1 Пятиминутка философии

Матлогика — наука с трудным прошлым, одна из самых древних дисциплин. В силу её долгого и ухабистого развития она используется в совершенно разных контекстах.

Общая идея матлогики — манипуляция высказываниями без особой заботы об их содержании.

Речь (Применение, начатое в трудах Аристотеля) Матлогика задаёт определённые шаблоны мышления, которые позволяют быстро комбинировать факты по известным правилам. Например, Аристотель ввёл систему силлогизмов — `https://en.wikipedia.org/wiki/Syllogism#Types` — которые позволяют последовательно и убедительно получать прописные истины уровня "каждый человек смертен, а Сократ человек — значит, Сократ смертен".

Риторика (Использование трудов Аристотеля в Средние века) Матлогика полезна своей громоздкостью, если её применять не к месту: если попытаться расписать какое-то неформальное доказательство строго, то потратится огромное количество сил. Этим можно воспользоваться: любую, даже самую слабую, идею можно так подробно и нудно расписать, что слушатель запутается в деталях и не сможет посреди гигантской горы формальных суждений выцепить ошибки.

Ассемблер для математики Лейбниц (XVII век) был недоволен тем, что ошибки в математических суждениях чаще всего происходят в каких-то граничных случаях, и предложил использовать то, что мы сейчас знаем как логику высказываний, вместе с понятием вложенности множеств как какую-то базу, на которой можно выстроить всю математику так, чтобы выводы можно было легко проверить, просто глядя на них внимательно и не включая творческое начало. С тех пор математики знают, что даже самую сложную идею, если постараться, можно выразить как набор применений правил той системы, в которой они работают; конечно, они так не делают, потому что писанины получается безумно много: например, существует "Principia Mathematica", книга, которая *полностью* формально расписывает все суждения, из-за чего доказательство $1+1=2$ частично приводится на странице 379 первого тома и завершается на странице 86 второго.

Высвобождение чистой мысли Когда-то те задачи, которыми сейчас занимаются компьютеры, требовали математиков, которые совершенно серьёзно строили — каждый раз заново — выводы, почему какая-то сложная формула верна. Математическая логика описывает, как можно какие-то шаблонные суждения автоматизировать и выполнять их посредством применения набора правил, без необходимости задумываться. Как пример, в домашних заданиях мы реализуем алгоритмы, которые манипулируют формулами, — а раньше, до изобретения этих алгоритмов, это же приходилось делать творчески. Чем больше математическая логика позволяет автоматизировать, тем меньше шагов отделяет проверку идеи от её зарождения.

2 Деревья выводов из лямбда-термов

Имея терм, можно получить соответствующий ему вывод в исчислении высказываний. Покажем это на сравнительно маленьком терме $i = A_2\ A_1\ A_1$, доказывающем $A \to A$.

Сначала возьмём вывод типа для этого терма:

$$\dfrac{\dfrac{\dfrac{}{A_2 : (A \to (B \to A) \to A) \to (A \to B \to A) \to A \to A} \quad \dfrac{}{A_1 : A \to (B \to A) \to A}}{A_2\ A_1 : (A \to B \to A) \to A \to A} \quad \dfrac{}{A_1 : A \to B \to A}}{A_2\ A_1\ A_1 : A \to A}$$

Тогда достаточно стереть сами термы, и получится вывод в ИВ:

$$\dfrac{\dfrac{}{A \to B \to A}\,A_1 \qquad \dfrac{\dfrac{}{A \to (B \to A) \to A}\,A_1 \quad \dfrac{}{(A \to (B \to A) \to A) \to (A \to B \to A) \to A \to A}\,A_2}{(A \to B \to A) \to A \to A}\,\text{MP}}{A \to A}\,\text{MP}$$

Эта эквивалентность выполняется и в том случае, если в терме есть лямбда-абстракции. Рассмотрим снова терм

```
\e c -> a8 (a9 c) id e
```

Вывод его типа:

$$\dfrac{\{e : A \lor Q, c : \neg A\} \vdash A_8 : (A \to Q) \to (Q \to Q) \to A \lor Q \to Q \quad \dfrac{\{e : A \lor Q, c : \neg A\} \vdash A_9 : \neg A \to A \to Q \quad \{e : A \lor Q, c : \neg A\} \vdash c : \neg A}{\{e : A \lor Q, c : \neg A\} \vdash A_9\ c : A \to Q}}{\{e : A \lor Q, c : \neg A\} \vdash A_8\ (A_9\ c) : (Q \to Q) \to A \lor Q \to Q}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\{e : A \lor Q, c : \neg A\} \vdash A_8\ (A_9\ c) : (Q \to Q) \to A \lor Q \to Q \quad \{e : A \lor Q, c : \neg A\} \vdash I : Q \to Q}{\{e : A \lor Q, c : \neg A\} \vdash A_8\ (A_9\ c)\ I : A \lor Q \to Q} \quad \{e : A \lor Q, c : \neg A\} \vdash e : A \lor Q}{\{e : A \lor Q, c : \neg A\} \vdash A_8\ (A_9\ c)\ I\ e : Q}}{\{e : A \lor Q\} \vdash \lambda c.\ A_8\ (A_9\ c)\ I\ e : \neg A \to Q}}{\vdash \lambda e\ c.\ A_8\ (A_9\ c)\ I\ e : A \lor Q \to \neg A \to Q}$$

Если отсюда стереть термы и переставить местами левое и правое деревья в разборе аппликации, получится:

$$\dfrac{\dfrac{\{A \lor Q, \neg A\} \vdash \neg A \quad \dfrac{}{\{A \lor Q, \neg A\} \vdash \neg A \to A \to Q}\,A_9}{\{A \lor Q, \neg A\} \vdash A \to Q}\,\text{MP} \quad \dfrac{}{\{A \lor Q, \neg A\} \vdash (A \to Q) \to (Q \to Q) \to A \lor Q \to Q}\,A_8}{\{A \lor Q, \neg A\} \vdash (Q \to Q) \to A \lor Q \to Q}\,\text{MP}$$

$$\dfrac{\dfrac{\dfrac{\{A \lor Q, \neg A\} \vdash A \lor Q \quad \dfrac{\{A \lor Q, \neg A\} \vdash Q \to Q \quad \{A \lor Q, \neg A\} \vdash (Q \to Q) \to A \lor Q \to Q}{\{A \lor Q, \neg A\} \vdash A \lor Q \to Q}\,\text{MP}}{\{A \lor Q, \neg A\} \vdash Q}\,\text{MP}}{\{A \lor Q\} \vdash \neg A \to Q}\,\text{Лемма о дедукции}}{\vdash A \lor Q \to \neg A \to Q}\,\text{Лемма о дедукции}$$

3 Теоремы как макросы

В лекции приводится несколько лемм — например, ($\land$intro). К ним можно относиться так же, как к макросам: пусть в дереве вывода нам встретился узел

$$\dfrac{A \quad B}{A \land B}\,\land\text{intro}$$

Тогда мы можем его заменить на набор узлов, которые даны в нашем исчислении:

$$\dfrac{B \quad \dfrac{A \quad \dfrac{}{A \to B \to A \land B}\,A_4}{B \to A \land B}\,\text{MP}}{A \land B}\,\text{MP}$$

4 Примеры выводов в исчислении высказываний

Всего этого не было на практике, но оно может быть полезно в домашних заданиях.

4.1 $(A \to C) \land (B \to C) \to (A \lor B) \to C$

Почему эта формула верна Из посылки $A \lor B$ следует, что либо верно A, либо верно B. Если верно A, его можно передать в $A \to C$ из пары в первой посылке; иначе верно B, его можно подать в $B \to C$.

Вывод с леммой о дедукции Видно, что в первую очередь нужно воспользоваться $A8$, потому что по смыслу внешняя операция в выводе — разбор случаев для $A \vee B$. Для этого нужно предоставить функции $A \to C$ и $B \to C$. С леммой о дедукции это просто: в контексте есть $(A \to C) \wedge (B \to C)$, откуда можно получить по $A3$ и $A4$ нужные компоненты.

1. $(A \to C) \wedge (B \to C) \vdash (A \to C) \wedge (B \to C)$
2. $(A \to C) \wedge (B \to C) \vdash (A \to C) \wedge (B \to C) \to (A \to C)$ (A_3)
3. $(A \to C) \wedge (B \to C) \vdash (A \to C) \wedge (B \to C) \to (B \to C)$ (A_4)
4. $(A \to C) \wedge (B \to C) \vdash (A \to C)$ $(\mathrm{MP}(1,2))$
5. $(A \to C) \wedge (B \to C) \vdash (B \to C)$ $(\mathrm{MP}(1,3))$
6. $(A \to C) \wedge (B \to C) \vdash (A \to C) \to (B \to C) \to (A \vee B) \to C$ (A_8)
7. $(A \to C) \wedge (B \to C) \vdash (B \to C) \to (A \vee B) \to C$ $(MP(4,6))$
8. $(A \to C) \wedge (B \to C) \vdash (A \vee B) \to C$ $(\mathrm{MP}(5,7))$
9. $\vdash (A \to C) \wedge (B \to C) \to (A \vee B) \to C$ (лемма о дедукции (8))

Вывод по соответствию Карри-Говарда Транслируя этот вывод в Haskell, получим

```
f1 :: (a -> c, b -> c) -> Either a b -> c
f1 p = a8 (a3 p) (a4 p)
```

Вывод без леммы о дедукции Полученный терм очень легко перевести по заданному алгоритму в SK-базис:

```
\p -> a8 (a3 p) (a4 p) ==
a1 a8 <*> a3 <*> a4 ==
a2 (a1 a8 <*> a3) a4 ==
a2 (a2 (a1 a8) a3) a4
```

4.2 $A \to \neg\neg A$

Почему эта формула верна Если верна формула A, то не может быть верно, что $\neg A$.

Вывод с леммой о дедукции $\neg\neg A$ верно, потому что предположение $\neg A$ приведёт к противоречию. Это выражает A_{10}. Осталось понять, какое утверждение B оказывается и истинным, и ложным, если мы предполагаем и A, и $\neg A$. Несложно догадаться, что $B = A$.

1. $A \vdash A$
2. $A \vdash A \to \neg A \to A$ (A_1)
3. $A \vdash \neg A \to A$ $(\mathrm{MP}(1,2))$
4. $A \vdash \neg A \to \neg A$ ($P \to P$ доказывалось на лекции)
5. $A \vdash (\neg A \to A) \to (\neg A \to \neg A) \to \neg\neg A$ (A_{10})
6. $A \vdash (\neg A \to \neg A) \to \neg\neg A$ $(\mathrm{MP}(3,5))$
7. $A \vdash \neg\neg A$ $(\mathrm{MP}(4,6))$
8. $\vdash A \to \neg\neg A$ (лемма о дедукции (7))

Вывод по соответствию Карри-Говарда Прямая трансляция (здесь i = a2 a1 a1):

```
f2 :: a -> Not (Not a)
f2 a = a10 (a1 a) i
```

Вывод без леммы о дедукции Как и ранее:

```
\a -> a10 (a1 a) i ==
a1 a10 <*> a1 <*> a1 i ==
a2 (a1 a10 <*> a1) (a1 i) ==
a2 (a2 (a1 a10) a1) (a1 i)
```

4.2.1 $(A \lor Q) \to \neg A \to Q$

Почему эта формула верна Мы знаем, что $A \lor Q$, то есть выполняется либо A, либо Q. Если выполняется Q, то всё в порядке. Иначе получаем, что выполняется A — но мы знаем, что выполняется $\neg A$, а отсюда следует что угодно.

Вывод с леммой о дедукции Рассуждению "$A \lor Q$, то есть выполняется либо A, либо Q" соответствует аксиома 8, которая говорит, что если нам дан $A \lor B$ и мы хотим получить C, достаточно рассмотреть, как получить C из A и из B по отдельности. Чтобы воспользоваться этой аксиомой, нужно предоставить функции $Q \to Q$ и $A \to Q$. Как получить первую из них, разобрано на лекции: это a2 a1 a1.

Рассуждению "выполняются A и $\neg A$, а отсюда следует что угодно" соответствует аксиома 9, которая утверждает, что из противоречивого контекста выводимо любое утверждение.

1. $\vdash Q \to Q$ (с лекции)
2. $\vdash \neg A \to A \to Q$ (A9)
3. $\neg A \vdash A \to Q$ (лемма о дедукции (2))
4. $\vdash (A \to Q) \to (Q \to Q) \to A \lor Q \to Q$ (A8)
5. $\neg A \vdash (Q \to Q) \to A \lor Q \to Q$ (MP(3, 4))
6. $\neg A \vdash A \lor Q \to Q$ (MP(1, 5))
7. $\neg A, A \lor Q \vdash Q$ (лемма о дедукции (6))
8. $A \lor Q \vdash \neg A \to Q$ (лемма о дедукции (7))
9. $\vdash A \lor Q \to \neg A \to Q$ (лемма о дедукции (8))

```
\e c -> a8 (\a -> a9 c a) id e
```

Упростим по эта-редукции:

```
\e c -> a8 (a9 c) id e
```

```
\e -> (\c -> a8) <*> a9 <*> (\c -> id) <*> (\c -> e)

\e -> a1 a8 <*> a9 <*> a1 id <*> a1 e

\e -> ((a1 a8 <*> a9) <*> a1 id) <*> a1 e

\e -> a2 (a2 (a2 (a1 a8) a9) (a1 id)) (a1 e)

(\e -> a2) <*> (\e -> a2 (a2 (a1 a8) a9) (a1 id)) <*> (\e -> a1 e)

a1 a2 <*> a1 (a2 (a2 (a1 a8) a9) (a1 id)) <*> a1

a2 (a2 (a1 a2) (a1 (a2 (a2 (a1 a8) a9) (a1 id)))) a1

a2 (a2 (a1 a2) (a1 (a2 (a2 (a1 a8) a9) (a1 (a2 a1 a1))))) a1
```