1 Исчисление предикатов

1.1 Исчисление предикатов

1.1.1 Язык исчисления предикатов

1. Два типа: предметные и логические выражения.
2. Предметные выражения: метапеременная θ.
   (a) Предметные переменные: a, b, c, ..., метапеременные x, y.
   (b) Функциональные выражения: $f(\theta_1, \ldots, \theta_n)$, метапеременные f, g, ...
   (c) Примеры: r, $q(p(x, s), r)$.
3. Логические выражения: метапеременные α, β, γ, ...
   (a) Предикатные выражения: $P(\theta_1, \ldots, \theta_n)$, метапеременная P.
   Имена: A, B, C, ...
   (b) Связки: $(\varphi \vee \psi)$, $(\varphi \mathbin{\&} \psi)$, $(\varphi \rightarrow \psi)$, $(\neg\varphi)$.
   (c) Кванторы: $(\forall x.\varphi)$ и $(\exists x.\varphi)$.

1.1.2 Сокращения записи, метаязык

1. Метапеременные:
   (a) ψ, ϕ, π, ... — формулы
   (b) P, Q, ... — предикатные символы
   (c) θ, ... — термы
   (d) f, g, ... — функциональные символы
   (e) x, y, ... — предметные переменные
2. Скобки — как в И.В.; квантор — жадный:

$$(\forall a. \underbrace{A \vee B \vee C \rightarrow \exists b. \underbrace{D \mathbin{\&} \neg E}_{\exists b. \ldots}}_{\forall a. \ldots}) \mathbin{\&} F$$

3. Дополнительные обозначения при необходимости:
   (a) $(\theta_1 = \theta_2)$ вместо $E(\theta_1, \theta_2)$
   (b) $(\theta_1 + \theta_2)$ вместо $p(\theta_1, \theta_2)$
   (c) 0 вместо z
   (d) ...

1.1.3 Оценка исчисления предикатов

Определение. Оценка — упорядоченная четвёрка $\langle D, F, P, E \rangle$, где:

1. D — предметное множество;
2. F — оценка для функциональных символов; пусть f_n — n-местный функциональный символ:

$$F_{f_n} : D^n \rightarrow D$$

3. P — оценка для предикатных символов; пусть T_n — n-местный предикатный символ:

$$P_{T_n} : D^n \rightarrow V \qquad V = \{\text{И}, \text{Л}\}$$

4. E — оценка для предметных переменных.

$$E(x) \in D$$

1.1.4 Оценка формулы

Запись и сокращения записи подобны исчислению высказываний:

$$[\![\phi]\!] \in V, \quad [\![Q(x, f(x)) \vee R]\!]^{x:=1, f(t):=t^2, R:=\text{И}} = \text{И}$$

1. Правила для связок $\vee$, &, $\neg$, $\to$ остаются прежние;
2. $[\![f_n(\theta_1, \theta_2, \ldots, \theta_n)]\!] = F_{f_n}([\![\theta_1]\!], [\![\theta_2]\!], \ldots, [\![\theta_n]\!])$
3. $[\![P_n(\theta_1, \theta_2, \ldots, \theta_n)]\!] = P_{T_n}([\![\theta_1]\!], [\![\theta_2]\!], \ldots, [\![\theta_n]\!])$
4. $$[\![\forall x.\phi]\!] = \begin{cases} \text{И}, & \text{если } [\![\phi]\!]^{x:=t} = \text{И при всех } t \in D \\ \text{Л}, & \text{если найдётся } t \in D, \text{ что } [\![\phi]\!]^{x:=t} = \text{Л} \end{cases}$$
5. $$[\![\exists x.\phi]\!] = \begin{cases} \text{И}, & \text{если найдётся } t \in D, \text{ что } [\![\phi]\!]^{x:=t} = \text{И} \\ \text{Л}, & \text{если } [\![\phi]\!]^{x:=t} = \text{Л при всех } t \in D \end{cases}$$

1.2 Общезначимость, следование, выводимость

1.2.1 Общезначимость

Определение. Формула исчисления предикатов общезначима, если истинна при любой оценке:

$$\models \phi$$

То есть истинна при любых D, F, P и E.

1.2.2 Следование, выводимость

Рассмотрим язык исчисления предикатов. Возьмём все схемы аксиом классического исчисления высказываний и добавим ещё две схемы аксиом (здесь везде θ свободен для подстановки вместо x в φ):

11. $(\forall x.\varphi) \to \varphi[x := \theta]$
12. $\varphi[x := \theta] \to \exists x.\varphi$

Добавим ещё два правила вывода (здесь везде x не входит свободно в φ):

$$\frac{\varphi \to \psi}{\varphi \to \forall x.\psi} \text{ Правило для } \forall$$

$$\frac{\psi \to \varphi}{(\exists x.\psi) \to \varphi} \text{ Правило для } \exists$$

Определение. Доказуемость, выводимость, полнота, корректность — аналогично исчислению высказываний.

1.3 Теорема о дедукции для исчисления предикатов

1.3.1 Теорема

Теорема. Если $\Gamma \vdash \alpha \to \beta$, то $\Gamma, \alpha \vdash \beta$. Если $\Gamma, \alpha \vdash \beta$ и в доказательстве не применяются правила для кванторов по свободным переменным из α, то $\Gamma \vdash \alpha \to \beta$.

1.3.2 Следование

Определение. $\gamma_1, \gamma_2, \ldots, \gamma_n \models \alpha$, если выполнено два условия:

1. α выполнено всегда, когда выполнено $\gamma_1, \gamma_2, \ldots, \gamma_n$;
2. α не использует кванторов по переменным, входящим свободно в $\gamma_1, \gamma_2, \ldots, \gamma_n$.

Теорема. Если $\Gamma \vdash \alpha$ и в доказательстве не используются кванторы по свободным переменным из Γ, то $\Gamma \models \alpha$

1.4 Корректность

1.4.1 Теорема

Теорема. Если θ свободен для подстановки вместо x в φ, то $[\![\varphi]\!]^{x:=[\![\theta]\!]} = [\![\varphi[x := \theta]]\!]$

2 Непротиворечивое множество формул

2.1 Непротиворечивое множество формул

2.1.1 Определение

Определение. Γ — *непротиворечивое множество формул*, если $\Gamma \nvdash \alpha \,\&\, \neg\alpha$ для любого α

2.1.2 Примеры

1. непротиворечиво:
   - $\Gamma = \{A \to B \to A\}$
   - $\Gamma = \{P(x,y) \to \neg P(x,y), \forall x.\forall y.\neg P(x,y)\}$;
2. противоречиво:
   - $\Gamma = \{P \to \neg P, \neg P \to P\}$
     так как $P \to \neg P, \neg P \to P \ \vdash\ \neg P \,\&\, \neg\neg P$
3. пусть $D = \mathbb{Z}$ и $P(x) \equiv (x > 0)$, аналогом для этой модели будет $\Gamma = \{P(1), P(2), P(3), \dots\}$

2.2 Полное непротиворечивое множество формул

Определение. Γ — *полное* непротиворечивое множество замкнутых бескванторных формул, если:

1. Γ содержит только замкнутые бескванторные формулы;
2. если α — некоторая замкнутая бескванторная формула, то $\alpha \in \Gamma$ или $\neg\alpha \in \Gamma$.

Определение. Γ — *полное* непротиворечивое множество замкнутых формул, если:

1. Γ содержит только замкнутые формулы;
2. если α — некоторая замкнутая формула, то $\alpha \in \Gamma$, или $\neg\alpha \in \Gamma$.

Теорема. Пусть Γ — непротиворечивое множество замкнутых (бескванторных) формул. Тогда, какова бы ни была замкнутая (бескванторная) формула φ, хотя бы $\Gamma \cup \{\varphi\}$ или $\Gamma \cup \{\neg\varphi\}$ — непротиворечиво.

2.3 Доказательство существования моделей у непротиворечивых множеств формул в бескванторном исчислении предикатов.

2.3.1 Модель для множества формул

Определение. Моделью для множества формул F назовём такую модель $\mathcal{M}$, что при всяком $\varphi \in F$ выполнено $[\![\varphi]\!]_{\mathcal{M}} = \text{И}$
Альтернативное обозначение: $\mathcal{M} \models \varphi$.

2.3.2 Теорема

Теорема. Любое непротиворечивое множество замкнутых бескванторных формул имеет модель.

2.3.3 Доказательство теоремы о существовании модели

Лемма. Пусть φ — бескванторная формула, тогда $\mathcal{M} \models \varphi$ тогда и только тогда, когда $\varphi \in M$
Докозательство теоремы.

Пусть M — непротиворечивое множество замкнутых бескванторных формул.

По теореме о пополнении существует M' — полное непротиворечивое множество замкнутых бескванторных формул, что $M \subseteq M'$.

По лемме M' имеет модель, эта модель подойдёт для M.

2.4 Теорема Гёделя о полноте исчисления предикатов

2.4.1 Теорема

Теорема. Если M — замкнутое непротиворечивое множество формул, то оно имеет модель.

2.5 Полнота исчисления предикатов

2.5.1 Следствие

Следствие (из теоремы Гёделя о полноте). Исчисление предикатов полно.

3 Машина Тьюринга. Задача об останове, её неразрешимость. Неразрешимость исчисления предикатов.

3.1 Машина Тьюринга.

Определение. Машина Тьюринга:

1. Внешний алфавит $q_1, \dots, q_n$, выделенный символ-заполнитель q_ε
2. Внутренний алфавит (состояний) $s_1, \dots, s_k$; s_s — начальное, s_f — допускающее, s_r — отвергающее.
3. Таблица переходов $\langle k, s \rangle \Rightarrow \langle k', s', \leftrightarrow \rangle$

Определение. Состояние машины Тьюринга:

1. Бесконечная лента с символом-заполнителем q_ε, текст конечной длины.
2. Головка над определённым символом.
3. Символ состояния (состояние в узком смысле) — символ внутреннего алфавита.

3.2 Задача об останове, её неразрешимость.

3.2.1 Разрешимость.

Определение. Язык — множество строк
Определение. Язык L разрешим, если существует машина Тьюринга, которая для любого слова w переходит в допускающее состояние, если $w \in L$, и в отвергающее, если $w \notin L$.

3.2.2 Неразрешимость задачи останова.

Определение. Рассмотрим все возможные описания машин Тьюринга. Составим упорядоченные пары: описание машины Тьюринга и входная строка. Из них выделим язык останавливающихся на данном входе машин Тьюринга.
Теорема. Язык всех останавливающихся машин Тьюринга неразрешим.

3.3 Неразрешимость исчисления предикатов: доказательство

Теорема. Язык всех доказуемых формул исчисления предикатов неразрешим Т.е. нет машины Тьюринга, которая бы по любой формуле α определяла, доказуема ли она.
Доказательство. Пусть существует машина Тьюринга, разрешающая любую формулу. На её основе тогда несложно построить некоторую машину Тьюринга, перестраивающую любую машину S (с допускающим состоянием s_f и входом y) в её ограничения C и разрешающую формулу ИП $C \to \exists w_l.\exists w_r.F_{S,y}(w_l, w_r, s_f)$. Эта машина разрешит задачу останова.

4 Порядок теории (0, 1, 2). Теории первого порядка. Аксиоматика Пеано. Арифметические операции. Формальная арифметика.

4.1 Порядок теории (0, 1, 2).

4.2 Аксиоматика Пеано.

4.2.1 Натуральные числа: аксиоматика Пеано, 1889

$$\mathbb{N} : 1, 2, \dots \text{ или } \mathbb{N}_0 : 0, 1, 2, \dots$$

Определение. N (или, более точно, $\langle N, 0, (') \rangle$) *соответствует* аксиоматике Пеано, если следующее определено/выполнено:

1. Операция «штрих» $(')$: $N \to N$, причём нет $a, b \in N$, что $a \neq b$, но $a' = b'$.
   Если $x = y'$, то x назовём следующим за y, а y — предшествующим x.
2. Константа $0 \in N$: нет $x \in N$, что $x' = 0$.

3. Индукция. Каково бы ни было свойство («предикат») $P : N \to V$, если:
   (a) $P(0)$
   (b) При любом $x \in N$ из $P(x)$ следует $P(x')$

   то при любом $x \in N$ выполнено $P(x)$.

Как построить? Например, в стиле алгебры Линденбаума:

1. N — язык, порождённый грамматикой $\nu ::= \texttt{0} \mid \nu\texttt{«'»}$
2. 0 — это «0», x' — это $x \mathbin{+\!\!+}$ «'»

4.3 Арифметические операции.

4.3.1 Обозначения и определения

Определение. $1 = 0'$, $2 = 0''$, $3 = 0'''$, $4 = 0''''$, $5 = 0'''''$, $6 = 0''''''$, $7 = 0'''''''$, $8 = 0''''''''$, $9 = 0'''''''''$

Определение.

$$a + b = \begin{cases} a, & \text{если } b = 0 \\ (a + c)', & \text{если } b = c' \end{cases}$$

Например,

$$2 + 2 = 0'' + 0'' = (0'' + 0')' = ((0'' + 0)')' = ((0'')')' = 0'''' = 4$$

Определение.

$$a \cdot b = \begin{cases} 0, & \text{если } b = 0 \\ a \cdot c + a, & \text{если } b = c' \end{cases}$$

4.3.2 Коммутативность сложения.

Теорема. $a + b = b + a$

4.4 Формальная арифметика.

Определение. Формальная арифметика — теория первого порядка, со следующими добавленными нелогическими ...

1. двухместными функциональными символами $(+)$, $(\cdot)$; одноместным функциональным символом $(')$, нульместным функциональным символом 0;
2. двухместным предикатным символом $(=)$;
3. восемью нелогическими *аксиомами*:

(A1) $a = b \to a = c \to b = c$	(A5) $a + 0 = a$
(A2) $a = b \to a' = b'$	(A6) $a + b' = (a + b)'$
(A3) $a' = b' \to a = b$	(A7) $a \cdot 0 = 0$
(A4) $\neg a' = 0$	(A8) $a \cdot b' = a \cdot b + a$

4. нелогической схемой аксиом индукции $\psi[x := 0] \,\&\, (\forall x.\psi \to \psi[x := x']) \to \psi$ с метапеременными x и ψ.

5 Примитивно-рекурсивные и рекурсивные функции. функций вычисления простых чисел. Частичный логарифм. Выразимость отношений и представимость функций в формальной арифметике. Характеристические функции. Функция Аккермана.

5.1 Примитивно-рекурсивные функции

Определение (Примитивы Z, N, U, S).

1. Примитив «Ноль» (Z) $\qquad Z : \mathbb{N}_0 \to \mathbb{N}_0, \quad Z(x_1) = 0$
2. Примитив «Инкремент» (N) $\qquad N : \mathbb{N}_0 \to \mathbb{N}_0, \quad N(x_1) = x_1 + 1$
3. Примитив «Проекция» (U) — семейство функций; пусть $k, n \in \mathbb{N}_0, k \leqslant n$
   $$U_n^k : \mathbb{N}_0^n \to \mathbb{N}_0, \quad U_n^k(\overrightarrow{x}) = x_k$$
4. Примитив «Подстановка» (S) — семейство функций; пусть $g : \mathbb{N}_0^k \to \mathbb{N}_0, \quad f_1, \dots, f_k : \mathbb{N}_0^n \to \mathbb{N}_0$
   $$S\langle g, f_1, f_2, \dots, f_k\rangle(\overrightarrow{x}) = g(f_1(\overrightarrow{x}), \dots, f_k(\overrightarrow{x}))$$

5.2 Примитивная рекурсия

5.2.1 Определения.

Определение. [примитив «примитивная рекурсия», R] Пусть $f : \mathbb{N}_0^n \to \mathbb{N}_0$ и $g : \mathbb{N}_0^{n+2} \to \mathbb{N}_0$. Тогда $R\langle f, g\rangle : \mathbb{N}_0^{n+1} \to \mathbb{N}_0$, причём

$$R\langle f,g\rangle(\overrightarrow{x}, y) = \begin{cases} f(\overrightarrow{x}), & y = 0 \\ g(\overrightarrow{x}, y-1, R\langle f,g\rangle(\overrightarrow{x}, y-1)), & y > 0 \end{cases}$$

Определение. Функция f — примитивно-рекурсивна, если может быть выражена как композиция примитивов Z, N, U, S и R.
Теорема. $f(x) = x + 2$ примитивно-рекурсивна
Лемма. $f(a, b) = a + b$ примитивно-рекурсивна

5.2.2 Какие функции примитивно-рекурсивные?

1. Сложение, вычитание
2. Умножение, деление
3. Вычисление простых чисел
4. Неформально: все функции, вычисляемые конечным числом вложенных циклов `for`:

```
for (int i1 = 0; i1 < g1(x1...xn); i1++) {
    for (int i2 = 0; i2 < g2(x1...xn,i1); i2++) {
        ...
           for (int ik = 0; ik < gk(x1...xn,i1,i2...); ik++) {
               // выражение без циклов
           }
        ...
    }
}
```