

- Инфо
- Вычислимость
 - 1. Машина Шёнфилда
 - 2. Функции, вычислимые на МШ
 - 3. Операторы S, R, M; Простейшие функции
 - 4. ПРФ и ЧВФ
 - 5. Вычислимые функции и множества
 - 6. ВПМ: эквивалентные определения
 - 7. Теорема Поста
 - 8. Замкнутость ВПМ и ВМ относительно операций
 - 9. Примеры ПРФ и ПРМ
 - 10. Эквивалентность ЧВФ и вычислимых на Шёнфилде функций
 - От МШ к функциям
 - От функций к МШ
 - Коды
 - Совместная рекурсия
 - Регистры и счётчик
 - Ещё раз теореме об эквивалентности ЧВФ и вычислимых на Шёнфилде функций
 - 11. Универсальные функции для семейств
 - 12. Существование и несуществование универсальных функций
 - 13. Канторовская нумерующая функция
 - 14. Теорема об униформизации
 - 15. Теорема о редукции
 - 16. Теорема о графике
 - 17. Существование вычислимо неотделимой пары ВПМ
 - 18. Теорема Майхилла об изоморфизме
 - 19. Эквивалентные определения цилиндра
 - 20. Определение простого множества
 - 21. Определение продуктивного множества
 - 22. Полнота множеств и творческие множества
- Лямбды
 - 1. Лямбда-термы
 - 2. Альфа- и бета-конверсии
 - 3. Теорема Чёрча-Россера
 - 4. Лямбда-исчисление
 - 5. Определение нормализуемого и сильно нормализуемого лямбда-термов
 - 6. Типизация лямбда-терма
 - 7. Алгоритм унификации Дж. РОбинсона и Хиндли
 - 8. Модель множеств, экстенциональная структура представлений
 - 9. Бета-модель, не экстенциональность

- 10. Модель бета-эта-термов
- 11. Модели Хенкина. Корректность и полнота
- 12. Основная теорема об ЛП
- 13. Омега-домены и домены
- 14. Непрерывность и омега-непрерывность

Инфо

Тут их называют "конвертами"... Ну что за безвкусица?.. Ладно, не суть. В общем, сюда вставляю выжимки из [основного файла](#)

Вычислимость

1. Машина Шёнфилда

Состоит из двух команд:

- INC I - увеличивает содержимое регистра I и счётчик команд на единицу
- DEC I, n - если содержимое регистр I больше нуля, уменьшает его на единицу и ставит счётчик команд на позицию n, иначе ничего не делает с I-м регистром и инкрементирует счётчик команд

Машина Шёнфилда задаётся:

- Потенциально бесконечным множеством регистров, занумерованных натуральными числами и содержащих натуральные числа. Любая фиксированная машина Шёнфилда работает с конечным числом регистров
- Счётчик команд - особая ячейка памяти, значение которой в начальный момент времени равно нулю
- Программа - конечное множество команд, пронумерованных натуральными числами.
Шаг машины - выполнение команды в ячейке, равной счётчику команд. Если команды с таким номером нет, программа остановится

2. Функции, вычислимые на МШ

Частичная числовая функция от n аргументов будет вычислимой на машине Шёнфилда, если в ячейках $i: 1 \leq i \leq n$ хранятся аргументы (**в остальных - нули**), а после выполнения программы P , машина завершится и в нулевой ячейке будет значение функции. Если функция не определена, машина должна работать бесконечно

Функции в машине Шёнфилда удобно использовать через макросы.

3. Операторы S, R, M; Простейшие функции

Простейшие функции:

- $\theta(x) = 0$
- $SUCC(x) = x + 1$
- $I_m^n(x_1, \dots, x_n) = x_m - n$, m - натуральные (целые и ≥ 1), $m \leq n$ - функция проекции

Оператор суперпозиции (S). Пусть есть частичные функции такие, $h(y_1, \dots, y_M)$, $g_1(x_1, \dots, x_N)$, ..., $g_M(x_1, \dots, x_N)$ и результатом применения оператора суперпозиции к этим функциям назовём функцию $f(x_1, \dots, x_N) = S(h, g_1, \dots, g_M)$ такую, что $f(x_1, \dots, x_N) = h(g_1(x_1, \dots, x_N), \dots, g_M(x_1, \dots, x_N))$

Оператор примитивной рекурсии (R). Пусть есть частичные функции $h^n(\vec{x})$ и $g^{n+2}(\vec{x}, y, z)$, то результатом применения оператора примитивной рекурсии $R(h, g)$ к этим функциям назовём функции f^{n+1} :

$$\begin{cases} f(\vec{x}, 0) = h(\vec{x}) \\ f(\vec{x}, y + 1) = g(\vec{x}, y, f(\vec{x}, y)) \end{cases}$$

Примитивная рекурсия для числа $a \in \omega$ и $h(y, z)$ будет определяться проще:

$$\begin{cases} f(0) = a \\ f(y + 1) = h(y, f(y)) \end{cases}$$

Оператор минимизации (M) Принимает функцию $g(\vec{x}, z)$, и возвращает $f(\vec{x})$ такую, что $f(\vec{x}) = y \Leftrightarrow \forall i < y (g(\vec{x}, i) \text{ определена и не равна нулю})$ и $g(\vec{x}, y) = 0$. В противном случае f не определена

$f(\vec{x}) = \mu y (g(\vec{x}, y) = 0)$ - так полностью записывается оператор минимизации

Функция $f(\vec{x})$ получается из всюду определённых функций $g(\vec{x}, y)$ и $h(\vec{x})$ с помощью ограниченного μ -оператора, если $g(\vec{x}, y) = 0$ и значение $y \leq h(\vec{x})$ и $\forall i < y : g(\vec{x}, i) \neq 0$, тогда $f(\vec{x}) = y$

Если $g(\vec{x}, y) \neq 0$ для всех $y \leq h(\vec{x})$, то $f(\vec{x}) = h(\vec{x}) + 1$

Записывается ограниченное μ так:

$$\mu y \leq h(\vec{x}) (g(\vec{x}, y) = 0)$$

Говоря по-человечески, мы ищем значение не ограниченно далеко вверх, а до какого-то значения, определяемого функцией $h(\vec{x})$. Если условие $g(\vec{x}, y) = 0$ при таких значениях не было удовлетворено, то мы результатом оператора будем $h(\vec{x}) + 1$

Если g, h - ПРФ, то ограниченная минимизация тоже вернёт ПРФ. Доказывается через тот факт, что результат ограниченной минимизации представим через:

$$f(\vec{x}) = \sum_{i=0}^{h(\vec{x})} sg(\prod_{j=0}^i g(\vec{x}, j))$$

4. ПРФ и ЧВФ

Частичная функция f называется **примитивно рекурсивной**, если существует последовательность таких функций $f_1, \dots, f_n = f$, что каждая функция либо простейшая, либо вычислима из предыдущих операторами суперпозиции и рекурсии. Говоря иначе, класс ПРФ является замыканием класса простейших функций относительно операторов S, R . Любая ПРФ всюду определена.

Частичная функция f называется **частично вычислимой**, если существует последовательность таких функций $f_1, \dots, f_n = f$, что каждая функция либо простейшая, либо вычислима из предыдущих операторами суперпозиции, рекурсии и минимизации. Говоря иначе, класс ПРФ является замыканием класса простейших функций относительно операторов S, R, μ . ЧВФ называется просто **вычислимой**, если она всюду определена

$$\text{ПРФ} \subset \text{ВФ} \subset \text{ЧВФ}$$

5. Вычислимые функции и множества

ВФ - всюду определённая ЧВФ (см. прошлый пункт), а тут речь пойдёт о множествах (их также справедливо называть отношениями)

Отношение $R \in \omega^n$ называется **вычислимым (примитивно рекурсивным)**, если его характеристическая функция, равняется нулю для любой точки в множестве R и единице для любой точки вне него, примитивно рекурсивна.

6. ВПМ: эквивалентные определения

Множество ВПМ, если оно в точности совпадает с ООФ некоторой ЧВФ

Эквивалентные утверждения:

1. $A = \delta\phi, \phi$ - ЧВФ
2. χ_A^* - ЧВФ
3. $A = \rho\phi, \phi$ - ЧВФ
4. $A = \emptyset$ или $A = \rho f, f$ - ВФ
5. A конечно или $A = \rho f, f$ - инъективная ВФ
6. $A = \exists y Q(x, y), Q$ - вычислимый предикат
7. Существует такая сильно вычислимая последовательность, что $\emptyset = A_0 \subseteq A_1 \subseteq A_2 \subseteq \dots \subseteq A_s \subseteq A_{s+1} \subseteq \cup_s A_s = A$

8. Существует сильно вычислимая последовательность из пункта выше и дополнительно с условием $\forall s \in \omega : |A_{s+1} - A_s| \leq 1$

Док-ва (надеюсь, для критов они не нужны, но пусть будут):

- 1 \Rightarrow 2 - такая ЧХФ - это буквально та самая ЧВФ из первого условия, на которую навесили 0()
- 2 \Rightarrow 1 - очевидно, δ такой ЧХФ даёт нам A
- 2 \Rightarrow 3 - из такой ЧХФ легко сделать функцию $\psi(x) = x * s(\chi_A^*(x)) \Rightarrow \rho\psi = \delta\chi_A^* = A$
- 3 \Rightarrow 4 - по теореме Клини, для $\psi(x)$ из прошлого пункта найдутся ПРФ U и $T(x, y)$, отсюда определим:

$$f_1(x, s) = \begin{cases} U(\mu y \leq s.T(x, y)), \exists y \leq s.T(x, y) \\ x_0, \forall y \leq s!T(x, y) \end{cases}$$

($x_0 \in A$), отсюда определим $f(x) = f_1(l(x), r(x)) \Rightarrow A = \rho f$

- 5 \Rightarrow 4 - если A бесконечно, то найдётся какая-нибудь ВФ с $\rho f = A$, если A конечно, то через сумму произведений элементов на сигнумы номеров легко определить функцию с $A = \rho f$
- 4 \Rightarrow 5 - если A бесконечно и f - ВФ с $A = \rho f$, то определив вычислимую $g(x)$ выдающую наименьшие f -номера элементов, мы получим $f_0(x) = f(g(x))$, которая будет вычислима, инъективна и $A = \rho f_0$
- 4 \Rightarrow 8:
 - Если $A = \emptyset$, то сильно вычислимая последовательность определяется тривиально с ХФМ тождественно единицей и $|A_n| = 0$
 - Если есть некая ВФ f с $A = \rho f$, то отношение принадлежности легко определяется через $\exists i < n : m = f(i)$, а $\max(A_n \cup \{0\}) = g(n)$, где $g(n)$ рекурсивно ищет наибольший элемент из всех множеств (хотя выглядит вообще так, будто тут рекурсия не очень нужна)
 - Если $A_0 = \emptyset$, то $\exists i < 0 : m = f(i)$ (что это за херня?! Как у нас может быть число меньше нуля?..)
 - $A_s \subseteq A_{s+1}$ - очевидно, что $\forall x \in A_s : x = f(i_j)$ - и те же самые индексы будут для всех тех же элементов и в A_{s+1}
 - $A_s \subseteq A$ - аналогично
 - $(A \subseteq \cup_s A_s)$ - тут я не понял, что за дичь написана, поэтому также предлагаю сказать "в силу общих индексов в $f(i)$ "
- 8 \Rightarrow 7 - очевидно (забавно, что тут это реально очевидно)
- 7 \Rightarrow 6 - $Q(x, y)$ - тот самый предикат для пар из определения сильно вычислимой последовательности
- 6 \Rightarrow 1 - навешиваем на Q минимизацию и получаем, что δ это ЧВФ $= A$

Унарные ВПМ и k -местные преобразуются одни в другие по леммам о взаимном преобразовании унарных и k -местных функций за счёт канторовской нумерации

7. Теорема Поста

Пусть $A \subseteq \omega$. Тогда A вычислимо $\Leftrightarrow A$ и $\bar{A} = \omega \setminus A$ - ВПМ

Доказательство:

\Rightarrow следует из того, что вычислимые множества также являются и ВПМ, а также ВМ замкнуты относительно операции дополнения

\Leftarrow предположим, что A и \bar{A} - ВПМ, тогда будут предикаты $A = \exists y Q_0(x, y)$ и $\bar{A} = \exists y Q_1(x, y)$, где Q_0, Q_1 - вычислимые. Далее возьмём функцию $f(x)$, которая будет подбирать такой y , чтобы он удовлетворял хотя бы одному из предикатов: $f(x) = \mu y (Q_0(x, y) \vee Q_1(x, y))$ - всюду определённая ЧВФ (т.к. $A \cup \bar{A} = \omega$), тогда $x \in A \Leftrightarrow Q_0(x, f(x)) \Rightarrow \chi_A(x) = \chi_{Q_0}(x, f(x)) \Rightarrow A$ вычислимо

8. Замкнутость ВПМ и ВМ относительно операций

Для двух таких отношений определяются операции объединения, пересечения, дополнения и импликации, причём если оба отношения ПР, то и отношение, полученное как результат операций, также будет ПР. *Доказывается через очень тривиально определяемые характеристические функции новых множеств (произведение хар. функций исходных множеств для объединения и т.п.)*

Также достаточно тривиально определяется декартово произведение двух отношений. И также отношение, образованное этой операцией, будет ПР, если ПР его аргументы.

Отсюда делаем вывод, в частности, для бинарных отношений $=, \neq, <, >, \leq, \geq$. В частности, для отношения равенства характеристической функцией будет $sg(|x - y|)$

Суперпозиция для отношений определяется тривиально как суперпозиция для характеристической функции

Оператор минимизации вводится для отношений через условие равенства характеристической функции нулю (ака принадлежности отношению), причём результат минимизации для ПР отношения будет ЧВФ, а для ПР отношения и ограниченной минимизации - ПРФ.

Операции над ВПМ

- Пересечение и декартово произведение ВПМ - тоже ВПМ (общая δ - это ООФ композиции их ЧВФ)
- Объединение ВПМ - тоже ВПМ (здесь получим общий предикат Q за счёт \vee между изначальными)

- Проекция ВПМ с минус одной размерностью также будет ВПМ. Доказывается за счёт уменьшения размерности предиката Q при помощи канторовской k-местной нумерации
- Все предикаты $\exists <, \exists \leq, \forall <, \forall \leq$ - также ВПМ
- Суперпозиция над элементами ВПМ даст также ВПМ
- Отображение ВПМ при помощи ЧВФ даст ВПМ

9. Примеры ПРФ и ПРМ

Примитивно рекурсивными будут функции:

- $x + y$
- $x * y$
- x^y ($0^0 = 1$)
- $sg(x)$
- $!sg(x)$
- $x \dot{-} y$
- $|x - y|$

Т. Если функция $g(\vec{x}, y)$ - ЧВФ (ПРФ), то и функции

$$f(\vec{x}, y) = \sum_{i=0}^y g(\vec{x}, i)$$

$$h(\vec{x}, y) = \prod_{i=0}^y g(\vec{x}, i)$$

также будут ЧВФ (ПРФ). Доказывается через тот факт, что и сумма и произведение представимы через оператор примитивной рекурсии

Используя бинарные отношения (они все ПРМ) $=, \neq, <, >, \leq, \geq$, может ещё ряд операций определить:

$x \text{ div } y$ - ЧВФ. Деление на ноль не опр.

Если мы используем оператор огр. минимизации, то $x \text{ div } y$ станет ПРФ, но при этом $x \text{ div } 0 = x$

ПРФ:

1. $x \bmod y: f(x, y) = x - (x \text{ div } y) * y$
2. $y|x = \exists z \leq y (z * x = y)$
3. $\text{prime}(x)$ - x - простое число - $((x > 1) \wedge \forall y \leq x (y|x \rightarrow (y = x) \vee (y = 1)))$
4. $p(x)$ - x -е простое число
5. $\text{ex}(i, x)$ - показатель i -го простого числа в разложении числа x

Шатал я переписывать эти формулы, так что для них будут просто скрины:

Доказательство.

1) $\left[\begin{array}{l} p(0) = 2, \\ p(x+1) = \mu y \leq s(p(x)!).(Prime(y) \wedge (y > p(x))); \end{array} \right.$
 в этом случае $f = R(a, g)$, где $a = 2$ и
 $g(x, z) = \mu y \leq s(z!)(Prime(y) \wedge (y > z))$ (из доказательства
 теоремы Евклида о бесконечности простых чисел вытекает, что
 $p_{x+1} \leq p_0 \cdot p_1 \cdot \dots \cdot p_x + 1 \leq (p_x)! + 1$).
 2) $ex(i, x) = \mu y \leq x.(\neg Div(p_i^{y+1}, x) \vee (x = 0))$ (действительно,
 $y < p_i^y \leq x$). □

10. Эквивалентность ЧВФ и вычислимых на Шёнфилде функций

От МШ к функциям

Т. Любая ЧВФ вычислима на некоторой машине Шёнфилда. Доказывается через тот факт, что все примитивные функции и операторы S, R, M вычислимы на машине Шёнфилда:

Доказательство.

Простейшие функции

$0(x)$ 0 : ZERO 0
 $s(x)$ 0 : INC 1; 1 : [1] \rightarrow [0]
 I_m^n 0 : [m] \rightarrow [0]
Оператор S ($f^m = S(g^n, h_1^m, h_2^m, \dots, h_n^m)$)
 0 : $h_1([1], [2], \dots, [m]) \rightarrow [m+1]$
 1 : $h_2([1], [2], \dots, [m]) \rightarrow [m+2]$
 ...
 $n-1$: $h_n([1], [2], \dots, [m]) \rightarrow [m+n]$
 n : $g([m+1], [m+2], \dots, [m+n]) \rightarrow [0]$

Доказательство (окончание).

Оператор R ($f^{n+1} = R(g^n, h^{n+2})$)
 0 : $g([1], [2], \dots, [n]) \rightarrow [0]$
 1 : $[n+1] \rightarrow [n+2]$
 2 : ZERO $n+1$
 3 : INC 0
 4 : DEC 0, 7
 5 : $h([1], [2], \dots, [n], [n+1], [0]) \rightarrow [0]$
 6 : INC $n+1$
 7 : DEC $n+2, 5$
Оператор M ($f^n = M(g^{n+1})$)
 0 : INC 0
 1 : DEC 0, 3
 2 : INC 0
 3 : $g([1], [2], \dots, [n], [0]) \rightarrow [n+1]$
 4 : DEC $n+1, 2$ □

От функций к МШ

Вообще, есть вторая запись об этой теме... Да и пусть она тут тоже будет в целом, тут мы будем не определять функции на МШ, а описывать МШ при помощи функций

Коды

$code(< x_0, \dots, x_{k-1} >)$ - код последовательности, определяется как $code(< x_0, \dots, x_{k-1} >) = p_0^{x_0+1} * \dots * p_{k-1}^{x_{k-1}+1}$ и $code(< >) = 1$

- В таком случае, если x - код, то $lh(x) = \mu i \leq x.(ex(i, x) = 0)$ - длина последовательности

- $(x)_i = ex(i, x) - 1$ - i -я координата кода
- Множество Seq всех кодов является примитивно рекурсивным

Теперь можем определить коды команд:

- $cd(INC\ i) = code(<0, i>)$
- $cd(DEC\ i, n) = code(<1, i, n>)$
- Множество Com всех кодов команд примитивно рекурсивно (определяется тривиально через функции для кодов)

Теперь мы можем закодировать программу из P_0, \dots, P_{k-1} команд как $code(P) = code(<cd(P_0), \dots, cd(P_{k-1})>)$

- Множество Prog всех кодов программ примитивно рекурсивно (определяется через принадлежность координаты множеству Com)

Совместная рекурсия

$$\begin{aligned} f_0(\vec{x}, 0) &= g_0(\vec{x}) \\ f_1(\vec{x}, 0) &= g_1(\vec{x}) \\ f_0(\vec{x}, y + 1) &= h_0(\vec{x}, y, f_0(\vec{x}, y), f_1(\vec{x}, y)) \\ f_1(\vec{x}, y + 1) &= h_1(\vec{x}, y, f_0(\vec{x}, y), f_1(\vec{x}, y)) \end{aligned}$$

Если g_0, g_1, h_0, h_1 - ПРФ, то f_0, f_1 - ПРФ.

Доказывается через тот факт, что мы можем определить функцию $F(\vec{x}, y) = code(<f_0(\vec{x}, y), f_1(\vec{x}, y)>)$, которая будет кодироваться как бы по дереву (ну не хочу я эту безумную формулу выписывать)

Регистры и счётчик

Определим 2 функции, описывающими всё состояние определённой машины Шёнфилда:

- $st(e, x, n)$ - возвращает значение счётчика команд после выполнения n команд программы с кодом e и начальными данными с кодом x (то есть $x = code(<x_0, \dots, x_{k-1}>) \Rightarrow 0, x_0, \dots, x_{k-1}, 0, \dots, 0, \dots$ - содержимое регистров в начальный момент времени)
- $rg(e, x, n)$ - аргументы по смыслу те же - возвращает код регистров $r = code(<r_0, \dots, r_{e+k-1}>)$
- Если функциям даны некорректные данные (коды не соответствуют ни одной программе (хотя не уверен, что это возможно)), то обе функции вернут ноль
- Обе функции - ПРФ

Последнее утверждение имеет просто монструозное доказательство, поэтому вот несколько скриншотов:

Воспользуемся схемой совместной рекурсии.

$$ct(e, x, 0) = 0.$$

Для задания $rg(e, x, 0)$ определим вспомогательную прф:

$$\alpha(i, x) = \begin{cases} ex(i \dot{-} 1, x), & \text{если } 1 \leq i \leq lh(x); \\ 1 & \text{в противном случае.} \end{cases}$$

Тогда имеем

$$rg(e, x, 0) = \begin{cases} \prod_{i=0}^{e+(lh(x) \dot{-} 1)} p_i^{\alpha(i, x)}, & \text{если } Prog(e) \wedge Seq(x); \\ 0, & \text{если } \neg Prog(e) \vee \neg Seq(x). \end{cases}$$

$$ct(e, x, n+1) =$$

$$\begin{cases} s(z), & \text{если } Prog(e) \wedge Seq(x) \wedge (z < lh(e)) \wedge (((e)_z)_0 = 0) \vee \\ & \vee (((e)_z)_0 = 1) \wedge ((v)_{((e)_z)_1} = 0)); \\ ((e)_z)_2, & \text{если } Prog(e) \wedge Seq(x) \wedge (z < lh(e)) \wedge \\ & \wedge (((e)_z)_0 = 1) \wedge ((v)_{((e)_z)_1} > 0)); \\ z, & \text{если } Prog(e) \wedge Seq(x) \wedge (z \geq lh(e)); \\ 0 & \text{в остальных случаях.} \end{cases}$$

Введём вспомогательную прф $\beta(i, x, y) = \left[\frac{x}{p_i^{ex(i, x)}} \right] \cdot p_i^{y+1}$. Данная

функция удовлетворяет следующему условию: если

$x = code(\langle x_0, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_k \rangle)$ и $i \leq k$, то

$\beta(i, x, y) = code(\langle x_0, \dots, x_{i-1}, y, x_{i+1}, \dots, x_k \rangle)$.

Формально (снова $z = ct(e, x, n)$, $v = rg(e, x, n)$):

$$rg(e, x, n+1) =$$

$$\begin{cases} \beta(((e)_z)_1, v, s((v)_{((e)_z)_1})), & \text{если } Prog(e) \wedge Seq(x) \wedge (z < lh(e)) \wedge \\ & \wedge (((e)_z)_0 = 0); \\ \beta(((e)_z)_1, v, (v)_{((e)_z)_1} \dot{-} 1), & \text{если } Prog(e) \wedge Seq(x) \wedge (z < lh(e)) \wedge \\ & \wedge (((e)_z)_0 = 1); \\ v, & \text{если } Prog(e) \wedge Seq(x) \wedge (z \geq lh(e)); \\ 0 & \text{в остальных случаях.} \end{cases} \quad \square$$

А теперь менее формально:

- И ту, и другую функцию можно выразить через оператор совместной рекурсии, теперь надо определить g и h

- st в начальный момент времени равно нулю, rg - код регистров вида $0, x_0, \dots, x_{k-1}, 0, \dots, 0, \dots$
- Далее st на n -м шаге определяется весьма тривиально через разбор всех возможных случаев с разными командами (мы можем получить номер любой команды и понять, что делать, за счёт операций над кодами, которые также ПРФ)
- rg на n -м шаге также может либо увеличить какой-то регистр на единицу, либо уменьшить, либо не менять ни одного регистра, Проверая коды, проводим нужные модификации
- Таким образом, функции st и rg определились чрез совместную рекурсию с использованием сугубо ПРФ - **доказано**

$stop(e, x, n)$ - предикат остановки программы с кодом e с входными данными с кодом x на шаге n . Также ПРФ

Теперь можем определить код вычисления как $y = code(\langle rg(e, x, 0), rg(e, x, 1), \dots, rg(e, x, n) \rangle)$ - отсюда мы можем получить результат вычисления, обратившись к нулевой координате последнего элемента - обозначим функцию взятия результата как $U(y)$. Если e, x не удовлетворяют $stop(e, x, n)$ ни для какого n , то код вычисления считаем неопределённым.

Предикат Клини $T(e, x_1, \dots, x_k, y)$ говорит о том, что программа e с начальным кодом $x = code(\langle x_1, \dots, x_k \rangle)$ придёт к коду программы y . ПРФ для $k \geq 1$

Ещё раз теореме об эквивалентности ЧВФ и вычислимых на Шёнфилде функций

Любая частичная функция, вычисляемая на Шёнфилде - ЧВФ

Возьмём произвольную функцию, вычисляемую на Шёнфилде и произвольные начальные данные n_1, \dots, n_k :

- Если $f(n_1, \dots, n_k)$ определена, то машина остановится. Возьмём минимальный код y_0 , на котором это произойдёт, тогда $f(n_1, \dots, n_k) = U(y_0) = U(\mu y. (T(e, n_1, \dots, n_k, y)))$ - а эта функция ЧВФ
- Если $f(n_1, \dots, n_k)$ не определена, то $T(e, n_1, \dots, n_k, y)$ всегда ложно, а значит и $U(\mu y. (T(e, n_1, \dots, n_k, y)))$ не определена

11. Универсальные функции для семейств

Для $k \geq 1$ местных частичных функций, собранных в семейство S $k + 1$ местная функция F называется **универсальной для семейства S** , если $S = \{\lambda x_1 x_2 \dots x_k. F(e, x_1, \dots, x_k) \mid e \in \omega\}$. Если S - семейство всех частичных функций, то F будет называться просто **универсальной**

Человеческая интерпретация: универсальной будет называться функция, которая может при подборе одного аргумента образовать любую функцию из множества

12. Существование и несуществование универсальных функций

Какого бы ни было $k \geq 1$, не существует универсальной ЧВФ (ПРФ) семейства всех k -местных ПРФ. Доказывается из того факта, что в противном случае у нас существовала бы универсальная функция, обращаемая бы одновременно в f и $s(f)$, что создаёт противоречие

Какого бы ни было $k \geq 1$, не существует универсальной ЧВФ (ПРФ) семейства всех k -местных ПРФ, принимающих значения 0 или 1. Доказывается из того факта, что в противном случае у нас существовала бы универсальная функция, обращаемая бы одновременно в f и $!sg(f)$, что создаёт противоречие

Т. Какого бы ни было $k \geq 1$, существует универсальная $k + 1$ местная ЧВФ для семейства всех k -местных ЧВФ. Доказывается через существование $U(\mu y.T_k(e_0, x_1, \dots, x_k, y))$, связанной с машинами Шёнфилда (либо просто говорим, что с точки зрения машин мы можем считать номером универсальной функции номер команды) (с теми же словами справедливо и для функций, принимающих значения только 0 и 1)

13. Канторовская нумерующая функция

Канторовская нумерация определяется ПРФ $C^2(x, y) = \frac{(x+y+1)(x+y)}{2} + x = \frac{(x+y)^2 + 3x+y}{2}$, отображает \mathbb{N}^2 в \mathbb{N} и делает это биективно

- $l(C^2(x, y)) = x$
- $r(C^2(x, y)) = y$
- $C^2(l(x), r(x)) = x$

Рядами в поле таких чисел будут нумероваться по побочным диагоналям, начиная обход с верхнего левого угла.

Количество рядов до точки $(x, y) = x + y$

Сумма точек до ряда с точкой $(x, y) = (x + y + 1)(x + y)/2$

Основной смысл канторовской нумерации в том, что мы можем осуществлять взаимно однозначное соответствие между \mathbb{N}^2 и \mathbb{N} , что позволяет оперировать нам с парами, как обычными числами. В более общем случае мы можем взаимно отобразить \mathbb{N}^n в \mathbb{N} , рекурсивно применяя канторовскую нумерацию к первым двум аргументам:

- $c^1(x) = x$
- $c^n(x_1, \dots, x_n) = c^2(c^{n-1}(x_1, \dots, x_{n-1}), x_n)$
- Тогда определена также ПРФ $c_{n,i}(c^n(x_1, \dots, x_n)) = x_i$

14. Теорема об униформизации

Пусть $R \subseteq \omega^{n+1}$ - ВПМ, тогда найдётся n -местная ЧВФ ψ , униформизирующая данный предикат, то есть выполняющая 2 условия:

- $\delta\psi = \exists y : R(x_1, \dots, x_n, y)$
- $\Gamma_\psi \subseteq R$ (говоря иначе, $\forall \langle x_1, \dots, x_n \rangle \in \delta\psi : R(x_1, \dots, x_n, \psi(x_1, \dots, x_n))$)

Попытка в человеческую интерпретацию: униформизирующая функция - это функция, которая позволяет понизить размерность множества на единицу в любой его точке, причём комбинация из её аргументов и значения будет принадлежать изначальному множеству

Доказательство:

- Для начала определим эту самую ψ . Если $R \subseteq \omega^{n+1}$ - ВП, то и $c^{n+1}(R) \subseteq \omega$ - ВП $\Rightarrow c^{n+1}(R) = \exists y : Q(x, y) \Rightarrow R = \exists y : Q(c^{n+1}(x_1, \dots, x_{n+1}), y)$, тогда определим $\psi(x_1, \dots, x_n) = l(\mu z. Q(c^{n+1}(x_1, \dots, l(z)), r(z)))$ - ЧВФ. Теперь докажем, что она удовлетворяет условиям теоремы
- Определённость $\psi(x_1, \dots, x_n)$ при $\exists y : R(x_1, \dots, x_n, y)$:
 - \Rightarrow если такая пси определена, то будет такое z_0 , которое даст нам $y = r(z_0)$, значит $Q(c^{n+1}(x_1, \dots, l(z_0)), r(z_0))$, значит $R(x_1, \dots, l(z_0))$, а значит $\exists y : R(x_1, \dots, x_n, y)$
 - \Leftarrow если $\exists y : R(x_1, \dots, x_n, y)$, то $Q(c^{n+1}(x_1, \dots, x), y)$, если возьмём а дальше $z = c(x, y)$, а затем подберём минимальное z_0 для $Q(c^{n+1}(x_1, \dots, l(z)), r(z))$ через минимизацию, то получим в точности ψ , которая, следовательно, будет определена
 - В сущности, выглядит так, будто в обе стороны доказывается фразой "вытекает из определения ψ , которое тесно связано со свойством $R = \exists y : Q$ "
- Ну а тут реально доказывается за счёт существования вот этого построения

15. Теорема о редукции

Какими бы ни были ВПМ A, B , найдутся такие ВПМ A_0, B_0 , что:

- $A_0 \subseteq A, B_0 \subseteq B$
- $A_0 \cup B_0 = A \cup B$
- $A_0 \cap B_0 = \emptyset$

По-человечески: любые 2 множества могут быть заменены меньшими или равными множествами так, чтобы новые множества покрывали всю ту же область и не пересекались

Доказательство: возьмём множество $R = \{A \times \{0\}\} \cup \{B \times \{1\}\}$ (объединим множества в, разделив по номерам источников элементов), а затем определим униформизирующую функцию ψ , которая по теореме об униформизации найдётся и при том её область определения будет в точности $\exists y : R(x, y) = A \cup B$, а область значений 0 и 1. Обозначим $A_0 = \psi^{-1}(0), B_0 = \psi^{-1}(1)$, тогда $A_0 \cap B_0 = \psi^{-1}(0) \cap \psi^{-1}(1) = \emptyset, A_0 \cup B_0 = \delta\psi = A \cup B$, ну а из факта наличия элемента в новых множествах следует **(и только в одну**

сторону), что он есть в $R(x, 0)$ либо в $R(x, 1)$ только в случае его наличия в исходных множествах, а значит $A_0 \subseteq A$ и $B_0 \subseteq B$

16. Теорема о графике

Пусть $\psi(x_1, \dots, x_k)$ - ЧФ. Она будет ЧВФ $\Leftrightarrow \Gamma_\psi$ - ВПМ

Доказательство:

- \Rightarrow Пусть $\psi(x_1, \dots, x_k)$ - ЧВФ, определим далее ЧВФ $\psi_1(x_1, \dots, x_k) = c^{k+1}(x_1, \dots, x_k, \psi(x_1, \dots, x_k)) \Rightarrow \rho\psi_1 = \rho\psi_2 = c^{k+1}(\Gamma_\psi)$, где $\psi_2(x) = c^{k+1}(c_{k,1}(x), \dots, c_{k,k}(x), \psi(c_{k,1}(x), \dots, c_{k,k}(x)))$, отсюда следует, что $c^{k+1}(\Gamma_\psi)$ - ВПМ - а значит и Γ_ψ - также ВПМ
- \Leftarrow Если у нас уже есть Γ_ψ - ВПМ, то по теореме об униформизации найдётся новая k -местный униформизатор, причём ЧВФ, значение которой в силу условия будет равняться y , а значит ψ будет равняться униформизатору, из чего следует, что она тоже ЧВФ

17. Существование вычислимо неотделимой пары ВПМ

Непересекающиеся ВПМ A, B называются **вычислимо отделимыми** в случае, когда существует ВМ C такое, что $A \subseteq C \subseteq \overline{B}$ (можно окружить одно из множеств так, чтобы граница не касалась другого множества). В противном случае будет называться **вычислимо неотделимым**.

Неотделимая пара ВПМ существует (например, множества значений универсальной функции для семейства унарных функций, принимающих значения 0 и 1 - в одно множество вносим все унификаторы для значения 0, в другое - для значения 1)

18. Теорема Майхилла об изоморфизме

Немного вводной части из нумераций:

Нумерация - любое сюръективное отображение натуральных чисел на непустое не более чем счётное множество S . Все нумерации множества обозначаются как $N(S)$

Нумерация ν_0 сводится к нумерации ν_1 , если существует f такая, что $\nu_0 = \nu_1 f$ (обозначается как $\nu_0 \leq \nu_1$)

Нумерация ν_0 **1-сводится** к нумерации ν_1 , если существует инъективная ВФ f такая, что $\forall n \in \omega : \nu_0(n) = \nu_1 f(n)$ (обозначается как $\nu_0 \leq_1 \nu_1$)

Нумерации ν_0 и ν_1 **вычислимо изоморфны**, если существует вычислимая перестановка p такая, что $\forall n \in \omega : \nu_0(n) = \nu_1 p(n)$ (обозначается как $\nu_0 \approx \nu_1$)

T. $\nu_0 \leq_1 \nu_1 \wedge \nu_1 \leq_1 \nu_0 \Rightarrow \nu_0 \approx \nu_1$ (собственно, почти Майхилл, но настоящий Майхилл для множеств)

Доказательство не буду тут приводить, ясное дело. Только совсем общую мысль - мы строим особую конструкцию множества из пар, где первые и вторые элементы уникальны по отдельности, за счёт инъективности сводящих функций делаем так, что она будет покрывать всю нумерацию, а далее просто заявляем, что это множество и представляем собой ту самую вычислимую перестановку

А теперь для множеств

$A, B \in \omega$

A **m-сводится** к B , если существует ВФ f такая, что $x \in A \Leftrightarrow f(x) \in B$. Обозначается $A \leq_m B$

Если $A \leq_m B$ и $B \leq_m A$, то $A \equiv_m B$ (**m-эквивалентны**)

1-сводятся - f вдобавок инъективна

Вычислимая изморфность определяется аналогично через вычислимую перестановку с учётом замены записи на принадлежность множествам

Ну и, собственно, Майхилл:

T. Если $A \leq_1 B$ и $B \leq_1 A$, то $A \approx B$

19. Эквивалентные определения цилиндра

Сначала о цилиндрах для нумераций

Цилиндром нумерации ν называется нумерация $c(\nu) : \omega \rightarrow S$ для которой $c(\nu)(c(x, y)) = y$ либо, что то же самое $c(\nu)(x) = \nu(r(x))$ (да-да! Так круто, что для обозначения изменения нумерации используется та же буква, что и для канторовской функции...)

Нумерация будет **цилиндрической**, если она вычислимо изоморфна своему цилиндру

Следующие утверждения эквивалентны:

- ν - цилиндрическая нумерация
- \exists ВФ такая, что $\forall x : (f(x) > x) \wedge (\nu(f(x)) = \nu x)$
- $\forall \nu' : (\nu' \leq \nu) \rightarrow (\nu' \leq_1 \nu)$

Доказательства:

- 1 \Rightarrow 2 - из цилиндричности нумерации следует существование перестановок p_1, p_2 таких, что $\nu = c(\nu)p_1$ и $c(\nu) = \nu p_2$, тогда определим $f(x) = p_2(c(z, rp_1(x)))$, где z

будет подбираться минимизацией так, чтобы значение $f(x)$ было больше x . В силу того факта, что это канторовская нумерация и перестановки, такое число будет найдено, а значит $f(x)$ вычислима, в силу определения, $\nu f = \nu p_2(c(z, rp_1(x))) = c(\nu)(c(z, rp_1(x))) = \nu rp_1(x) = c(\nu)p_1(x) = \nu x$

- 2 \Rightarrow 3 - определим сводящую функцию g и сплинттер F над функцией f , а дополнительно ещё и рекурсивную h , которая при нуле будет просто возвращать все сведённые элементы, применяя к ним сплинттер. В силу такого определения функций и полученное из пункта 2 функции f, h будет инъективна, из чего следует, что $\nu' \leq \nu \rightarrow \nu' \leq_1 \nu$
- 3 \Rightarrow 1 - просто берём ν' как $c(\nu)$, функцией для $c(\nu) \leq_1 \nu$ будет $r(x)$, а для $\nu \leq_1 c(\nu)$ - $\lambda x. c(0, x)$ - из чего по определению и следует цилиндрическая нумерация

С. Всякий цилиндр является цилиндрической нумерацией (по второму свойству из списка выше берём $f(x) = c(s(l(x)), r(x))$ - нетрудно заметить, что её значение будет всегда больше x , а $c(\nu)(x) = \nu r x = c(\nu)(f(x)) = c(\nu)(c(s(l(x)), r(x))) = c(\nu)(r(x)) = \nu r x$)

А теперь о цилиндрах для множеств:

Множество $A \subseteq \omega$ называется **цилиндром** если A и $c(\omega \times D)$ вычислимо изоморфны для некоторого $D \subseteq \omega$

A - **цилиндрификация** множества B , если $A = c(\omega \times B)$

Следующие утверждения эквивалентны:

- A - цилиндр
- \exists ВФ такая, что $\forall x : (f(x) > x) \wedge (x \in A \Leftrightarrow f(x) \in A)$
- $\forall B : (B \leq_m A) \rightarrow (B \leq_1 A)$

20. Определение простого множества

Множество A **иммунно**, если оно бесконечно и не содержит в качестве подмножества бесконечное ВПМ (Забавный логичный факт - если A иммунно, то оно не может быть ВПМ)

ВПМ с иммунным дополнением называются **простыми**

Свойства иммунности и простоты инвариантны (то есть не зависят от перестановок)

21. Определение продуктивного множества

Множество P **продуктивно**, если существует ЧВФ $\psi(x)$, называемая **продуктивной функцией** для P , что $\forall x : (\pi_x \subseteq P) \Rightarrow (\psi(x) \downarrow \wedge (\psi(x) \in P \setminus \pi_x))$ (то есть P включает себя и ОДЗ и ООФ некоторой функции для всех ОДЗ, входящих в него)

ВПМ C **творческое**, если \overline{C} продуктивно

Пример: \overline{K} продуктивно для $\psi(x) = x$

22. Полнота множеств и творческие множества

ВПМ M называется **m-полным**, если $\forall A$ ВПМ : $A \leq_m M$

1-полное множество определяется аналогично

Любые 2 1-полных множества вычислимо изоморфны

ВПМ C **творческое**, если \overline{C} продуктивно

Пример: множество $K = \pi_x$ творческое, поскольку \overline{K} продуктивно для $\psi(x) = x$

Т.

- Если P - продуктивно, то $\overline{K} \leq_1 P$
- Если C - творческое, то оно 1-полно и в частности, $C \approx \overline{K}$ (доказывается напрямую из первого утверждения и определения творческого множества)

эквивалентные утверждения для продуктивных множеств:

- P продуктивно
- $\overline{K} \leq_1 P$
- $\overline{K} \leq_m P$

эквивалентные утверждения для творческих множеств:

- C творческое
- C 1-полно
- C m-полно

Лямбды

1. Лямбда-термы

Лямбда-терм (выражение) можно определить индуктивно:

1. Переменная - лямбда-терм
2. Константа сигнатуры σ - лямбда-терм
3. Если A, B - лямбда-термы, то $(A \ B)$ - лямбда-терм
 - $(A \ B)$ - оператор аппликации
4. Если A - лямбда-терм и x - переменная, то $\lambda x.A$ - лямбда-терм
 - Операция называется абстракцией
5. Других лямбда-термов нет

2. Альфа- и бета-конверсии

Альфа-конверсия - преобразование $\lambda x.T \rightarrow \lambda y.[T]_y^x$. При этом y Должна быть свободна для x в T

- \rightarrow означает "за один шаг переходит в ..."
- \Rightarrow означает "за конечное число шагов переходит в ..." или просто "преобразуется в ..." - рефлексивное транзитивное замыкание оператора \rightarrow

Бета-конверсия - преобразование $(\lambda x.TR) \rightarrow [T]_R^x$, при условии, что R свободная для x в T

3. Теорема Чёрча-Россера

Если $T \Rightarrow R$ и $T \Rightarrow S$, то найдётся такой терм Q , что $R \Rightarrow Q$ и $S \Rightarrow Q$. Это свойство называется **конфлюэнтностью**

Доказательство:

- Вводится особый вид конверсии \rightarrow_1 , при которой бета-конверсия к каждому подтерму применяется не более одного раза. В силу определения для неё будут справедливы леммы:
 - Если $T \rightarrow_1 R$, то $T \Rightarrow R$
 - \Rightarrow - транзитивное замыкание \rightarrow_1 и альфа-конверсии
 - Если $\lambda x.T \rightarrow_1 R$, то R имеет вид $\lambda x.R_0$ для подходящего R_0
 - $(TR) \rightarrow_1 S$, то $S = (QV)$, где либо $T \rightarrow_1 Q$ и $R \rightarrow_1 V$, либо $T = \lambda x.Q$, тогда $S = [Q_1]_{R_1}^x$ где $T \rightarrow_1 Q_1$ и $R \rightarrow R_1$
 - Если $T \rightarrow_1 R$ и $S \rightarrow_1 Q$, то $[T]_S^x \rightarrow_1 [T]_Q^x$ (при соблюдении свободности, разумеется. Доказывается индукцией по построению)
 - \rightarrow_1 обладает свойством конфлюэнтности (Доказывается также индукцией с применением прошлых лемм)
- А теперь мы просто говорим, что бета-конверсия - транзитивное замыкание \rightarrow_1 , которое инфлюэнтно, а значит и \Rightarrow конфлюэнтно

4. Лямбда-исчисление

Лямбда-исчисление или **комбинаторная логика** - набор конверсий и равенств, соответствующих одному и тому же выражению:

- $t = t$
- Альфа-конверсия
- Бета-конверсия

- $$\frac{t = u}{u = t}$$

- $$\frac{t = u; u = r}{t = r}$$
- $$\frac{t = u}{tr = ur}$$
- $$\frac{t = u}{rt = ru}$$
- $$\frac{t = u}{\lambda x. t = \lambda x. u}$$

Формула $t = u$ выводима в комбинаторной логике \Leftrightarrow существует такой терм v , что $t \Rightarrow v \wedge u \Rightarrow v$

5. Определение нормализуемого и сильно нормализуемого лямбда-термов

Нормальная форма лямбда-терма - лямбда-терм, если к любому его подтерму или результату альфа-конверсии над ним не применима бета-конверсия.

Нормальзуемый терм - терм, приводящийся к нормальному за конечное число конверсий

Для термов $M_1 \Rightarrow_\beta^1 M_2$ означает, что существует такой терм с дырой T , что $M_1 = T[R_1]$ и $M_2 = T[R_2]$, где $R_1 \rightarrow_\beta R_2$. Если $M_1 \Rightarrow_\beta^1 M_2$, то $M_1 \Rightarrow M_2$

Обозначение $M_1 \Rightarrow_\beta^n M_2$ означает, что существуют такие $n + 1$ термов N_i , что $N_i \Rightarrow_\beta^1 N_{i+1}$, $M_1 = N_0$, $M_2 = N_n$

Бета-ранг терма M : будем считать, что $rk_\beta(M) \geq n$, если:

- $n = 0$: M - любой терм
- $0 < n < \omega$: $\exists M' : M \Rightarrow_\beta^n M'$
- $n = \omega$: существует (бесконечная счётная?) последовательность M_i : $M_i \Rightarrow_\beta^1 M_{i+1}$

Соответственно, равным n ранг будет, если $rk_\beta(M) \geq n \wedge rk_\beta(M) < n + 1$

Нормальный терм имеет нулевой ранг

Лямбда-терм **сильно нормализуем**, если его бета-ранг меньше ω

Любой сильно нормализуемый лямбда-терм нормализуем (доказывается от противного (а я бы и вовсе сказал, что в силу определения ранга мы можем заявить, что терм с рангом меньше бесконечности допускает лишь конечное число бета-конверсий, что и доказывает его нормализуемость))

6. Типизация лямбда-терма

Каждому лямбда-терму приписывается тип, это накладывает ограничения.

Предполагаем, что есть набор базовых типов

Можем строить составные типы по τ_0 и τ_1 , обозначая тип как $(\tau_0 \rightarrow \tau_1)$ - называется классом всех отображений из τ_0 в τ_1

Типом будет любой базовый тип, а также составной тип, состоящий из типов

$x : \tau$ - x имеет тип τ . Сопоставление типа переменной будет называться **типизирование**

Как задать тип терму:

1. Нужно задать типы всех переменных и констант
2. Если $M : \pi \rightarrow \tau$ и $N : \pi$, то $(M N) : \tau$
3. Если $M : \tau$ и $x : \pi$, то $\lambda x.M : \pi \rightarrow \tau$

Лямбда-терм типизируемый, если его переменным можно приписать типы так, чтобы сам терм тоже получил тип

Какова бы ни была типизация переменных, терм может получить не более одного типа (доказывается индукцией)

Тип терма не меняется от конверсий.

7. Алгоритм унификации Дж. РОбинсона и Хиндли

Немного базы про унификацию:

Типовые переменные - простейшие типы

Подстановка типов пишется как $\sigma[\alpha_1/\rho_1, \dots, \alpha_n/\rho_n]$, где α_i - различные типовые переменные, ρ_i - произвольные типы, σ - преобразуемый тип. $S = \sigma[\alpha_1/\rho_1, \dots, \alpha_n/\rho_n]$. Остаётся добавить лишь одно правило $(\alpha \rightarrow \beta)S = (\alpha S \rightarrow \beta S)$ (ну и, разумеется, не затронутые подстановкой типы остаются неизменными)

Равенство $\sigma = \tau$ имеет **унификатор**, если существует подстановка S такая, что $\sigma S \equiv \tau S$. S - собственно, **унификатор**

S - **наиболее общий унификатор (НОУ)**, если S унификатор для σ, τ и для любого другого их унификатора S' найдётся такой унификатор T , что $\sigma S' \equiv \sigma ST$ и $\tau S' \equiv \tau ST$

Если один тип содержит другой, то у них не будет унификатора

Робинсон:

Для типов строится дерево, листья которого помечаются (собственно, типами):

- Типовая переменная - лист

- Если тип $(\alpha \rightarrow \beta)$, то будет 2 поддерева по таким же правилам

Сам алгоритм:

- Строим деревья для обоих типов
- Начинаем параллельный левосторонний обход в глубину
- Если дошли для вершин, из которых хотя бы одна помечена, то смотрим, нет ли помеченной тем же типом вершины во втором поддереве (очевидно, всё второе поддерево также может быть вершиной):
 - Если есть, унификация невозможна
 - В ином случае добавляем к унификатору замену типа из помеченной вершины на тип второго поддерева
- Когда различий в деревьях не осталось, можно заявить, что унификатор построен

T. Алгоритм Робинсона позволяет определить, имеет ли выражение $\sigma = \tau$ унификатор. Если унификатор имеется, то алгоритм выдаст наиболее общий унификатор

Доказательство:

- \Rightarrow Алгоритм конечен в силу конечности типов, а значит если алгоритм дошёл до конца, равенство унифицируемо
- \Leftarrow Если равенство унифицируемо, то докажем индукцией, что на любом шаге у нас будет унификатор, который можно получить из НОУ
 - То есть у нас имеется некий промежуточный унификатор S' , а строить мы будем T_i
 - $T_0 = S'$
 - Если на $i + 1$ шаге возможно добавление подстановки к S_i , то та же подстановка будет выниматься из T_i , а значит, в силу базы индукции и того, что $S_i T_i = S'$, получаем $S_{i+1} T_{i+1}$
- В силу построения, заключаем, что финальное S_n будет НОУ

Хиндли:

Система равенств $\sigma_1 = \tau_1, \dots, \sigma_n = \tau_n$ **унифицируема**, если существует подстановка, унифицирующая все равенства в ней

Проблема унификации системы сводится к проблеме унификации $\sigma = (\dots(\sigma_1 \rightarrow \sigma_2) \Rightarrow \dots \Rightarrow \sigma_n)$ и $\tau = (\dots(\tau_1 \rightarrow \tau_2) \Rightarrow \dots \Rightarrow \tau_n)$

σ - **наиболее общий тип (НОТ)** терма M с типами переменных T , если $T \vdash M : \sigma$ и для любого другого типа σ' с типизацией переменных T' такого, что $T' \vdash M : \sigma'$ найдётся подстановка S такая, что $\sigma' = \sigma S$. Определяется НОТ по построению:

- Если $M \equiv x$ - терм-переменная, то $((x : \sigma) \vdash x) : \sigma$ (σ - НОТ)
- Если $M \equiv \lambda x. P$ и $T \vdash P : \tau$ - НОТ, то $((T \setminus \{x : \sigma\}) \vdash \lambda x. P) : (\sigma \rightarrow \tau)$ - НОТ (если в T нет типа для x , то, очевидно, вычитать его не надо)

- Если $M \equiv (PQ)$ и $T_1 \vdash P : \sigma_1, T_2 \vdash Q : \sigma_2$, то производим переименование типов так, чтобы в сигмах были разные типы, а затем ищем унификатор для системы:

$$\circ \quad \begin{cases} \sigma_1 = (\sigma_2 \rightarrow \tau) \\ \alpha_i = \beta_i \\ \dots \end{cases}$$

- $\alpha_i = \beta_i$ нужны для тех случаев, когда у нас общая переменная в P и Q (**не путать с общими типами, от которых мы избавились в начале**)
- Если система унифицируемая НОУ S , то $((T_1 S \cup T_2 S) \vdash M) : \tau S$

Это и есть алгоритм Хиндли

Т. Алгоритм Хиндли выдаёт наиболее общий тип, если терм типизируем (*доказывается очень похоже на алгоритм Робинсона*)

Вариант очень душный и подробный. Сам Пузырь на консультации говорил, что Хиндли - это по факту Робинсон для систем, а главное его отличие в действии при аппликации

8. Модель множеств, экстенциональная структура представлений

Σ - множество всех типов, построенных из атомарных типов. Лямбда-термы (в частности, переменные) рассматриваются как априори типизированные. $A \rightarrow B$ - множество всех функций, действующих из A в B

Для атомарных типов фиксируем множества D_σ :

- $D_{\tau \rightarrow \sigma} = D_\tau \rightarrow D_\sigma$
- $D_{\alpha \rightarrow (\beta \rightarrow \gamma)} = (D_\alpha \times D_\beta) \rightarrow D_\gamma$

Означивание переменной $x : \sigma$ - Отображение $\delta : x \rightarrow D_\sigma$

$$\delta[y^\tau \rightarrow a](x^\sigma) = \begin{cases} \delta(x^\sigma), x^\sigma \not\equiv y^\tau \\ a, x^\sigma \equiv y^\tau \end{cases}$$

Я бы назвал это операцию... *Присваиванием?*

При помощи означивания δ мы можем провести **интерпретацию** термина M , определяемую индуктивно:

- $[[x^\sigma]]_\delta = \delta(x) \in D_\sigma$
- Если $M : \tau \rightarrow \sigma$ и $N : \tau$, также уже заданы $[[M]]_\delta \in D_{\tau \rightarrow \sigma}$ и $[[N]]_\delta \in D_\tau$, то $[[MN]]_\delta = [[M]]_\delta([N])_\delta \in D_\sigma$
- Если $M : \sigma$ и уже задано $[[M]]_\delta \in D_\sigma$, то $[[\lambda x^\tau. M]]_\delta = ((a \in D_\tau) \rightarrow [[M]]_{\delta[x^\tau \rightarrow a]}) : D_\tau \rightarrow D_\sigma \in D_{\tau \rightarrow \sigma}$

Интерпретация зависит **только от типов свободных переменных**. Таким образом, если свободных переменных нет, можно писать $[[M]]$

- Сопоставим каждому типу σ множество A_σ
- Каждой паре типов σ, τ отображение $app_{\sigma, \tau} : (A_{\sigma \rightarrow \tau} \rightarrow (A_\sigma \rightarrow A_\tau))$

Пара $A = (A_\sigma, app_{\sigma, \tau})$ называется **структурой представлений**. Если отображение app к тому же инъективно для всех $\sigma, \tau \in \Sigma$, то это будет **экстенсинальная структура**, то есть 2 различных элемента $f, g \in A_{\sigma \rightarrow \tau}$ будут кодировать разные функции $A_\sigma \rightarrow A_\tau$

Моделью множеств будет структура, в которой $app_{\sigma, \tau}$ - тождественное отображение

9. Бета-модель, не экстенсинальность

Для терма M типа σ будем обозначать через $\langle M \rangle$ класс бета-эквивалентностей типизированных термов M' , который также содержит и сам M . Обозначим $T_\sigma = \{\langle M \rangle\}$

Если $\langle M \rangle \in T_{\tau \rightarrow \sigma}$ и $\langle N \rangle \in T_\tau$, то $\langle N \rangle \rightarrow \langle MN \rangle : T_{\tau \rightarrow \sigma}$ - это отображение будет определять $app_{\tau, \sigma} : T_{\tau \rightarrow \sigma} \rightarrow (T_\tau \rightarrow T_\sigma)$.

Таким образом, $(T_\sigma, app_{\sigma, \tau})$ будет называться **бета-моделью** термов

Бета-модель **не экстенсинальна**. Пример:

- $Q_1 = y^{\sigma \rightarrow \tau}$
- $Q_2 = \lambda x^\sigma. (y^{\sigma \rightarrow \tau} x^\sigma)$
- Для какого либо $P \in T_\sigma$ получим, что:
 - $app_{\sigma, \tau}(\langle Q_1 \rangle)(\langle P \rangle) = \langle (y^{\sigma \rightarrow \tau} P) \rangle$
 - $app_{\sigma, \tau}(\langle Q_2 \rangle)(\langle P \rangle) = \langle (y^{\sigma \rightarrow \tau} P) \rangle$
- **Не инъективно**

10. Модель бета-эта-термов

$\lambda x.(Px) \Rightarrow_\eta P$, при условии, что x не входит свободно в P

Терм $E = \lambda x.\lambda y.(xy) \Rightarrow_\eta \lambda x.x \equiv I$ - примечательно, что оба терма изначально в нормальной форме (в силу её определения через возможность бета-конверсий), поэтому $E \neq I$, хотя их действие на 2 аргумента будет идентичным

η -нормальная форма терма - форма, в которой больше нельзя применить эта-редукцию. Каждый терм за конечное число эта-редукций приводится к эта-нормальной форме (в силу того, что по определению этот тип редукции сокращает выражение, избавляясь от абстракций безвозвратно)

Конечное число применений β, η редукций будем записывать как $M \Rightarrow_{\beta\eta} N$. Это отношение рефлексивно и транзитивно, а также замкнуто относительно контекста и подстановок

Отношение $\beta\eta$ -редукции конфлюэнтно (смотри про конфлюэнтность в [основном топики про теорему Чёрча-Россера](#))

Терм в **$\beta\eta$ -нормальной форме**, если он одновременно в эта-нормальной форме и в простой нормальной форме. (очевидно, если терм приводится к нормальной форме, то он приводится и к бета-эта, верно и обратное)

Термы M, N будут **$\beta\eta$ -эквивалентны** ($M \cong_{\beta\eta} N$), если найдётся такой терм P , что $M \Rightarrow_{\beta\eta} P \wedge N \Rightarrow_{\beta\eta} P$. Несколько дополнительных утверждений без доказательства из-за их очевидности:

- Каждый класс бета-эта-эквивалентности состоит из нескольких классов равенства, потому что $M = N$ влечёт $M \cong_{\beta\eta} N$
- $x \notin (FV(M) \cup FV(N)) \wedge (Mx) \cong_{\beta\eta} (Nx) \Rightarrow M \cong_{\beta\eta} N$
- $FV(P) \cap (FV(M) \cup FV(N)) = \emptyset \wedge (MP) \cong_{\beta\eta} (NP) \Rightarrow M \cong_{\beta\eta} N$

Возьмём для M класс бета-эта-эквивалентности $< M >_{\beta\eta}$, S_σ множество классов типа σ

Тогда $S = (S_\sigma, app_{\sigma,\tau})_{\sigma,\tau \in \Sigma}$ будет называться **моделью $\beta\eta$ -термов**. В отличие от бета-модели, она будет экстенциональна в силу свойств бета-эта-эквивалентности

11. Модели Хенкина. Корректность и полнота

Как мы задавали интерпретацию типов $[[\]]_\delta \in D_\sigma$, зададим теперь интерпретацию для компонентов экстенциональной структуры представлений $(A_\sigma, app_{\sigma,\tau})$. Также индукцией по построению (считая, что $\delta(x^\sigma) \in A_\sigma$):

- $[[x^\sigma]]_\delta = \delta(x)$
- Для термов $P : (\sigma \rightarrow \tau)$, $Q : \sigma$, для которых уже заданы интерпретации $[[P]]_\delta \in A_{\sigma \rightarrow \tau}$, $[[Q]]_\delta \in A_\sigma$, то полагаем $[[PQ]]_\delta = app([P]_\delta)([Q]_\delta) \in A_\tau$
- Для терма $Q : \tau$ с $[[Q]]_\delta \in A_\tau$ полагаем, что $[[\lambda x^\tau Q]]'_\delta = (a \rightarrow [[Q]]_{\delta[x^\sigma \rightarrow a]}) : (A_\sigma \rightarrow A_\tau)$, однако нам нужно определить $A_{\sigma \rightarrow \tau}$, поэтому любая функция вида $a \rightarrow [[Q]]_{\delta[x^\sigma \rightarrow a]}$ должна дополнительно удовлетворять условию попадания в образ отображения app (**условие Н**), тогда мы можем наконец заявить, что $[[\lambda x^\sigma . Q]]_\delta = app^{-1}([[\lambda x^\tau Q]]'_\delta)$

Экстенциональную структуру, удовлетворяющую Н, назовём **моделью Хенкина**

Модель множеств является моделью Хенкина, а вот для бета-эта-модели требуется доказательство.

Для этого возьмём бета-эта-модель $(S_\sigma, app_{\sigma,\tau})$, а также определим означивание δ , которое каждой переменной будет сопоставлять некоторый класс бета-эта-эквивалентности

подходящего терма, то есть $\delta(x^\sigma) = \langle r(x^\sigma) \rangle_{\beta\eta}$, теперь можем определить интерпретацию $M[r] = [M]_{r(x_1^{\sigma_1}), \dots, r(x_n^{\sigma_n})}$

А теперь заявляем, что бета-эта-модель будет моделью Хенкина такой, что $\forall M : [[M]]_\delta = \langle M[r] \rangle_{\beta\eta}$. Доказываться это будет индукцией по построению (есть вообще в лямбдах хоть что-то, что доказывается не индукцией???)

- $M \equiv x^\sigma : [[x^\sigma]]_\delta = \delta(x^\sigma) = \langle r(x^\sigma) \rangle_{\beta\eta} = \langle [x^\sigma]_{r(x^\sigma)}^\sigma \rangle_{\beta\eta}$
- $M \equiv (PQ), P : \sigma \rightarrow \tau, Q : \tau : [[(PQ)]]_\delta = app([P]_\delta)([Q]_\delta) = \langle (P[r]Q[r]) \rangle_{\beta\eta} = \langle (PQ)[r] \rangle_{\beta\eta}$
- $M \equiv \lambda x^\sigma. Q, Q : \tau$, а также возьмём все термы $N : \sigma$. Тогда $[[Q]]_{\delta[x^\sigma \rightarrow N]} = \langle Q[r[x^\sigma \rightarrow N]] \rangle_{\beta\eta} = \langle M[r]N \rangle_{\beta\eta} = app(\langle M[r] \rangle)(\langle N \rangle)$, отсюда заключаем, что $\langle N \rangle_{\beta\eta} \rightarrow [[Q]]_{\delta[x^\sigma \rightarrow N]}$ лежит в образе app , а значит $[[\lambda x^\sigma. Q]]_\delta = \langle M[r] \rangle_{\beta\eta}$

Модель Хенкина будет **корректной**, если из $M \cong_{\beta\eta} N$ следует $[[M]]_\delta = [[N]]_\delta$ для любого означивания

Аксиомы лямбда-бета-эта-исчисления представляют собой всё те же альфа- и бета-конверсии, эта-редукцию, эквивалентность + **формулы вывода** с той лишь разницей, что добавляется обозначение типов у иксов в λx .

Т. О корректности: Все модели Хенкина в лямбда-бета-эта-исчислении корректны

Доказательство:

- Все правила вывода и свойство эквивалентности проверяются из определения модели и модели Хенкина в частности, так что отдельно остановимся лишь на альфе, бете и эте
- η : Хитро манипулируем с отображением app : $[[\lambda x^\sigma. (Mx)]]_\delta = app^{-1}(a \rightarrow [[(Mx)]]_{\delta[x^\sigma \rightarrow a]}) = app^{-1}(a \rightarrow app([M]_\delta)([x]_{\delta[x^\sigma \rightarrow a]})) = app^{-1}(app([M]_\delta)) = [M]_\delta$ (последний переход мне мало понятен, но пусть он тут остаётся)
 - **Доп. лемма:** если N свободен для x в M , то $[[M]_N^x]_\delta = [M]_{\delta[x^\sigma \rightarrow [N]_\delta]}$
- α : $[[\lambda y^\sigma [M]_y^x]]_\delta = app^{-1}(a \rightarrow [[M]_{\delta[y \rightarrow a][x \rightarrow a]}) = app^{-1}(a \rightarrow [[M]_{\delta[x \rightarrow a]}) = [[\lambda x^\sigma. M]]_\delta$
- β : $[[\lambda x^\sigma. MN]]_\delta = app([[\lambda x^\sigma. M]]_\delta)([N]_\delta) = app(app^{-1}(a \rightarrow [[M]_{\delta[x \rightarrow a]}))([N]_\delta) = (a \rightarrow [[M]_{\delta[x \rightarrow a]}])([N]_\delta) = [[M_N^x]]_\delta$

Модель Хенкина называется **полной**, если из $[[M]]_\delta = [[N]]_\delta$ для любого означивания следует $M \cong_{\beta\eta} N$

Т. О полноте:

- Модель $\beta\eta$ -термов полна
- Если в модели множеств каждый атомарный тип интерпретируется бесконечным множеством, тогда эта модель полна

Доказательство:

- Почему-то только первого пункта... Если $[[M]]_\delta = [[N]]_\delta$, то для любого означивания будет справедливо, что $\langle M[r] \rangle_{\beta\eta} = \langle N[r] \rangle_{\beta\eta}$ (при условии, разумеется, что $\delta(x^\sigma) = \langle r(x^\sigma) \rangle_{\beta\eta}$ для любого входящего в M, N икса). Если при этом возьмём $r(x) = x$, то получим, что $M[r] = M, N[r] = N \Rightarrow \langle M \rangle_{\beta\eta} = \langle N \rangle_{\beta\eta} \Rightarrow M \cong_{\beta\eta} N$

12. Основная теорема об ЛП

Будем работать с моделью Хенкина $A = (D_\sigma, app_{\sigma,\tau})$, вместо $app_{\sigma,\tau}(f)$ будем писать \hat{f}

Логический предикат (ЛП) на A - семейство $\mathbb{R} = \{R_\sigma\}$ подмножеств $R_\sigma \subseteq D_\sigma$, где:

- Для каждого атомарного типа σ множество R_σ - произвольное фиксированное подмножество D_σ
- $R_{\sigma \rightarrow \tau} = \{f | f \in D_{\sigma \rightarrow \tau} \wedge (\forall a \in R_\sigma : \hat{f}(a) \in R_\tau)\}$ или, говоря проще, $f \in R_{\sigma \rightarrow \tau} \Leftrightarrow \hat{f}(R_\sigma) \subseteq R_\tau$

Основная теореме о ЛП: если \mathbb{R} - ЛП на модели Хенкина, то $[[M]] \in R_\sigma$ для любого замкнутого терма $M : \sigma$

13. Омега-домены и домены

УМ (D, \leq) называется **ω -доменом**, если оно имеет наименьший элемент \perp и любая ω -цепь элементов из D имеет точную верхнюю грань в D

УМ (D, \leq) называется **доменом**, если оно имеет наименьший элемент \perp и любая любое направленное подмножество из D имеет точную верхнюю грань в D (вообще, немного непонятно, зачем тут говорить про направленность, ведь цепь по определению тоже направленное УМ (хах, на следующей странице и сказали, что можно заменить на любую цепь))

- Любой домен является омега-доменом
- Плоский домен является доменом
- $\omega \cup \{\perp\}$ - ω -домен
- Любое конечное УМ с наименьшим элементом - домен

14. Непрерывность и омега-непрерывность

Для УМ $(D, \leq_1), (E, \leq_2)$:

- $f : D \rightarrow E$ **монотонна**, если $\forall a, b \in D : a \leq_1 b \Rightarrow f(a) \leq_2 f(b)$
- Если УМ - омега-домены, а функция монотонна и точная верхняя грань любой омега-цепи D отображаются функцией в точные грани цепей из E , то функция **омега-**

непрерывна (для простых доменов почти также с учётом той разницы, что там у нас простые цепи, ну и свойство называется просто **непрерывностью**)