Лекция 4

Сегодня мы продолжим наше рассмотрение в области логики.

Мы занимались уже логикой высказываний - объяснением того, как истинность более сложного высказывания возникает из истинности более простых – это называлось семантикой, и более того, сначала мы договаривались о том, как более сложные высказывания строятся из более простых (и как выглядят самые простые высказывания) – это называется синтаксисом. Для этого давались индуктивные определения.

Вы помните, что ещё раньше у нас было индуктивное определение того, что такое терм, что такое вывод в грамматике. Сегодня мы тоже будем обращаться к индуктивным определениям – это важный инструмент в математической логике, математике вообще и информатике.

Посмотрим сначала, откуда берётся истинность или ложность такого простейшего, «атомного», высказывания.

Начнем с общего понятия отношения. Пусть имеется непустое множество D и натуральное число n. Тогда n–местным отношением R на D называется всякое подмножество n–ой степени множества D. Можно сформулировать это определение несколько иначе, считая, что отношение – это свойство наборов элементов D. Тогда на элементах R это свойство истинно, а на n–ках вне R ложно. Когда мы говорили о перечислимости, то упоминали характеристические функции множества R: на R характеристическая функция равна 1, а на n-ках вне R – 0. Конечно, все эти определения говорят об одном и том же.

Нам будет удобно говорить и об отношениях, у которых счетное число аргументов. Такое отношение – это подмножество D^{ω}. С похожей конструкцией мы уже встречались, когда говорили о семантике термов. На самом деле мы будем рассматривать всегда такие отношения, которые зависят только от конечного числа мест в этом бесконечном произведении, так сказать, от конечного числа аргументов.

Наиболее универсальным, естественным, простым и часто встречающимся отношением является просто отношение равенства, его ещё иногда называют диагональю. Это множество тех пар, где первый элемент совпадает со вторым.

Мы переходим к построению логики отношений (она еще называется логикой предикатов). До этого мы имели дело с логикой высказываний. При этом в качестве простейших высказываний рассматривались логические переменные, у нас не было возможности спросить: «О чем, о каких объектах данное высказывание?» Теперь такая возможность у нас появляется. Мы параллельно будем вводить синтаксические конструкции и их интерпретацию. Нам понадобятся несколько упорядоченных алфавитов:

- Алфавит имен предметов $Ob=<a_0, a_1, a_2,\dots>$
- Алфавит имен функций $Fun=<f_0, f_1, f_2,\dots>$, каждому имени сопоставлена его арность (число аргументов). (Иногда предметы считаются функциями с нулем аргументов – 0-арными)
- Алфавит имен отношений $Pr=<P_0, P_1, P_2,\dots>$, каждому имени сопоставлена его арность (число аргументов). (Иногда логические имена считаются отношениями с нулем аргументов – 0-арными)

Сигнатура – это такой набор из этих трех алфавитов и функции, указывающей арность для имен функций и имен отношений. Это – синтаксис. Семантика же, возникает при интерпретации имен сигнатуры.

Итак, пусть *D* – некоторое непустое множество, область модели, Σ – сигнатура. Модель (алгебраическая система) – это набор $<D, \Sigma, Зн>$, где *Зн* сопоставляет именам:

- имени предмета – элемент из *D*
- имени функции – всюду определенное отображение (операцию) над *D* (с нужным числом аргументов)
- имени отношения – отношение на *D* (с нужным числом аргументов)

Число аргументов часто обозначается указанием списка переменных в скобках, например: $f\ (x_0,\ x_1,\dots,x_{n-1})$.

Заметим, что в математике традиционно используются и другие способы записи применения функции к аргументам, например, знак + ставится между аргументами. Такая запись, где плюс пишется впереди, встречается в некоторых языках программирования и называется префиксной записью, можно считать, что запись 2+3 является инфиксной. То же самое верно и для отношений: вместо 5<7 можно писать <(5,7) и т. д.

Примеры.

- Упорядоченное поле рациональных чисел:

область модели: Q, имена предметов: 0, 1, имена функций: +, *, имена отношений: =, >.

- Свободный моноид с двумя образующими:

Область модели – множество всех слов в алфавите{a, b}, имена предметов: Λ,a,b, имя функции: поскольку основная функция – это приписывание слов, то никое имя для нее не используется, но можно было бы договориться о каком-то имени, а потом, о его устранении, в результате соглашения о сокращении записи, имя отношения: =.

Заметим, что в поле взятие обратного не удается напрямую включить в сигнатуру, поскольку это – не всюду определенная функция. (Но можно было бы ее «доопределить» и все время выделять «отдельный случай нуля» и т. д.)

Продолжим построение нашего языка, синтаксиса и семантики. Фиксируем упорядоченный алфавит свободных предметных переменных $FVar= <x_0,\ x_1,\ x_2,\dots >$. (Дальше у нас будут не только свободные переменные.)

Термы (индуктивное определение, уже было):

- Имя предмета - терм
- Переменная - терм
- Если f - имя n-арной функции и $t_0,\ t_1,\dots t_{n-1}$ - термы, то $f(t_0,\ t_1,\dots t_{n-1})$ – терм

Атомные формулы

- **Если** P - имя n-арного отношения и $t_0,\ t_1,\dots t_{n-1}$- термы, то $P(t_0,\ t_1,\dots t_{n-1})$ – атомная формула

Пример:

- $P_2(f_1(a_1,\ f_2(f_2\ (a_5,\ f_4\ (x_2)),\ x_2)))$ – атомная формула

Вот какие термы использовались в ее построении:

- a_5, $f_4(x_2)$
- $f_2(a_5,\ f_4\ (x_2))$, x_2
- a_1, $f_2(f_2(a_5,\ f_4(x_2)),\ x_2)$
- $f_1(a_1,\ f_2(f_2(a_5,\ f_4(x_2)),\ x_2))$

Эти определения согласуются с тем содержательным обсуждением, которое мы провели выше. Дадим формальное определение семантики.

Пусть задана модель: $<D,\ \Sigma,\ Зн>$ и последовательность $\alpha = \alpha_0,\ \alpha_1,\ \alpha_2,\dots$ из D^ω

- $Зн\ x_i$ на заданной последовательности – это α_i
- *Зн* терма на заданной последовательности определяется индуктивно, начиная с уже имеющихся значений имен предметов и имен переменных. Зн терма $f(t_0,\ t_1,\dots t_{n-1})$ на последовательности α – *это* элемент *D*, *получаемый применением Зн f* к значениям аргументов (это определение уже было, когда мы говорили о семантике термов).
- *Зн* атомной формулы $P\ (t_0,\ t_1,\dots t_{n-1})$ на заданной последовательности α - это элемент из {0,1}- результат применения *Зн P* к $Зн\ t_0$, $Зн\ t_1,\dots Зн\ t_{n-1}$

Видно, что определение семантики атомной формулы напоминает определение семантики терма. Однако здесь нам не понадобилось индуктивное определение – атомных формул внутри атомных формул не бывает, аргументы – это термы. (В алгоритмических языках, в отличие от логических, бывает и иначе.)

Таким образом:

- *Зн* терма – это функция $D^{\omega} \rightarrow D$
- *Зн* атомной формулы – это отображение $D^{\omega} \rightarrow \{0,1\}$, то есть - ω-местное отношение; если номера всех переменных атомной формулы меньше n, то она задает n-местное отношение.

В математических текстах встречаются переменные специального вида, вместо которых нельзя подставлять никакого значения. К таким переменным относятся, например, переменные суммирования. Скажем, в формулу

Можно подставить конкретные значения вместо x,a,n, но нельзя, скажем 4 подставить вместо k.

Продолжим описание синтаксиса и семантики языка логики предикатов (отношений). Фиксируем упорядоченный алфавит связанных предметных переменных $BVar= <u_0, u_1, u_2,\dots >$.

Нам также понадобятся символы, называемые кванторами:

$\exists$ - квантор существования (читается – «существует» и т. п.)

$\forall$ - квантор всеобщности (читается «для всех» и т. п.)

Кроме этого – еще одно обозначение: $A[x/u]$ –результат одновременной замены в слове A всех вхождений символа x (у нас это обычно будет свободная переменная) на слово u.

Теперь мы можем определить понятие формулы:

Формула (заданной сигнатуры), индуктивное определение:

- Атомные формулы – формулы.
- Если Φ, Ψ - формулы, $\tau\in\{\wedge,\vee, \rightarrow, \equiv\}$, то $(\neg\Phi)$, $(\Phi\tau\Psi)$ – формулы.
- Если Φ - формула, x – свободная переменная ($x\in FVar$), u – связанная переменная ($u\in BVar$), не входящая в Φ,

то $(\forall u\ \Phi[x/u])$, $(\exists u\ \Phi[x/u])$, – формулы (в эти формулы x – не входит).

Сокращения.

- Опускание внешних скобок
- Вместо $\forall u\ \forall v$ пишем $\forall u,v$
- Как и в логике высказываний, у нас будет появляться много скобок и мы будем их опускать естественным образом.

Определим теперь семантику. Пусть задана модель: $<D, \Sigma, Зн>$. *Зн* формулы Φ на заданной последовательности $\alpha = \alpha_0, \alpha_1, \alpha_2,\dots$ *из* D^{ω} определяется индукцией по построению формулы

- *Зн* атомной формулы $P (t_0, t_1,\dots t_{n-1})$: применяем *Зн* P к *Зн* t_0, *Зн* t_1,… *Зн* t_{n-1} Получаем 0 или 1 (уже было сегодня)
- Значением формулы $(\Phi\tau\Psi)$, где $\tau\in\{\rightarrow,\wedge,\vee, \equiv\}$,является результат применения операции τ к значениям формул Φ, Ψ. Зн $(\neg\Phi)$ = 1- Зн Φ (аналогично логике высказываний)
- *Зн* $(\forall u\ \Phi[x_i/u]]) = 1 \Leftrightarrow$ для всех последовательностей β, совпадающих с α на всех местах, кроме i-го, выполнено Зн Φ = 1
- *Зн* $(\exists u\ \Phi[x_i/u])= 1 \Leftrightarrow$ существует последовательность β, совпадающая с α на всех местах, кроме i-го, на которой Зн Φ = 1

Пусть задана модель $M=<D, \Sigma, Зн>$. Тогда истинность (=1) формулы зависит только от значений ее (свободных) переменных (от соответствующих членов последовательности α). Если все свободные переменные Φ имеют номера, меньшие n, то Φ *выражает* n-местное отношение на *D, выразимое (или определимое) в M.*

Пример. число – простое. Можно выразить через сложение и умножение.
Задача изучения свойств (наборов) элементов, выразимых в данной модели.
Еще несколько определений и комментариев.

- *Замкнутая* формула – формула без свободных переменных
- Замкнутая формула истинна (Зн =1) или ложна (Зн = 0) в данной модели

Модель является *моделью системы замкнутых формул,* если все формулы системы истинны в ней. В конкретном контексте эти формулы могут называться аксиомами, тогда мы говорим, например, о группе как о модели системы аксиом теории групп и изучаем все группы – модели этой системы.