Введение в Coq

Аналогия, откуда она берется?

Тип	Высказывание
$A \to B$ – функции из A в B	$A \to B$
$A \times B$ – декартово произведение	$A \wedge B$
$A \oplus B$ – дизъюнктное объединение	$A \vee B$
`void`	$\bot$
$\Pi_{x:T} B(x)$	$\forall x : T \,.\, B(x)$
$\Sigma_{x:T} B(x)$	$\exists x : T \,.\, B(x)$

Классическая ЛВ, семантика

- Язык — пропозициональные переменные, связки $\neg, \wedge, \vee, \rightarrow$, формулы.
- Формулы означают классические высказывания — текст (string) + истинностное значение (0,1).
- Логические операции ($\neg, \wedge, \vee, \rightarrow$) действуют как союзы на текстах высказываний и вычисляются по таблицам истинности на их значениях.
- Тавтологии — формулы, которые принимают значение 1 при всех допустимых интерпретациях переменных.

Классическая ЛВ, аксиоматизация (Cl)

- $F \to (G \to F)$
  $(F \to (G \to H)) \to ((F \to G) \to (F \to H))$
- $F \to (G \to F \wedge G)$
  $F \wedge G \to F \qquad F \wedge G \to G$
- $F \to F \vee G \qquad G \to F \vee G$
  $(F \to H) \to ((G \to H) \to (F \vee G \to H))$
- $(F \to \neg G) \to ((F \to G) \to \neg F)$
  $\neg\neg F \to F$

Правило вывода: $\dfrac{F \to G \quad F}{G}$ (*Modus Ponens*) .

Интуиционистская ЛВ, BHK-семантика (Брауэр, Гейтинг, Колмогоров)

- Язык — практически тот же, но для удобства добавим $\bot$ вместо отрицания; $\neg F := (F \to \bot)$.
- Формулы означают **задачи** — формулировка (string) + что считать решением.
- Логические операции на формулировках действуют как союзы.
- **Как они действуют на решениях — отдельный вопрос.** (см. дальше)
- Формула считается верной (интуиционистская тавтология), если соответствующая ей задача всегда имеет решение.

Интуиционистская ЛВ, BHK-семантика (Брауэр, Гейтинг, Колмогоров)

- Язык — практически тот же, но для удобства добавим $\bot$ вместо отрицания; $\neg F := (F \to \bot)$.
- Формулы означают **задачи** — формулировка (string) + что считать решением.
- Логические операции на формулировках действуют как союзы.
- **Как они действуют на решениях — отдельный вопрос.** (см. дальше)
- Формула считается верной (интуиционистская тавтология), если соответствующая ей задача всегда имеет решение.

В функциональном программировании: **“задача”** $\equiv$ **“тип”**.

Импликация как сведение задач

- $A \to B$ — это задача "Свести задачу B к задаче A".
- Решение — функция (метод, конструкция, алгоритм), которая каждое решение задачи A преобразует в некоторое решение задачи B.

- Требуется явное описание решения, поэтому функция вычислимая, задается программой.
- Вложенные импликации требуют функций высших типов, аргументы и значения которых сами являются функциями.
- Классические аксиомы про импликацию оказываются интуиционистски верными.

Основные операции для функционального программирования

- Аппликация (умножение): если x – решение задачи $A \to B$, а y – решение задачи A, то $(x \cdot y)$ означает решение задачи B, полученное применением функции x к аргументу y.
- λ-абстракция: если выражение t (с параметром x) при каждом значении x, являющимся решением задачи A, оказывается решением задачи B, то $(\lambda x : A.\, t)$ обозначает функциональную зависимость t от x и является решением задачи $(A \to B)$.

$\mathbf{k} := \lambda x : F.\, \lambda y : G.\, x$ – решение задачи $F \to (G \to F)$.
$\mathbf{s} := \lambda x : (F \to (G \to H)).\, \lambda y : (F \to G).\, \lambda z : F.\, ((x \cdot z) \cdot (y \cdot z))$
– решение задачи $(F \to (G \to H)) \to ((F \to G) \to (F \to H))$.

Конъюнкция: $A \wedge B = A \times B$

Решение задачи $A \wedge B$ есть упорядоченная пара, состоящая из решений задач A и B.

Операции над решениями:

- $pair : A \to (B \to A \times B)$
- $first : A \times B \to A$
- $second : A \times B \to B$

Эти операции — стандартные решения задач, соответствующих классическим аксиомам про конъюнкцию. Тем самым, эти аксиомы оказываются интуиционистски верными.

Дизъюнкция: $A \vee B = A \oplus B$ (дизъюнктное объединение, сумма)

Множество решений задачи $A \vee B$ есть объединение множеств решений задач A и B, снабженных дополнительными пометками (l или r), указывающими на задачу, которую они решают.

Операции над решениями:

- $inl : A \to A \oplus B$
- $inr : B \to A \oplus B$
- $choice : (A \to C) \to ((B \to C) \to (A \vee B \to C))$

Эти операции — стандартные решения задач, соответствующих классическим аксиомам про дизъюнкцию. Тем самым, эти аксиомы оказываются интуиционистски верными.

Ложь: $\bot = void$ *(отрицание определяется как* $\neg A := A \to \bot$*)*

Задача $\bot$ решений не имеет.

Операции над решениями:

- $c : \bot \to A$

Первая аксиома про отрицание оказывается частным случаем аксиомы для импликации:

$$(F \to (G \to \bot)) \to ((F \to G) \to (F \to \bot)).$$

Вторая аксиома НЕ выполняется: $\neg\neg F \not\to F$!

В итоге получаем интуиционистское исчисление высказываний IPC.

Исчисление IPC

- Все классические аксиомы для $\rightarrow, \wedge, \vee$,
- $\bot \rightarrow F$.

Правило вывода: $\dfrac{F \rightarrow G \quad F}{G}$ (*Modus Ponens*) .

Исчисление IPC

- Все классические аксиомы для $\rightarrow, \wedge, \vee$,
- $\bot \rightarrow F$.

Правило вывода: $\dfrac{F \rightarrow G \quad F}{G}$ (*Modus Ponens*) .

Интуиционистская логика корректна относительно классической, но слабее: $IPC \vdash F \Rightarrow Cl \vdash F$.

Исчисление IPC

- Все классические аксиомы для $\to, \wedge, \vee$,
- $\bot \to F$.

Правило вывода: $\dfrac{F \to G \quad F}{G}$ (*Modus Ponens*) .

Интуиционистская логика корректна относительно классической, но слабее: $IPC \vdash F \Rightarrow Cl \vdash F$.

Имеется эквивалентный IPC формализм естественного вывода (natural deduction), в котором перечисленные операции над решениями ($\cdot, \lambda, pair, \ldots$) и есть правила вывода. Построенные из них **термы и есть выводы**. Именно он и его обогащения используются в теориях типов и Coq'е.

Естественный вывод (отношение выводимости из гипотез)

$$x_1 : A_1, \ldots, x_n : A_n \vdash x_i : A_i$$

$$\frac{\overline{x} : \Gamma, y : A \vdash t : B}{\overline{x} : \Gamma \vdash (\lambda y.\, t) : (A \to B)} \qquad \frac{\overline{x} : \Gamma \vdash s : (A \to B) \quad \overline{x} : \Gamma \vdash t : A}{\overline{x} : \Gamma \vdash (s \cdot t) : B}$$

и аналогично для остальных операций.

Формула F выводима в IPC тогда и только тогда, когда для некоторого t выводима секвенция $\vdash t : F$ (с пустой левой частью и замкнутым термом t). Из t извлекается вывод в IPC, а по выводу в IPC строится терм t.

Секвенции в Coq'е ("задача"≡ "тип", Prop ⊂ Type)

$x_1 : A_1, x_2 : A_2(x_1), \ldots, x_n : A_n(x_1, \ldots, x_{n-1}) \vdash t(\overline{x}) : B(\overline{x})$

- Кроме типов-высказываний добавлены типы-данные (напр., Nat), понимаемые как дополнительные логические константы. (Каждое из чисел $0, 1, 2, \ldots$ доказывает, что Nat верно.)
- Левая часть секвенции декларирует переменные: "Пусть x_1 означает ...". Что означает? Доказательство (когда A_1 высказывание) или объект (когда A_1 тип данных).
- Вся секвенция – суждение о типизации терма t: "Тогда t доказывает B." – если B – высказывание, "Тогда t имеет тип B." – если B – тип данных.

Построение вывода

Цель работы в Coq'е – определить терм t посредством построения секвенциального вывода снизу-вверх, т.е. он имеет секвенцию с метапеременной ("неизвестной"):

$$x_1 : A_1, x_2 : A_2(x_1), \ldots, x_n : A_n(x_1, \ldots, x_{n-1}) \vdash (???) : B(\overline{x}).$$

Текущая ситуация (задача для пользователя, goal) изображается таблицей без (???):

$$\frac{\begin{array}{l} x_1 : A_1 \\ x_2 : A_2(x_1) \\ \vdots \\ x_n : A_n(x_1, \ldots, x_{n-1}) \end{array}}{B(x_1, \ldots, x_n)}$$

Построение вывода, тактики

Пользователь восстанавливает терм и секвенциальный вывод с помощью тактик.

- Тактики, это (встроенные) скрипты, сводящие такую задачу к подзадачам (subgoals) посредством применения секвенциальных правил снизу-вверх.
- См. `https://coq.inria.fr/distrib/current/refman/`. Рабочие тактики обычно соответствуют не базисным правилам (мелкие шаги), а их допустимым комбинациям (крупные шаги вывода). Их постигают постепенно.
- Можно их комбинировать (с помощью тактикалов), а также писать свои (есть язык тактик, крутой).
- Источник всего: `https://coq.inria.fr/`