arrow[r, "\sigma_n"] \arrow[d, "f_n"'] & \Sigma X_{n+1} \arrow[d, "\Sigma f_{n+1}"] \\
Y_n \arrow[r, "\sigma_n"] & \Sigma Y_{n+1}
\end{tikzcd}$$

Т-Спектри та досніпи спектрів. Для потреб стабільного теорії гомотопії та для схемного узагальнення різних видів спектрів: Адамса, Ейленберга-Мак Лейна, К-теорій, в тому числі для потреб $\mathbb{A}^1$-теорії гомотопій, локальної гомотопічної теорії, тощо.

Функторіальні йоги. Перші шість йог Гроендіка визначають наступні спряжені функтори: геометричні морфізми, похідного функтора образу прямого пучка та оберненого образу, дуальності Верд'є, та дуальне спряження тензорного добутку в симетричних моноїдальних категоріях та внутрішнього функтора Hom. Такі йогічні вправи дали змогу побудувати похідні категорії, та виділити системи коефіцієнтів і їх узагальнення: $D(X, \mathbb{Q})$ — l-адичні пучки і l-адична когомологія; $D(X(\mathbb{C}), \mathbb{Z})$ — аналітичні пучки і когомології Бетті; $D(\mathcal{D}_X)$ — голономічні $D\mathcal{D}$-модулі і когомології де Рама; $D(\mathrm{Coh}(X))$ — когерентні пучки та когерентні когомології; $DM(X)$ — мотивні пучки і 0-зважені мотивні когомології; $SH(X)$ — стабільні мотивні гомотопічні пучки і стабільні мотивні 0-зважені групи когомотопій.

Функторіальні йоги і пошуки спряжень Ловіра відкривають наступні спряженя в інфініті зв'язаних топосах, або спряжені трійки ($F \dashv G \dashv H$): 1) $\exists \dashv f^{-1} \dashv \Pi$; 2) $\Sigma \dashv f_\star \dashv \exists$; 3) $\Sigma \dashv f_\star \dashv \Pi$; 4) монадичні і комонадичні дуальності; 5) спряження Квілена; 6) спряження Фробеніуса; 7) $\mathrm{sh} \dashv \flat \dashv \sharp$; 8) $\Pi \dashv \mathrm{Disc} \dashv \Gamma \dashv \mathrm{coDisc}$; 9) $\mathfrak{R} \dashv \mathfrak{I} \dashv \&$; 10) $\rightrightarrows \dashv \rightsquigarrow \dashv \mathrm{Rh}$.

Pratya-veksanā-jñāna

rang gi shes rab gzhan po

Потім самі теорії, які виражується вже як інстанції формул попереднього, другого рівня абстрактності, наприклад теорії (ко-)гомологій та (ко-)гомотопій. Це — третій рівень «абстрактності».

Теорія гомотопій. Категорія моделей для теорії гомотопій Ho(C) — це категорія, де об'єкти це гомотопічні типи, а морфізми це гомотопії між цими типами. Категорія моделей має зв'язок з теорією гомотопій за допомогою забуваючого функтора, який переводить кожен гомотопічний тип на його модель. Теорія гомотопій на C — це категорія моделей для теорії гомотопій Ho(C), де Ho(C) є гомотопічною категорією, отриманою шляхом локалізації категорії C з морфізмами, які стають ізоморфізмами в категорії гомотопічних функторів.

Гомологічна алгебра. Алгебраїчна структура, яка взаємозв'язана з топологічною теорією і використовується для вивчення властивостей топологічних просторів. У категорійному підході, гомологічна алгебра визначається як об'єкт у категорії кольцевих об'єктів в категорії модулів над деякою категорією або топологічним простором. Зазвичай, це є категорія модулів над комутативним кільцем.

Алгебраїчна геометрія. Формально, алгебраїчна геометрія може бути визначена як вивчення категорії афінних алгебраїчних множин та алгебраїчних відображень між ними. Ця категорія побудована на основі категорії комутативних алгебр над заданим полем. Алгебраїчна геометрія вивчає структуру цієї категорії, таку як топологію, властивості ідеалів та їх геометричні властивості, а також вивчає властивості алгебраїчних відображень, таких як внутрішні гомоморфізми та проективні відображення.

Диференціальна геометрія. Функтор який приписує кожному гладкому многовиду M комутативну алгебру H(M), яка залежить від топологічних та диференціальних властивостей многовиду і зберігає зв'язок між гладкими відображеннями многовидів та алгебрними гомоморфізмами між відповідними комутативними алгебрами називється кобордизмом. Цей функтор відображає категорію гладких многовидів у категорію комутативних алгебр, які звуться алгебрами Хопфа.

Категоріальна супергеометрія. Або теорія супер формальних гладких інфініті групоїдів на супер формальних Картанових просторах:

Krtyānusthāna-jñāna

byed par phyin pa'i shes rab gzhan po

Потім йде четвертий рівень «абстрактності» — прикладні теорії з просторами та алгебрами, або бозонно-ферміонними суперпросторами та супераогебрами Лі. Теорія суперструн, квантова топологічна теорія поля.

Топологічна квантова теорія поля. Топологічна квантова теорія поля, яка є розділом математичної фізики, має справу з математичною структурою топологічних просторів.

Супергеометрія М-теорії: не-пертурбативне доповнення всіх струнних теорій.

Йогачара і Математика

В тибетському буддизмі основними текстами по Йогачарі вважуються тексти Міпама Рінпоче. Згідно канону, свідомості (rnam shes) Йогачари, яких є 8, визначаються так:

1) (mig, око);
2) (rna, вухо);
3) (sna, ніс);
4) (lce, смак);
5) (lus, тіло);
6) (yid, логіка);
7) (nyon yid, алогіка);
8) (kun gzhi, основа).

8-й вид мислення. Усі конструкції наведені в цій статті розповсюджуються на 4 рівня очищення само-усвідомлення для 8-ї свідомості алаявіджняни. Якщо мнемонічно кодувати, або езотеризувати цю модель, то вийде: 1. Основа примордіального (табула раса); 2. Основа звільнення (другий рівень абстрактності); 3. Основа загального (геометричні теорії); 4. Основа ілюзій (теорія суперструн, або нісіння причини та наслідку).

1) Гедоністичний (pramuditābhūmi; rab tu dga' ba).
2) Міцний (vimalābhūmi; dri ma med pa).
3) Ілюмінуючий (prabhākarībhūmi; 'od byed pa).
4) Радіантний (arcismatībhūmi; 'od 'phro can).
5) Незворотній (sudurjayābhūmi; shin tu sbyang dka' ba).
6) Ясний (abhimukhībhūmi; mngon du gyur ba).
7) Прогресуючий (durangamabhūmi; ring du song ba).
8) Непорушний (acālabhūmi; mi g.yo ba).
9) Неперевершений (sādhumatībhūmi; legs pa'i blo gros).
10) Хмари Дхарми (dharmameghaābhūmi; chos kyi sprin).

В тибетській традиції Йогачари, чотири з останніх 10 бхумі шляху Бодхісатви відповідають чотирьом рівнями само-усвідомлення Алая-віджняни: 8-й вид мислення це онтолічний мисленний процес який покриває усі можливі абстрактні і прикладні математики, тому його модель — це одна з можливих індексацій бібліотеки формальної математики.

7-й вид мислення це мислення з bottom, яким може скічитися довільне обчислення довільного морфізму, або мислення з парадоксами, або мислення з Fixpoint, самовимотуючий процес мислення. Формальні теорії з парадоксами. Розглядається вбудовування в категорях повних частково-впорядкованих множин. В сутності — це всі мови з неконтрольовану рекурсію, теорії рекурсивних функцій, тощо. Більшість неформальних (просто формалізуємих) інтерпритаторів з неконтрольовану рекурсією які нереалізують Тюрінг повноту, через обмеженість ресурсів, також класифікують як вимотуюче мислення. Більшість помилок в індустрії, а також в теорії паралельних обчислень на наявність неконтрольованих циклів, які в граничних помилкових ситуаціях та споживають значну кількість ресурсів. 7-й вид мислення моделюється коіндукивними процесами, паралельними процесами, тощо.

6-й вид мислення або sems sde, звичайний гільбертовий обчислювач, або топос Гротендіка. CoC, MLTT — теж відносяться до 6-го типу мисленння. Чисті формальні теорії без парадоксів. Хоча декартово-замкнені категорії та симетричні моноїдальні категорії представлені як чотири ступені 8-го виду мислення, можна згадати їх тут, так як вони є моделями самоусвідомлення базового мислення. Яке з приходом наступного рівня зустрічається за парадоксами, які можуть стати на заваді до 8-го виду мислення. 6-й вид мислення — це фібраційна ясність просвітлення.

5 когнітивних мислень які маніфестуються в Йогачарі як (оціночні) дуальності та мандала п'яти сімейств. Активно проаналізовані можливості текстуальної генерації, відео та аудіо генерації та впроваджені вигляді публічних натренованих когнітивних сервісів. В основному використовується лінійна алгебра, теорія груп, топологічний аналіз даних, теорія нейромереж.

Кон'юнктура

Немає математики за межами цієї вежі, немає мислення за межами цієї вежі, немає просторів за межами цієї вежі. Можливо, навіть, це спряжена трійка:

$$\text{Простір} \dashv \text{Мислення} \dashv \text{Математика}$$

16 Два види мислення

Dissecting Details

The Dissecting Details approach, exemplified by Jean Leray and Jean Dieudonné, breaks complex problems into fundamental components, ensuring every step is rigorously verified. Leray's spectral sequences and Dieudonné's formalizations in Bourbaki's *Éléments de Mathématique* provided structured methods to predict outcomes, such as homology groups or algebraic properties, laying critical foundations for later work. However, their meticulousness often resulted in results so intricate that they are rarely applied or shared effectively. The predicative power of Dissecting Details lies in its ability to ensure reliable outcomes through systematic, rigorous methods. For instance, Leray's spectral sequences predict homology groups by organizing computations into a structured grid, while Dieudonné's formal algebra provides a foundation for predicting structural properties. Yet, the complexity of these methods can make predictions difficult to verify or extend, limiting their practical impact. Dissecting Details struggles with knowledge transfer due to the dense, technical nature of its results. While theoretically sound, the resulting proofs are often inaccessible, like intricate mosaics that are correct but hard to convey. Below are examples of significant theorems broken down with such rigor that their complexity hinders practical use and dissemination: 1. Leray's Early Spectral Sequences (1940s): Leray's spectral sequences for fiber bundles enabled precise homology computations but required tracking differentials across multiple complex stages. Their intricacy made them difficult to teach or apply, and simpler alternatives, like the Serre spectral sequence, became preferred for their accessibility. 2. Dieudonné's Lie Algebra Formalization (1950s): Dieudonné's exhaustive classification of Lie algebras in Bourbaki's treatise was a rigorous milestone, but its dense notation and case-by-case analysis limited its adoption. Modern treatments using root systems are more teachable, relegating Dieudonné's work to a theoretical reference. 3. Weyl's Original Character Formula Proof (1920s): Hermann Weyl's proof of the character formula for semisimple Lie algebras involved meticulous computations of weights and roots. Its complexity made it challenging to verify or share, and later geometric proofs became standard for their clarity.

These examples highlight how Dissecting Details, while powerful in breaking down complex problems, often fails to produce transferable knowledge. Students risk producing work that, though correct, remains isolated due to its inaccessibility, underscoring the need for broader perspectives.

Trivializing Complexity

Building on the rigorous foundations of his predecessors, Alexander Grothendieck revolutionized mathematics by creating frameworks that simplify profound problems, akin to water seamlessly filling gaps. His schemes in algebraic geometry and toposes in category theory reframed challenges like the Weil Conjectures, making solutions predictable within a unified system. The predicative power of Trivializing Complexity lies in its ability to anticipate results through abstraction. Schemes enable predictions about geometric properties by embedding them in a universal algebraic context, as seen in étale cohomology's foresight of connections between geometry and topology. This approach allows mathematicians to hypothesize outcomes for problems like the Riemann Hypothesis by leveraging coherent, general structures. Trivializing Complexity excels in transferring knowledge to other minds. By filling conceptual gaps with intuitive, general frameworks, Grothendieck's theories—such as schemes—are widely taught and adapted across mathematical domains. For example, the concept of a scheme is a cornerstone of algebraic geometry, enabling students and researchers to grasp and extend complex ideas. This transferability stems from the approach's ability to simplify without sacrificing depth, making it communicable and versatile. However, grand visions require technical grounding to be effective. Without the rigorous details provided by Dissecting Details, overambitious frameworks risk becoming speculative, failing to deliver concrete predictions or communicable insights. Students chasing monumental problems must balance ambition with precision to ensure their ideas are transferable.

Conclusion

Historically, Dissecting Details laid the groundwork for Trivializing Complexity. Dieudonné's rigorous algebra enabled Grothendieck's schemes, and Leray's technical tools supported broader topological insights. Combining meticulous rigor with visionary abstraction maximizes predicative power and knowledge transfer, ensuring theories are both predictive and teachable. Students eager to tackle grand challenges, like the Riemann Hypothesis, must heed Grothendieck's allegory: the "water" of unifying ideas needs a container of technical precision. Excessive Dissecting Details risks producing isolated, overly complex results, as seen in the examples above, while ungrounded Trivializing Complexity yields speculative theories. A balanced approach empowers you to predict outcomes and share them effectively with the mathematical community.

17 Мадг'яміка в MLTT баченні

У тибетській традиції є таке поняття — Дхармадхату — простор всіх Дхарм. Це простір всіх думок, які існують поза контекстами. Уявити складно – ясна річ. Але у нас є MLTT теорія, яка допоможе це уявити.

Уявіть собі простір всіх мов програмування — ясна річ всі знають про лямбда числення. Але не багато ж рубістів або ПХП програмісти замислюються, що вони пишуть мовою простору — який з точністю до бітів моделюється теорією типів Мартіна Льофа (треба тільки всесвіт правильно відконфігурувати, це навчилися робити в 2001 році, коли Coq все дружно писали). Звичайно я сам у це спочатку не вірив, і думав, що є все-таки обмеження і що Данило Майстер, який вручну все сам робить замість того, щоб екстрактити це з Інфініті Топоса робить недаремну працю. Мені довелося написати прувер щоб зрозуміти — таки так, кожен Данило Майстер, який вважає себе інженером, робить тимчасову і марну працю, даючи ярлики феноменам не бачачи їхньої суті.

Простір типів безмежний і всі типи одночасно живуть у цьому нескінченномірному просторі — де кожен його вимір нашаровується один на одного, а самі типи утворюють патерни, схожі на гомотопічні групи. Оскільки ізоморфних типів у просторі нескінченного топосу вистачає, то MLTT теж є щось схоже на матрицю гомотопічних груп. Вони мають різні імена і можуть у принципі не проходити карбування на рівність тощо, але при компіляції в нетимізовану лямбду ізоморфні типи генеруватимуть сумісний код.

Всі типи мають чітку логіку заселення простору нескінченного топосу, подібно до того, як жителі самсари заселяють шість лок. Починається все з нижнього дна пекла — типу Bottom. І потім по цеглині, починаючи з Unit(), потім A -> A, потім Nat, потім Stream, потім List, потім. і так далі аж до інфініті-групоїда, потім все починає повторюватися і візерунок починає змінюватися. Де у цього візерунка дірки я поки що не бачу, і яка у нього структура, але ніби відчуваю трохи подих цієї мандали. Ця мандала доступна в принципі всім програмістам, які можуть писати цикли, складати числа, більше рівня не потрібно. Всю цю математику можна переформулювати так, щоб MLTT викладати 11-річним дітям — вважайте, що експеримент розпочато.

KLONG тибетською - це простір, простір всіх феноменів, KLONG CHEN NYING THIG широке простір серцевої сутності. Так само як Інфініті Топос - це простір всіх MLTT типів, який поєднує континуум і дискретний, а також ковтає всю математику з усіма її логіками і теоріями, тому що всі математики розмовляють вже давно цією мовою кванторів і нескінченних всесвітів.

Так ось, так само, як усі мови народжені з цього простору, так само і думки всі народжуються з одного простору, і всі вони пройменовані, так само, як пройменовані всі типи у всіх всесвітах. Народжуються тут умовно, тому що List не народжується, його можна виявити зрозуміти, але він завжди присутній у Топосі у всіх рівнях всесвіту. Усі типи є нествореними, доки їх не оголосить програміст як аксіом MLTT. Але якщо їх розглядати через призму нечистого бачення — нетипізованого лямбду обчислення, то розглянути їх не вийде або дуже складно (Алонсо Черч та Боєм із Берардуччі не змогли цього зробити). Ну а аналогія просвітління - це вихід у цю мандалу, де видно всі типи відразу, або куди не кинеш погляд, бачиш скрізь схожі патерни. Послухай функціональних програмістів — натуральні ж шизофреніки, бачать якісь катаморфізми там, де звичайна людина бачить while чи for; бачать групи, де звичайна людина бачить record; чому звичайні інженери дали вже тисячу ярликів. Знаєте, є таке в духовних практиках бути остороненим і не вішати ярлики на феномени, бо це безглуздо. Це теж саме що List/Cons і Stream/Mk — мільйони програмістів дали різні назви цим штукам, але ці штуки існують самі по собі і побачити їх можна при індуктивному розгляді заповнення простору типами починаючи з Unit, і зробити це можна вже з третього кроку.

Взагалі аналогії настільки міцні, що я можу в священних текстах Тибету замінювати Дхармадхату на "Інфініті Топос SEMS або Розум на "правила висновку"(до речі один з перекладів SEMS - це обчислювач) або на "компютешинал аксіоми CHOS або DHARMA (ФО). Процес мислення — це те, як ви конструюєте теореми, як ви вибудовуєте рекурсивні ланцюжки, тобто. як ви композитуєте думки (рекурсія та індукція). Якщо у типу немає рекурсора, ви не можете пустити думку з цього феномену. А пустити думку щодо феномену означає його усвідомлення тобто. звільнення в дхармакаї, що впринципі відповідає етимології слова елімінатор.

Пора покласти край суперечкам різних шкіл Мадхьяміки (моделі сватантрик) та описати всі ці логіки у вигляді MLTT програм.