Математические модели вычислений

Исчисление конструкций

Типы произведений и сумм

Зависимые суммы (квантификация существования)

Универсумы и парадокс Бурали-Форти для исчисления конструкций

Реализации исчисления конструкций (Coq, Agda, Idris)

maxim.krivchikov@gmail.com

Материалы курса: https://maxxk.github.io/formal-models-2015/

Объявление

Нужно определиться со временем (и аудиторией) зачётов и подать их в учебную часть.
Я надеюсь, мы справимся за 2 раза (а то и за один), но нужно поставить все 3.

Слушатели не из 510 группы, которые хотят сдать спецкурс:

1. Нужно уточнить в своей учебной части о форме вашей сдачи этого спецкурса (Математические модели вычислений, 0.5 г., зачёт, обязательная дисциплина кафедры вычислительной математики), желательно заранее, чтобы, если от нашей кафедры нужно будет что-то дополнительно подать, мы бы успели это сделать.
2. Можно приходить сдавать на зачёты, можно договориться провести экзамен в какой-то день — пишите.

Исчисление Конструкций Кокана

T. Coquand, G. Huet. The calculus of constructions. 1988.

Контексты — список типов переменных (записывается как $x : A, y : B$, обозначается греческими буквами Γ)

Термы:

- типы — $\Gamma \vdash \mathbf{Type} : \mathbf{Type}^*$
- переменные (x) — $\dfrac{\Gamma \vdash A : \mathbf{Type}}{\Gamma, x : A \vdash x : A}$
- зависимые произведения ($\Pi(x : A).\, B(x)$, формация) — $\dfrac{\Gamma \vdash A : \mathbf{Type} \qquad \Gamma, x : A \vdash B(x) : \mathbf{Type}}{\Gamma \vdash \Pi(x : A).\, B(x) : \mathbf{Type}}$
- абстракция ($\lambda(x : A).\, b$, введение) — $\dfrac{\Gamma \vdash A : \mathbf{Type}, \qquad \Gamma, x : A \vdash b : B(x)}{\Gamma \vdash \lambda(x : A).\, b : \Pi(x : A).\, B(x)}$
- приложение ($M \cdot N$, удаление) — $\dfrac{\Gamma \vdash M : \Pi(x : A).\, B \qquad \Gamma \vdash N : A}{\Gamma \vdash M \cdot N : B[x := N]}$
- β-редукция — $(\lambda(x : A).\, b) \cdot z \longrightarrow_\beta b[x := z]$
- преобразование типов — $\dfrac{\Gamma \vdash M : A \qquad A \longrightarrow_\beta^* B, \qquad \Gamma \vdash B : \mathbf{Type}}{\Gamma \vdash M : B}$

Тип суммы («или»)

Элемент типа $A + B$ — это или элемент типа A или элемент типа B.

формация — $\mathbf{A + B}$

$$\frac{\Gamma \vdash A, B : \mathbf{Type}}{\Gamma \vdash A + B : \mathbf{Type}}$$

введение — $\mathbf{inl(a)}$, $\mathbf{inr(b)}$

$$\frac{a : A}{\mathrm{inl}_{\mathrm{B}}(a) : A + B}$$

$$\frac{b : B}{\mathrm{inr}_{\mathrm{A}}(b) : A + B}$$

удаление — $\mathbf{case'}$

$$\frac{C : \mathbf{Type} \qquad f_a : \Pi A.\, C \qquad f_b : \Pi B.\, C \qquad x : A + B}{\mathrm{case}'(C, f_a, f_b, x) : C}$$

редукция

$$\mathrm{case}'(C, f_a, f_b, \mathrm{inl}_{\mathrm{B}}\, a) \longrightarrow f_a \cdot a$$

$$\mathrm{case}'(C, f_a, f_b, \mathrm{inr}_{\mathrm{A}}\, b) \longrightarrow f_b \cdot a$$

зависимое удаление — $\mathbf{case}$

$$\frac{C : \Pi(t : A + B).\, \mathbf{Type} \quad f_a : \Pi(a : A).\, C \cdot (\mathrm{inl}_{\mathrm{B}}(a)) \quad f_b : \Pi(b : B).\, C \cdot (\mathrm{inr}_{\mathrm{A}}(B)) \quad x : A}{\mathrm{case}(C, f_a, f_b, x) : C \cdot x}$$

Тип произведения («и»)

Элемент типа $A \times B$ — это пара из элемента типа A и элемента типа B

формация — A × B

$$\frac{\Gamma \vdash A, B : \mathbf{Type}}{\Gamma \vdash A \times B : \mathbf{Type}}$$

введение — (a, b)

$$\frac{\Gamma \vdash A \times B : \mathbf{Type} \qquad \Gamma \vdash a : A, \qquad \Gamma \vdash b : B}{\Gamma \vdash (a, b) : A \times B}$$

удаление — $\boldsymbol{\pi_1}$(x), $\boldsymbol{\pi_2}$(x)

$$\frac{t : A \times B}{\pi_1(t) : A} \qquad \frac{t : A \times B}{\pi_2(t) : B}$$

редукция

$$\pi_1((a, b)) \longrightarrow a$$

$$\pi_2((a, b)) \longrightarrow b$$

Тип зависимой суммы («существует»)

Luo, 1989. Extended Calculus of Constructions

Элемент типа $\Sigma(x : A).\, B(x)$ — это пара из элемента a типа A и элемента b_a типа $B(a)$.

формация — $\mathbf{\Sigma(x : A).\, B(x)}$

$$\frac{A : \mathbf{Type}, \qquad \Gamma, x : A \vdash B : \mathbf{Type}}{\Gamma \vdash \Sigma(x : A).\, B : \mathbf{Type}}$$

введение — $\mathbf{(a, b)_{\Sigma(x:A).B}}$

$$\frac{a : A, \qquad b : B[x := a]}{(a, b)_{\Sigma(x:A).B} : \Sigma(x : A).\, B}$$

удаление — split

можно ввести операторы проекции $\pi_{1,2}$ как для обычного произведения

зависимое удаление:

$$\frac{C : \Pi(\Sigma(x : A).\, B).\, \mathbf{Type} \qquad r : \Pi(a : A).\, (b : B(a)).\, C \cdot (a, b)_{\Sigma A.B} \qquad x : \Sigma A.\, B}{\mathrm{split}(C, r, x) : C \cdot x}$$

редукция

$$\mathrm{split}(C, r, (a, b)_{\Sigma A.B}) \longrightarrow r \cdot a \cdot b$$

Тип произведения — это частный случай типа зависимой суммы, когда B не зависит от x. Точно так же, как и тип функции/«стрелки» $A \to B$ — это частный случай зависимого произведения ($\Pi(x : A).\, B$).

Соответствие Карри-Говарда

для исчисления конструкций

Логика	Исчисление конструкций
«и» $A \land B$	$\Sigma A.B$
«или» $A \lor B$	$A + B$
$A \to B$	$\Pi A.B$
«не» A	#0 или $\Pi C.A.C$
$\forall a.P(a)$	$\Pi(a : A).P(a)$
$\exists a.P(a)$	$\Sigma(a : A).P(a)$

«Конструктивность» логики — все доказательства представляют собой эффективные вычисления, все правила вывода тоже возвращают вычисление. Например, оператор π_1 для зависимой суммы/квантора существования вернёт тот самый x, для которого верно $P(x)$.

Теории типов и теории подтипов

Разновидности исчисления конструкций, правила вывода которых задаются с помощью суждения типизации «x : A» называют теориями типов.

Есть понятие теорий подтипов, в которых вводится понятие отношения вложения типов $X <: Y$ и правило включения $\dfrac{\Gamma \vdash x : X, \qquad X <: Y}{\Gamma \vdash x : Y}$, но у них есть сложности с нормализацией.

Универсумы

Парадокс Бурали-Форти в исчислении конструкций

Type — универсум («тип всех типов»), в общем случаи такие конструкции опасны, т.к. позволяют определить парадоксы.

Самое опасное — это рекурсивное включение универсумов, которое мы дали в первом правиле редукции сегодняшней лекции: **Type** : **Type**. Оно делает противоречивым даже System F_ω — полиморфное λ-исчисление с конструкторами типов.

Парадокс

Hurkens, 1995. A simplification of Girard's paradox.

```
2^S ≡ Pow ≡ λ (S : Type). Π S . Type
Univ ≡ Π (X : Type). (Π (Π 2^{2^X} . X). 2^{2^X})
PPUniv ≡ 2^{2^Univ}
τ ≡ λ (t : PPUniv) (X : Type) (f : Π 2^{2^X} . X ) (p : 2^X ) .
  t · (λ (x : Univ) . (p (f ((x X) f))))
σ ≡ λ (s : Univ) . ((s · Univ) (λ (t : PPUniv) . τ · t))
Δ ≡ λ (y : Univ) . (Π (Π (p : (Pow Univ)). (σ y p) (p (τ (σ y)))) #0)
Ω ≡  (τ · (λ (p : 2^Univ)) · (Π (x : Univ). (σ · x · p) · (p · x)))
False ≡ (λ (O :  (Π (p : 2^Univ) (Π (x : Univ) . (σ · x · p) · (p · x)) (p · Ω) )).
  (((O Δ) (λ (x : Univ) (t : σ x Δ) (u : (Π (p : (Pow Univ)). (σ y p) (p (τ (σ y))))).
      (u · (λ (y : Univ)) (p (τ (σ y)))))))
  · (λ (p : 2^Univ) . (O (λ (y : Univ) . p (τ (σ y)))))
  · (λ (p : 2^Univ). (v : (Π (x : Univ). (σ x p) (p x))).
        (v · Ω) · (λ (x : Univ) . (v · (τ (σ x))))))
False : #0
```

Уровни универсумов

Для того, чтобы избежать таких парадоксов, рассматривают не один универсум $\mathbf{Type} : \mathbf{Type}$, а их набор $\mathbf{Type}_i$, где i — какой-то набор индексов с заданным строгим порядком $<$ (обычно используется что-то вроде решётки, чтобы у любой пары индексов был максимальный $\max$, нестрого больший $i, j \leqslant \max(i, j)$).

Тогда правило типизации универсума преобразуется к следующему виду:

$$\frac{i < j}{\mathbf{Type}_i : \mathbf{Type}_j},$$

А правила типизации зависимой суммы и произведения — к виду:

$$\frac{\Gamma \vdash A : \mathbf{Type}_i \qquad \Gamma, x : A \vdash B(x) : \mathbf{Type}_j}{\Gamma \vdash \Pi(x : A).\, B(x) : \mathbf{Type}_k, [i, j \leqslant k]}$$

При проверке типов индексы строятся в виде ориентированного графа, рёбра которого помечены $<$ и $\leqslant$, а потом, например, алгоритмом Тарьяна, определяется, можно ли получить на основе этого графа требуемый порядок (можно, если нет ориентированного цикла с одним из рёбер, помеченных $<$)

Реализации исчисления конструкций

Coq

https://coq.inria.fr/

Agda

http://wiki.portal.chalmers.se/agda/pmwiki.php

Idris

http://www.idris-lang.org/

Интерактивные примеры в Coq

http://proofweb.cs.ru.nl/

Proof Assistant, выбрать Coq, Guest Login

Задачи со звёздочкой

Задача 7.1* Определить $\pi_{1,2}|$ через split|.

- бонусная * — определить $\eta|$-эквивалентность на зависимых суммах и определить split через $\pi_{1,2}|$

Задача 7.2** Реализовать сортировку списка натуральных чисел в Coq.

Задача 7.3** Сформулировать одну из «больших» теорем математических курсов (мат. анализ, алгебра и т.п.) в Coq

- бонусные ** — доказать теорему