

Математическая логика

(mk.cs.msu.ru → Лекционные курсы → Математическая логика (группы 318, 241))

Лекция 11

Машины Тьюринга
Теорема Чёрча
Как устроены математические доказательства
Логические исчисления

Лектор:
Подымов Владислав Васильевич

E-mail:
valdus@yandex.ru

Напоминание

Массовая задача (проблема) — это отображение $\mathfrak{T} : \mathfrak{I} \rightarrow \mathfrak{O}$
входных данных в выходные (правильные ответы)

Алгоритм — это особая совокупность действий, применяющаяся к входным данным для получения выходных данных

Алгоритмом \mathcal{A} *реализуется* частично определённое отображение $\overline{\mathcal{A}}$ входных данных в выходные (*вычисляемые ответы*)

Алгоритм \mathcal{A} *решает* задачу \mathfrak{T} , если $\overline{\mathcal{A}} = \mathfrak{T}$

Задача распознавания $\mathfrak{T}_1 : \mathfrak{I}_1 \rightarrow \{1, 0\}$

m-сводится к задаче распознавания $\mathfrak{T}_2 : \mathfrak{I}_2 \rightarrow \{1, 0\}$,

если существует алгоритм \mathcal{A} , такой что

$\overline{\mathcal{A}} : \mathfrak{I}_1 \rightarrow \mathfrak{I}_2$ — всюду определённое отображение и $\mathfrak{T}_1(i) = \mathfrak{T}_2(\overline{\mathcal{A}}(i))$

Теорема Чёрча

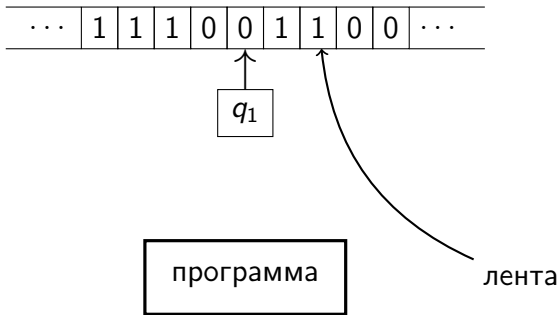
Проблема общезначимости формул логики предикатов алгоритмически неразрешима

Согласно *теореме об m -сводимости*, для обоснования теоремы Чёрча достаточно предложить

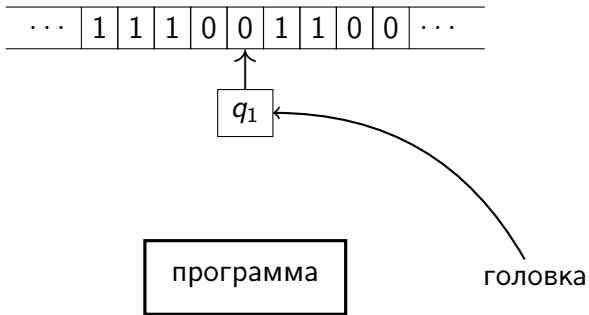
- ▶ неразрешимую задачу \mathcal{S} и
- ▶ алгоритм \mathcal{A} , m -сводящий \mathcal{S} к проблеме общезначимости формул

В качестве \mathcal{S} возьмём самую известную неразрешимую проблему:
проблему останова машин Тьюринга

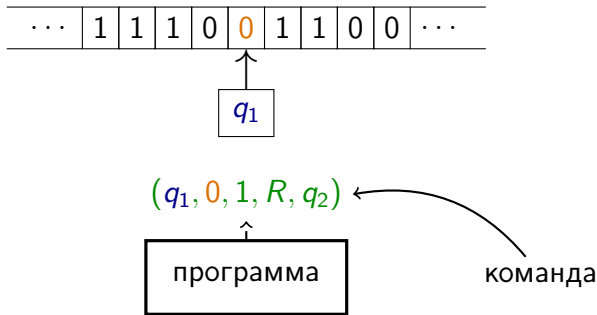
Машины Тьюринга



Машины Тьюринга



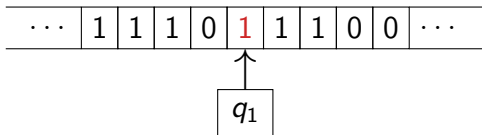
Машины Тьюринга



Шаг вычисления выглядит так:

- ▶ по текущему состоянию и обозреваемому символу выбираем команду

Машины Тьюринга



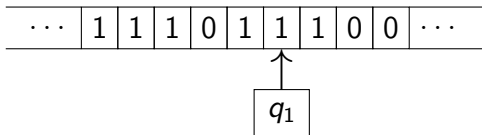
$(q_1, 0, 1, R, q_2)$

программа

Шаг вычисления выглядит так:

- ▶ по текущему состоянию и обозреваемому символу выбираем команду
- ▶ записываем в ячейку **НОВЫЙ СИМВОЛ**

Машины Тьюринга



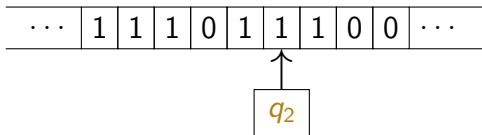
$(q_1, 0, 1, R, q_2)$

программа

Шаг вычисления выглядит так:

- ▶ по текущему состоянию и обозреваемому символу выбираем команду
- ▶ записываем в ячейку новый символ
- ▶ **сдвигаем** головку

Машины Тьюринга



$(q_1, 0, 1, R, q_2)$

программа

Шаг вычисления выглядит так:

- ▶ по текущему состоянию и обозреваемому символу выбираем команду
- ▶ записываем в ячейку новый символ
- ▶ сдвигаем головку
- ▶ **меняем состояние**

Машины Тьюринга

Определим всё с начала, по порядку и строго

Алфавит — это непустое конечное множество **символов** (букв)

Машина Тьюринга¹ (МТ) — это система $(\mathcal{A}, \Lambda, \mathcal{Q}, q_0, q_f, \mathcal{P})$, где

- ▶ \mathcal{A} — алфавит ленты
- ▶ $\Lambda \in \mathcal{A}$ — пустой символ
- ▶ \mathcal{Q} — алфавит состояний
- ▶ $q_0, q_f \in \mathcal{Q}$ — соответственно начальное состояние и заключительное состояние
- ▶ $\mathcal{P} : (\mathcal{Q} \setminus \{q_f\}) \times \mathcal{A} \rightarrow \mathcal{A} \times \{L, R\} \times \mathcal{Q}$ — программа

Программа МТ также будет пониматься как множество **команд**:

$$(q, a, b, S, p) \in \mathcal{P} \Leftrightarrow \mathcal{P}(q, a) = (b, S, p)$$

¹ “Машина Тьюринга” — это собирательный термин: придумано великое множество “примерно одинаковых” вариаций таких машин. В каждом конкретном случае принято выбирать вариацию, наиболее удобную для технических выкладок — что делается и здесь.

Машины Тьюринга

(МТ $M = (\mathcal{A}, \Lambda, \mathcal{Q}, q_0, q_f, \mathcal{P})$)

Ленточное слово — это конечная последовательность букв из \mathcal{A}

В записи слов принято опускать разделители (запятые) между символами:

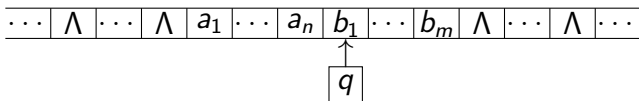
$$\overline{x_1, x_2, \dots, x_n} \quad x_1 x_2 \dots x_n$$

\mathcal{A}^* — множество всех ленточных слов

\mathcal{A}^+ — множество всех непустых ленточных слов

Конфигурация машины Тьюринга — это набор (α, q, β) , где $\alpha, \beta \in \mathcal{A}^+$ и $q \in \mathcal{Q}$

Пояснение: конфигурация $(a_1 \dots a_n, q, b_1 \dots b_m)$ означает, что МТ находится в состоянии q , на ленте записано слово $a_1 \dots a_n b_1 \dots b_m$, окружённое пустыми символами, и обозревается символ b_1



Машины Тьюринга

(МТ $M = (\mathcal{A}, \Lambda, Q, q_0, q_f, P)$)

Способ преобразования конфигураций командой C можно задать как двуместное отношение \rightarrow_C на множестве конфигураций

Если $C = (q, a, b, R, p)$, то \rightarrow_C состоит из следующих пар:

$$(\alpha, q, ax\beta) \rightarrow_C (\alpha b, p, x\beta)$$

$$(\alpha, q, a) \rightarrow_C (\alpha b, p, \Lambda)$$

Если $C = (q, a, b, L, p)$, то \rightarrow_C состоит из следующих пар:

$$(\alpha x y, q, a\beta) \rightarrow_C (\alpha x, p, y b \beta)$$

$$(y, q, a\beta) \rightarrow_C (\Lambda, p, y b \beta)$$

$$(\alpha, \beta \in \mathcal{A}^*; x, y \in \mathcal{A})$$

Отношение переходов \rightarrow_M МТ M — это объединение отношений \rightarrow_C , соответствующих всем командам M

Трасса МТ M — это последовательность конфигураций, в которой каждая пара соседних конфигураций σ_i, σ_{i+1} входит в отношение переходов: $\sigma_i \rightarrow_M \sigma_{i+1}$

Машины Тьюринга

(МТ $M = (\mathcal{A}, \Lambda, \mathcal{Q}, q_0, q_f, \mathcal{P})$)

Конфигурация (α, q, β) — **заключительная**, если $q = q_f$

Вычисление МТ M на ленточном слове w — это трасса,

- ▶ начинающаяся в конфигурации $(\Lambda, q_0, w\Lambda)$ и
- ▶ либо бесконечная,
либо оканчивающаяся в заключительной конфигурации

Утверждение. Для любой МТ M и любого ленточного слова w существует единственное вычисление M на w

Доказательство.

Достаточно заметить, что в соотношении $\sigma_i \rightarrow_M \sigma_{i+1}$

- ▶ σ_i может быть любой незаключительной конфигурацией и не может быть заключительной, и
- ▶ σ_{i+1} однозначно определяется конфигурацией σ_i ▼

Машины Тьюринга

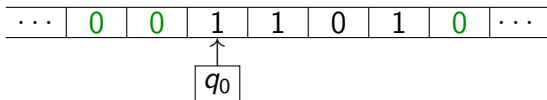
Пример: $M = (\{0, 1\}, 0, \{q_0, q_1, q_f\}, q_0, q_f, \mathcal{P})$, где:

$$\mathcal{P}(q_0, 0) = (0, L, q_1) \quad \mathcal{P}(q_1, 0) = (0, R, q_f)$$

$$\mathcal{P}(q_0, 1) = (1, R, q_0) \quad \mathcal{P}(q_1, 1) = (0, L, q_1)$$

Вычисление M на слове **1101**:

(0, q_0 , 11010)



Машины Тьюринга

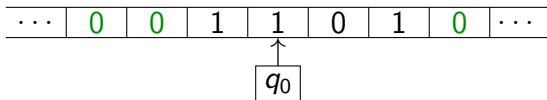
Пример: $M = (\{0, 1\}, 0, \{q_0, q_1, q_f\}, q_0, q_f, \mathcal{P})$, где:

$$\mathcal{P}(q_0, 0) = (0, L, q_1) \quad \mathcal{P}(q_1, 0) = (0, R, q_f)$$

$$\mathcal{P}(q_0, 1) = (1, R, q_0) \quad \mathcal{P}(q_1, 1) = (0, L, q_1)$$

Вычисление M на слове 1101:

$$\begin{array}{l} (0, q_0, 11010) \\ (01, q_0, 1010) \end{array} \rightarrow_M$$



Машины Тьюринга

Пример: $M = (\{0, 1\}, 0, \{q_0, q_1, q_f\}, q_0, q_f, \mathcal{P})$, где:

$$\mathcal{P}(q_0, 0) = (0, L, q_1) \quad \mathcal{P}(q_1, 0) = (0, R, q_f)$$

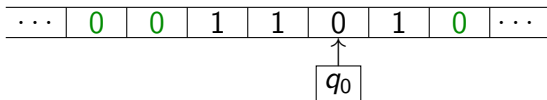
$$\mathcal{P}(q_0, 1) = (1, R, q_0) \quad \mathcal{P}(q_1, 1) = (0, L, q_1)$$

Вычисление M на слове **1101**:

$$(0, q_0, 11010) \rightarrow_M$$

$$(01, q_0, 1010) \rightarrow_M$$

$$(011, q_0, 010)$$



Машины Тьюринга

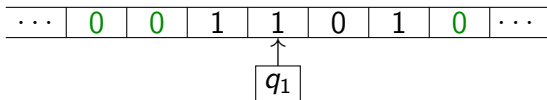
Пример: $M = (\{0, 1\}, 0, \{q_0, q_1, q_f\}, q_0, q_f, \mathcal{P})$, где:

$$\mathcal{P}(q_0, 0) = (0, L, q_1) \quad \mathcal{P}(q_1, 0) = (0, R, q_f)$$

$$\mathcal{P}(q_0, 1) = (1, R, q_0) \quad \mathcal{P}(q_1, 1) = (0, L, q_1)$$

Вычисление M на слове 1101:

$$\begin{array}{lll} (0, q_0, 11010) & \rightarrow_M & \\ (01, q_0, 1010) & \rightarrow_M & \\ (011, q_0, 010) & \rightarrow_M & \\ (01, q_1, 1010) & & \end{array}$$



Машины Тьюринга

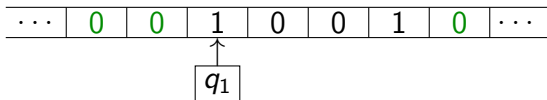
Пример: $M = (\{0, 1\}, 0, \{q_0, q_1, q_f\}, q_0, q_f, \mathcal{P})$, где:

$$\mathcal{P}(q_0, 0) = (0, L, q_1) \quad \mathcal{P}(q_1, 0) = (0, R, q_f)$$

$$\mathcal{P}(q_0, 1) = (1, R, q_0) \quad \mathcal{P}(q_1, 1) = (0, L, q_1)$$

Вычисление M на слове **1101**:

$$\begin{array}{lll} (0, q_0, 11010) & \rightarrow_M & \\ (01, q_0, 1010) & \rightarrow_M & \\ (011, q_0, 010) & \rightarrow_M & \\ (01, q_1, 1010) & \rightarrow_M & \\ (0, q_1, 10010) & & \end{array}$$



Машины Тьюринга

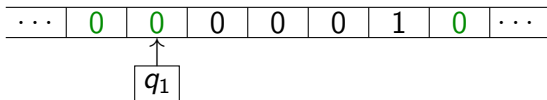
Пример: $M = (\{0, 1\}, 0, \{q_0, q_1, q_f\}, q_0, q_f, \mathcal{P})$, где:

$$\mathcal{P}(q_0, 0) = (0, L, q_1) \quad \mathcal{P}(q_1, 0) = (0, R, q_f)$$

$$\mathcal{P}(q_0, 1) = (1, R, q_0) \quad \mathcal{P}(q_1, 1) = (0, L, q_1)$$

Вычисление M на слове 1101:

$$\begin{array}{lll} (0, q_0, 11010) & \rightarrow_M & \\ (01, q_0, 1010) & \rightarrow_M & \\ (011, q_0, 010) & \rightarrow_M & \\ (01, q_1, 1010) & \rightarrow_M & \\ (0, q_1, 10010) & \rightarrow_M & \\ (0, q_1, 000010) & & \end{array}$$



Машины Тьюринга

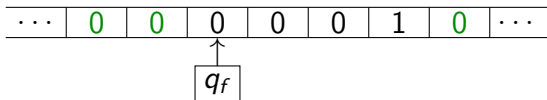
Пример: $M = (\{0, 1\}, 0, \{q_0, q_1, q_f\}, q_0, q_f, \mathcal{P})$, где:

$$\mathcal{P}(q_0, 0) = (0, L, q_1) \quad \mathcal{P}(q_1, 0) = (0, R, q_f)$$

$$\mathcal{P}(q_0, 1) = (1, R, q_0) \quad \mathcal{P}(q_1, 1) = (0, L, q_1)$$

Вычисление M на слове 1101:

$$\begin{array}{lll} (0, q_0, 11010) & \rightarrow_M & \\ (01, q_0, 1010) & \rightarrow_M & \\ (011, q_0, 010) & \rightarrow_M & \\ (01, q_1, 1010) & \rightarrow_M & \\ (0, q_1, 10010) & \rightarrow_M & \\ (0, q_1, 000010) & \rightarrow_M & \\ (00, q_f, 00010) & & \end{array}$$



Проблемы останова и общезначимости

Проблема останова машин Тьюринга — это задача распознавания **Halt** следующего вида:

- ▶ на вход подаётся пара (M, w) , где
 - ▶ M — произвольная машина Тьюринга и
 - ▶ w — произвольное ленточное слово (для M)
- ▶ $\text{Halt}(M, w) = 1 \Leftrightarrow$ вычисление M на слове w конечно

Проблема общезначимости формул логики предикатов —¹ это задача распознавания **Valid** следующего вида:

- ▶ на вход подаётся пара (σ, φ) , где
 - ▶ σ — произвольная сигнатура логики предикатов и
 - ▶ φ — произвольная формула сигнатуры σ
- ▶ $\text{Valid}(\sigma, \varphi) = 1 \Leftrightarrow \models \varphi$

¹ Это та же проблема общезначимости, что обсуждалась в предыдущих лекциях, но определённая более строго

Проблемы останова и общезначимости

Лемма(о сведении проблемы останова)

Проблема останова машин Тьюринга m -сводима к проблеме общезначимости формул логики предикатов

Доказательство.

По *определению m -сводимости*,

достаточно предложить алгоритм преобразования произвольных МТ M и ленточного слова w в сигнатуру $\sigma_{M,w}$ и формулу $\varphi_{M,w}$ этой сигнатуры, такие что $\mathbf{Halt}(M, w) = \mathbf{Valid}(\sigma_{M,w}, \varphi_{M,w})$

Сигнатуру $\sigma_{M,w}$ устроим так:

- ▶ Константы: $\mathcal{A} \cup \mathcal{Q} \cup \{\perp\}$, где $\perp \notin \mathcal{A} \cup \mathcal{Q}$
- ▶ Единственный функциональный символ: $\cdot^{(2)}$
 - ▶ Считаем этот символ ассоциативным вправо: $x \cdot y \cdot z = x \cdot (y \cdot z)$
- ▶ Единственный предикатный символ: $\text{Re}^{(3)}$

Доказательство леммы о сведении проблемы останова

$$M = (\mathcal{A}, \Lambda, Q, q_0, q_f, \mathcal{P}), w \in \mathcal{A}^* \rightsquigarrow \sigma_{M,w}, \varphi_{M,w}$$

Назовём конфигурацию σ **достижимой**, если она содержится в вычислении M на w :

- ▶ Конфигурация $(\Lambda, q_0, w\Lambda)$ достижима
- ▶ Если конфигурация σ достижима и $\sigma \rightarrow_M \sigma'$, то конфигурация σ' достижима
- ▶ Других достижимых конфигураций нет

Тогда $\text{Halt}(M, w) = 1 \Leftrightarrow$

существует достижимая заключительная конфигурация

Ленточному слову $\alpha = a_1 a_2 \dots a_k$ сопоставим терм

$$\tau_\alpha = a_1 \cdot a_2 \cdot \dots \cdot a_k \cdot \perp$$

Конфигурации $\sigma = (\alpha, q, \beta)$ МТ M сопоставим тройку термов

$\tau_\sigma = (\tau_{\alpha^-}, q, \tau_\beta)$, где α^- — **зеркальный образ** слова α :

если $\alpha = a_1 a_2 \dots a_k$, то $\alpha^- = a_k \dots a_2 a_1$

Фразе “конфигурация σ достижима” сопоставим атом $\text{Re}(\tau_\sigma)$

Доказательство леммы о сведении проблемы останова

$$M = (\mathcal{A}, \Lambda, Q, q_0, q_f, \mathcal{P}), w \in \mathcal{A}^* \rightsquigarrow \sigma_{M,w}, \varphi_{M,w}$$

Каждому правилу C , $C \in \mathcal{P}$, сопоставим формулу ψ_C :

- ▶ Если $C = (q, a, b, R, p)$, то $\psi_C = \psi_C^{R1} \& \psi_C^{R2}$, где
$$\psi_C^{R1} = \forall \alpha \forall \beta \forall x (\text{Re}(\alpha, q, a \cdot x \cdot \beta) \rightarrow \text{Re}(b \cdot \alpha, p, x \cdot \beta))$$
$$\psi_C^{R2} = \forall \alpha (\text{Re}(\alpha, q, a \cdot \perp) \rightarrow \text{Re}(b \cdot \alpha, p, \Lambda \cdot \perp))$$
- ▶ Если $C = (q, a, b, L, p)$, то $\psi_C = \psi_C^{L1} \& \psi_C^{L2}$, где
$$\psi_C^{L1} = \forall \alpha \forall x \forall \beta \forall y (\text{Re}(y \cdot x \cdot \alpha, q, a \cdot \beta) \rightarrow \text{Re}(x \cdot \alpha, p, y \cdot b \cdot \beta))$$
$$\psi_C^{L2} = \forall \beta \forall y (\text{Re}(y \cdot \perp, q, a \cdot \beta) \rightarrow \text{Re}(\Lambda \cdot \perp, p, y \cdot b \cdot \beta))$$

По *определению отношения* \rightarrow_C ,

если $\sigma \rightarrow_C \sigma'$, то $\psi_C \models \text{Re}(\tau_\sigma) \rightarrow \text{Re}(\tau_{\sigma'})$,

а значит, и $\psi_C, \text{Re}(\tau_\sigma) \models \text{Re}(\tau_{\sigma'})$

Программе \mathcal{P} сопоставим формулу $\psi_{\mathcal{P}} = \big\&_{C \in \mathcal{P}} \psi_C$

По *определению отношения* \rightarrow_M ,

если $\sigma \rightarrow_M \sigma'$, то $\psi_{\mathcal{P}}, \text{Re}(\tau_\sigma) \models \text{Re}(\tau_{\sigma'})$

Доказательство леммы о сведении проблемы останова

$$M = (\mathcal{A}, \Lambda, Q, q_0, q_f, P), w \in \mathcal{A}^* \rightsquigarrow \sigma_{M,w}, \varphi_{M,w}$$

Подходящая формула $\varphi_{M,w}$ устроена так: $\varphi_{M,w} = \psi_0 \& \psi_P \rightarrow \psi_f$, где

- ▶ $\psi_0 = \text{Re}(\tau_{(\Lambda, q_0, w\Lambda)})$ (“конфигурация $(\Lambda, q_0, w\Lambda)$ достижима”)
- ▶ $\psi_f = \exists \alpha \exists \beta \text{Re}(\alpha, q_f, \beta)$
 (“существует достижимая заключительная конфигурация”)

Покажем, что $\text{Halt}(M, w) = 1 \Leftrightarrow \text{Valid}(\sigma_{M,w}, \varphi_{M,w}) = 1$

(\Leftarrow): $\models \varphi_{M,w}$, а значит,

формула $\varphi_{M,w}$ выполняется в такой интерпретации \mathcal{I} :

- ▶ предметная область — все конфигурации M
- ▶ $\overline{\text{Re}}(t_\alpha, t_q, t_\beta) = \mathbf{t} \Leftrightarrow$
тройка (t_α, t_q, t_β) соответствует достижимой конфигурации

Тогда $\mathcal{I} \models \psi_0 \& \psi_M$

(это формульная запись определения достижимости)

Значит, верно и $\mathcal{I} \models \psi_f$:

существует достижимая заключительная конфигурация

Доказательство леммы о сведении проблемы останова

$$M = (\mathcal{A}, \Lambda, \mathcal{Q}, q_0, q_f, \mathcal{P}), w \in \mathcal{A}^* \rightsquigarrow \sigma_{M,w}, \varphi_{M,w}$$
$$\varphi_{M,w} = \psi_0 \& \psi_{\mathcal{P}} \rightarrow \psi_f; \psi_0 = \text{Re}(\tau_{(\Lambda, q_0, w\Lambda)}); \psi_f = \exists \alpha \exists \beta \text{Re}(\alpha, q_f, \beta)$$
$$\text{Halt}(M, w) = 1 \Leftrightarrow \text{Valid}(\sigma_{M,w}, \varphi_{M,w}) ?$$

(\Rightarrow): Рассмотрим (конечное) вычисление M на w :

$$(\Lambda, q_0, w\Lambda) \rightarrow_M (\alpha_1, q_1, \beta_1) \rightarrow_M \cdots \rightarrow_M (\alpha_n, q_n, \beta_n); q_n = q_f$$

Справедливы следующие соотношения:

- ▶ $\psi_0, \psi_{\mathcal{P}} \models \text{Re}(\tau_{(\alpha_1, q_1, \beta_1)})$
- ▶ $\text{Re}(\tau_{(\alpha_1, q_1, \beta_1)}), \psi_{\mathcal{P}} \models \text{Re}(\tau_{(\alpha_2, q_2, \beta_2)})$
- ▶ ...
- ▶ $\text{Re}(\tau_{(\alpha_{n-1}, q_{n-1}, \beta_{n-1})}), \psi_{\mathcal{P}} \models \text{Re}(\tau_{(\alpha_n, q_f, \beta_n)})$

При этом $\text{Re}(\tau_{(\alpha_n, q_f, \beta_n)}) \models \exists \alpha \exists \beta \text{Re}(\alpha, q_f, \beta)$

Следовательно, $\psi_0, \psi_{\mathcal{P}} \models \psi_f$

По *теореме о логическом следствии*, верно и $\models \psi_0 \& \psi_{\mathcal{P}} \rightarrow \psi_f \blacktriangledown$

Теорема Чёрча

Проблема общезначимости формул логики предикатов алгоритмически неразрешима

Доказательство.

По *лемме о сведении проблемы останова*, эта проблема (**Halt**) m -сводима к проблеме общезначимости формул логики предикатов (**Valid**)

Известно, что проблема **Halt** алгоритмически неразрешима

Значит, по *теореме об m -сводимости*, неразрешима и проблема **Valid** ▼

Интерлюдия

Утверждение. $\models A \& B \rightarrow A \vee B$

Попробуем обосновать это утверждение так,
как это принято в “обычных” математических доказательствах,
не используя особые методы
(семантических таблиц, резолюций, ...)

Доказательство. Предположим, что утверждение неверно

Тогда верно $A \& B$ и неверно $A \vee B$

Так как верно $A \& B$, верно и A

Так как верно A , верно и $A \vee B$
что противоречит полученному выше

Значит, предположение неверно ▼

Действительно ли этим “доказательством”
доказывается утверждение?

Интерлюдия

Утверждение. $\models A \& B \rightarrow A \vee B$

Доказательство. Предположим, что верно $A \& B$

Тогда, в частности, верно A

Значит, верно и $A \vee B$

Так как в предположении о верности $A \& B$ обоснована верность $A \vee B$, то верно и $A \& B \rightarrow A \vee B \blacktriangledown$

Доказательство. Это очевидно \blacktriangledown

Доказательство. Достаточно заметить, что из $A \& B$ всегда следует $A \vee B \blacktriangledown$

Доказательство. В лекциях под словом “утверждение” записываются только верные утверждения, а значит, это утверждение верно \blacktriangledown

Как отличить “правильные” доказательства от “неправильных”?

Что такое “доказательство”?

Как устроены доказательства

Прежде всего, **доказательство** — это текст, (*явно или неявно*) начинающийся словом “Доказательство” и оканчивающийся фразой “Что и требовалось доказать”

Основная часть доказательства — это последовательность **высказываний**, каждое из которых допускает ровно одну из двух оценок: **высказывание верно**, или **высказывание неверно** — например,

- ▶ $2 \times 2 = 4$: *всем известно, что* это верно
- ▶ **рассматриваемая последовательность чисел s монотонна**: читатель может и не понимать, монотонна ли s , но каждая последовательность чисел либо монотонна, либо нет
- ▶ $P \neq NP$: никто не знает, совпадают ли классы сложности P и NP , но в любом случае они либо совпадают, либо нет

Как устроены доказательства

Высказывания доказательства особым образом связаны между собой

Некоторые высказывания считаются **верными без доказательства**,

и их можно записать в любом месте доказательства — например,

- ▶ аксиомы и определения:

“отношение эквивалентности транзитивно, а значит ...”

- ▶ некоторые логические законы:

“А либо верно, либо неверно, и третьего не дано”

- ▶ теоремы, доказанные ранее:

“по теореме компактности Мальцева, ...”

- ▶ текущие предположения:

“предположим, что формула φ необщезначима”

Как устроены доказательства

Высказывания доказательства особым образом связаны между собой

Остальные высказывания должны следовать друг за другом согласно особым **правилам** (*логическим законам*) — например,

- ▶ правило отделения, оно же modus ponens:
если A верно и из A следует B , то B верно
- ▶ переход к частному:
если A верно для **всех** предметов,
то A верно и для **этого** предмета x
- ▶ приведение к абсурду, оно же рассуждение от противного:
если из верности A следует, что B и верно, и неверно,
то A неверно

В конце доказательства утверждения должна быть *так или иначе* записана фраза “**Утверждение верно**”, и эта фраза должна быть признана верной согласно используемым правилам (“**Утверждение доказано**”, “**Что и требовалось доказать**”, “**Ч.т.д.**”, “**▼**”, “**■**”, ...)

Как устроены доказательства

Вернёмся к примеру:

Утверждение. Формула $A \& B \rightarrow A \vee B$ обязательно верна

Доказательство. Предположим, что верно $A \& B$

Тогда, в частности, верно A

Значит, верно и $A \vee B$

Так как в предположении о верности $A \& B$

обоснована верность $A \vee B$, верно и $A \& B \rightarrow A \vee B \blacktriangledown$

Зелёный текст и знак “ \blacktriangledown ” — основная часть доказательства: последовательность (*верных*) высказываний, оканчивающаяся формулировкой утверждения

Остальной текст доказательства обозначает то, как одни высказывания получаются из других согласно правилам построения доказательства

Как устроены доказательства

Чтобы научиться отличать “правильные” доказательства от “неправильных”, следует определиться с тем,

- ▶ что такое **высказывание**
- ▶ по каким **правилам** из одних верных высказываний получаются другие верные высказывания
- ▶ какие высказывания **верны без доказательства**

Если определить эти понятия и свойства **математически строго**, то в результате получится система, в рамках которой можно строго определить понятие доказательства:

ЛОГИЧЕСКОЕ ИСЧИСЛЕНИЕ

Логические исчисления

Логическое исчисление включает в себя

- ▶ **алфавит**: множество символов, используемых для записи высказываний
- ▶ **синтаксис формул**: правила, по которым из символов строятся высказывания (**формулы**)
- ▶ множество **аксиом**: формул, верных без доказательства
- ▶ множество **правил вывода**, согласно которым можно получать одни формулы из других

В исчислениях встречаются формулы самых разнообразных и причудливых видов

В **исчислениях высказываний** формулами являются *формулы логики высказываний* (или наборы таких формул)

В **исчислениях предикатов** формулами являются *формулы логики предикатов* (или наборы таких формул)

Логические исчисления

Схема формулы — это запись, отличающаяся от формулы тем, что в произвольных местах вместо элементов формулы могут записываться **параметры**

Схемой Φ с параметрами \tilde{p}^k **порождается** любая формула

$$\Phi \llbracket p_1/v_1, \dots, p_k/v_k \rrbracket,$$

в которой на месте каждого вхождения каждого параметра p_i , $1 \leq i \leq k$, записано соответствующее **значение** v_i

Пример: $A \rightarrow (B \rightarrow A)$ — схема формулы с параметрами A и B , и если значения параметров — формулы логики высказываний, то

- ▶ схемой порождаются, например, формулы

$$x \rightarrow (x \rightarrow x) \quad \text{и} \quad x \vee y \rightarrow (z \& u \rightarrow x \vee y)$$

- ▶ схемой **не** порождаются, например, формулы

$$x \& y \quad \text{и} \quad x \rightarrow (x \rightarrow y)$$

Бесконечное множество **аксиом** исчисления нередко задаётся в виде конечного набора схем формул

Логические исчисления

Правило вывода местности n — это $(n + 1)$ -местное отношение на множестве формул исчисления

Формула φ **выводится** из формул $\varphi_1, \dots, \varphi_n$ по n -местному правилу R , если $(\varphi_1, \dots, \varphi_n, \varphi) \in R$

Правило вывода местности n обычно записывается так:

$$\frac{\Phi_1, \dots, \Phi_n}{\Phi}, \quad (*)$$

где $\Phi_1, \dots, \Phi_n, \Phi$ — схемы формул с общим набором параметров (p_1, \dots, p_k)

Правило, записанное в виде $(*)$, — это множество наборов $(\Phi_1 \llbracket p_1/v_1, \dots, p_k/v_k \rrbracket, \dots, \Phi_n \llbracket p_1/v_1, \dots, p_k/v_k \rrbracket, \Phi \llbracket p_1/v_1, \dots, p_k/v_k \rrbracket)$ для всевозможных значений v_1, \dots, v_k параметров p_1, \dots, p_k

Логические исчисления

Вывод формулы φ_k из множества формул Γ (в исчислении \mathcal{E}) — это последовательность формул

$$\varphi_1, \varphi_2, \dots, \varphi_k,$$

в которой для каждой формулы φ_i , $1 \leq i \leq k$, выполнено хотя бы одно из трёх условий:

1. $\varphi_i \in \Gamma$
2. φ_i — аксиома исчисления \mathcal{E}
3. Существуют формулы $\varphi_{j_1}, \dots, \varphi_{j_n}$ и n -местное правило вывода R исчисления \mathcal{E} , такие что
 - ▶ $j_1 < i, \dots, j_n < i$ и
 - ▶ φ_i получается из $\varphi_{j_1}, \dots, \varphi_{j_n}$ по правилу R

Если существует вывод формулы φ из множества формул Γ , то формула φ **выводима** из Γ

Если существует вывод \mathcal{D} формулы φ из множества \emptyset , то такой вывод называется **доказательством** формулы φ , и сама формула называется **доказуемой**

Логические исчисления

Пример: резолютивное исчисление дизъюнктов

Формулы исчисления — дизъюнкты логики предикатов

Множество аксиом исчисления пусто

Два правила исчисления — *правило резолюции* и *правило склейки*:

$$\frac{D_1 \vee A_1, D_2 \vee \neg A_2}{(D_1 \vee D_2)\theta}$$

$$\frac{D_1 \vee L_1 \vee L_2}{(D_1 \vee L_1)\eta}$$

Параметры этих правил — $D_1, D_2, A_1, A_2, \theta, L_1, L_2, \eta$

Значения параметров — соответственно два дизъюнкта, два атома, наиболее общий унификатор этих атомов,

две литеры и наиболее общий унификатор этих литер

По правилу резолюции

из пары дизъюнктов выводится их *резольвента*

По правилу склейки из дизъюнкта выводится его *склейка*

Понятия вывода и выводимости совпадают с понятиями

резолютивного вывода и *резолютивной выводимости*