Основы математической логики и логического программирования

ЛЕКТОР: В.А. Захаров

Лекция 16.

Алгоритмическая полнота логических программ. Моделирование машин Тьюринга логическим программами. Теорема Черча.

АЛГОРИТМИЧЕСКАЯ ПОЛНОТА

Вопросы:

А что могут вычислять хорновские логические программы?

Какие задачи можно решать с их помощью, а какие нельзя?

АЛГОРИТМИЧЕСКАЯ ПОЛНОТА

Есть много разных моделей вычислений. Одни из них проводят вычисления над строками (словами), другие — над графами, третьи — над молекулами ДНК, и т. п. Как сравнить их вычислительные способности?

Структуры данных, над которыми работают эти модели, позволяют кодировать натуральные числа. Поэтому чтобы сравнить вычислительные способности алгоритмических моделей, достаточно выяснить, какие функции натурального аргумента способны вычислять эти модели.

Исследования показали, что все известные **алгоритмические модели** способны вычислять только **частично-рекурсивные функции** натурального аргумента. Это открытие дало основание многим математикам (Тьюринг, Черч, Клини, Гедель, Марков и др.) выдвинуть следующий тезис.

АЛГОРИТМИЧЕСКАЯ ПОЛНОТА

Тезис Черча

Класс эффективно (алгоритмически) вычислимых функций арифметических функций в точности совпадает с классом арифметических функций, вычислимых в каждой из перечисленных ниже моделей вычислений

- машины Тьюринга—Поста,
- λ-исчисление Черча—Клини,
- системы равенств Эрбрана—Геделя,
- алгорифмы Маркова,
- системы Колмогорова—Шенхаге,
- машины Минского,
- ...

Модели вычислений такого вида называются **алгоритмически полными** . Найдется ли место в этом ряду для хорновских логических программ?

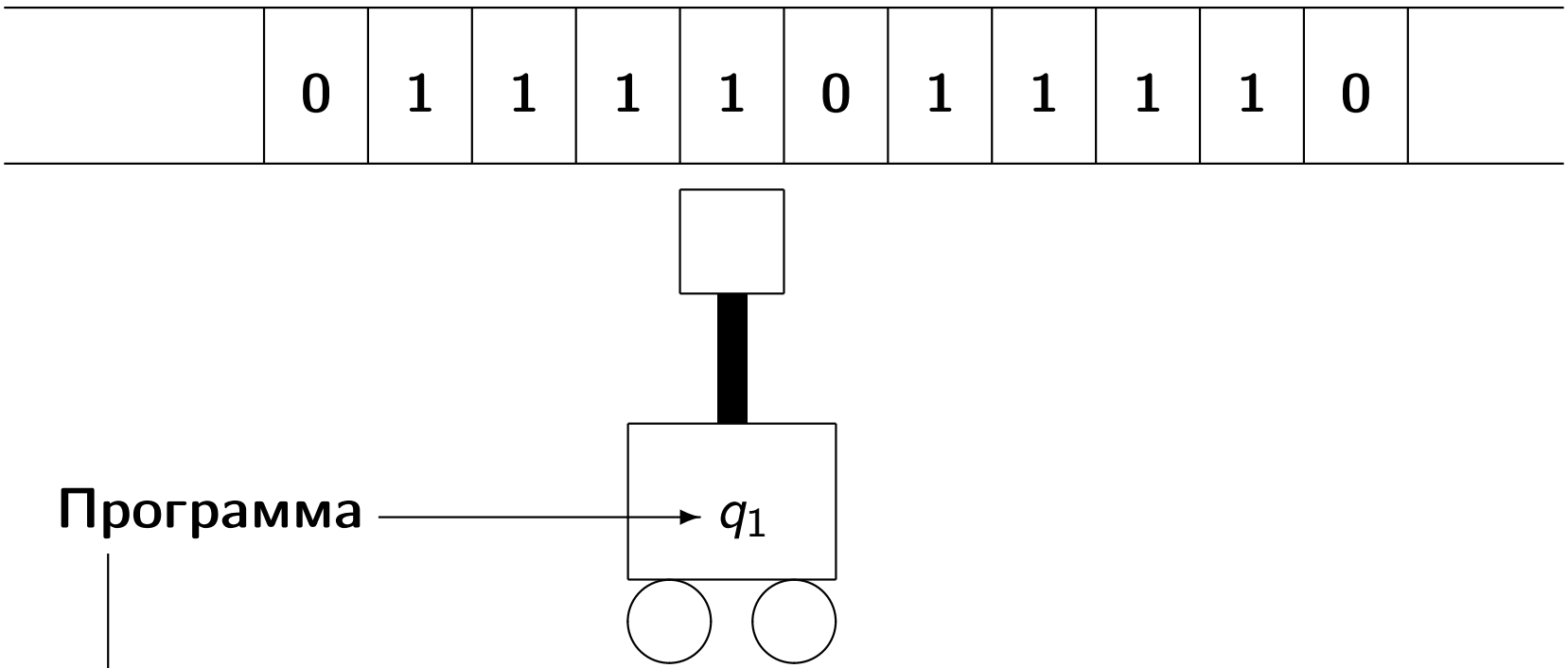
АЛГОРИТМИЧЕСКАЯ ПОЛНОТА

Для того чтобы убедиться в алгоритмической полноте хорновских логических программ, достаточно взять любую из перечисленных алгоритмически полных моделей вычислений (например, машины Тьюринга) и показать, что для любой программы П в выбранной модели найдется подходящая логическая программа $\mathcal{P}_{\text{П}}$, воспроизводящая (моделирующая) вычисления программы П.

Итак, вспомним, как устроены машины Тьюринга, и попробуем построить логическую программу-интерпретатор машин Тьюринга.

АЛГОРИТМИЧЕСКАЯ ПОЛНОТА

Машины Тьюринга

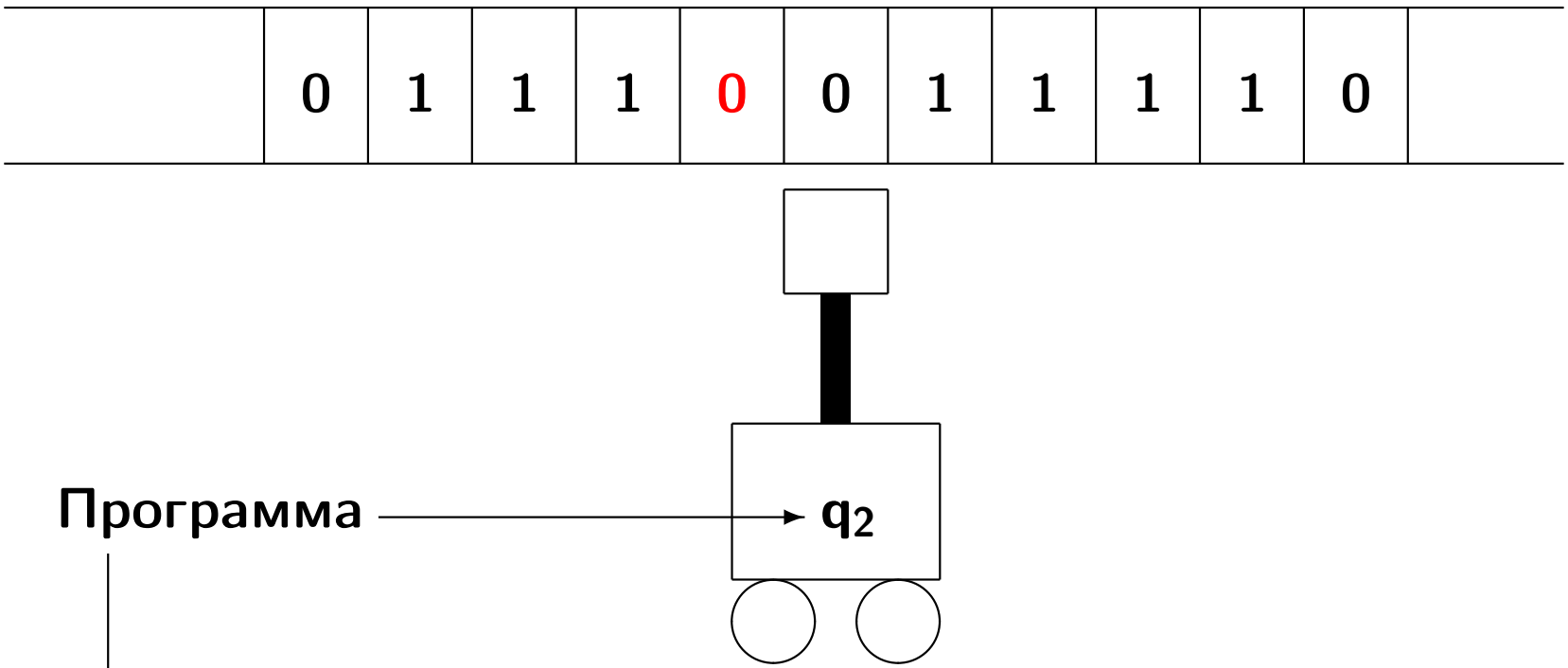


Команды:

увидев "1", стереть его, записать "0" и сдвинуться вправо

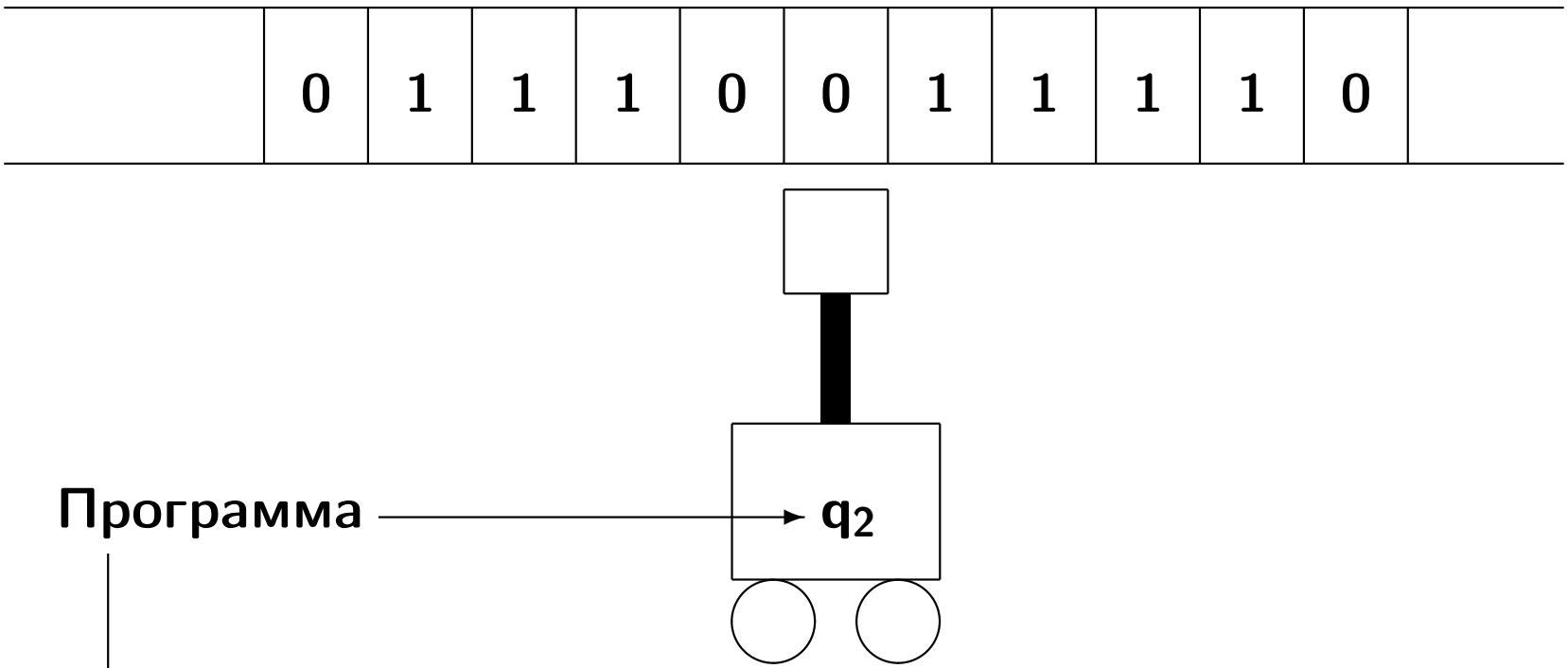
АЛГОРИТМИЧЕСКАЯ ПОЛНОТА

Машины Тьюринга



АЛГОРИТМИЧЕСКАЯ ПОЛНОТА

Машины Тьюринга

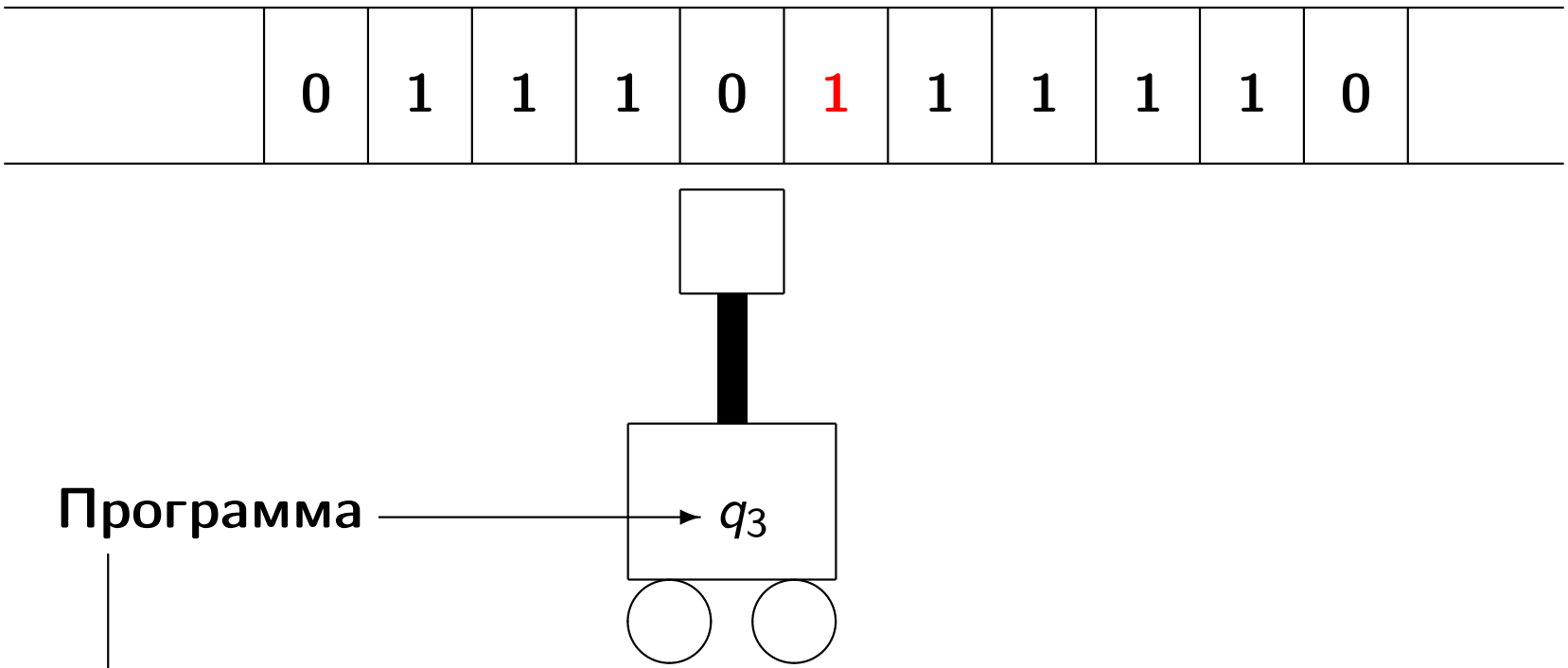


Команды:

увидев "0", стереть его, записать "1" и сдвинуться влево

АЛГОРИТМИЧЕСКАЯ ПОЛНОТА

Машины Тьюринга



АЛГОРИТМИЧЕСКАЯ ПОЛНОТА

Машины Тьюринга

Более строго модель вычислений машин Тьюринга описывается так.

Задан ленточный алфавит $\mathcal{A} = \{a_0, a_1, a_2, \ldots, a_n\}$, в котором особо выделен одна из букв a_0 (пустой символ).

Ленточным словом называется всякое слово в алфавите $\mathcal{A}$. Множество всех ленточных слов обозначим $\mathcal{A}^*$. Для каждого слова $w = z_1 z_2 \ldots z_{n-1} z_n$ будем использовать запись w^{-1} для обозначения обратного слова $z_n z_{n-1} \ldots z_2 z_1$.

Задан алфавит состояний $Q = \{q_0, q_1, q_2, \ldots, q_m\}$, в котором особо выделено одно из состояний q_0 (начальное состояние).

АЛГОРИТМИЧЕСКАЯ ПОЛНОТА

Машины Тьюринга

Ленточной конфигурацией называется всякое слово вида

$$w'\ q\ x\ w'', \qquad \text{где}$$

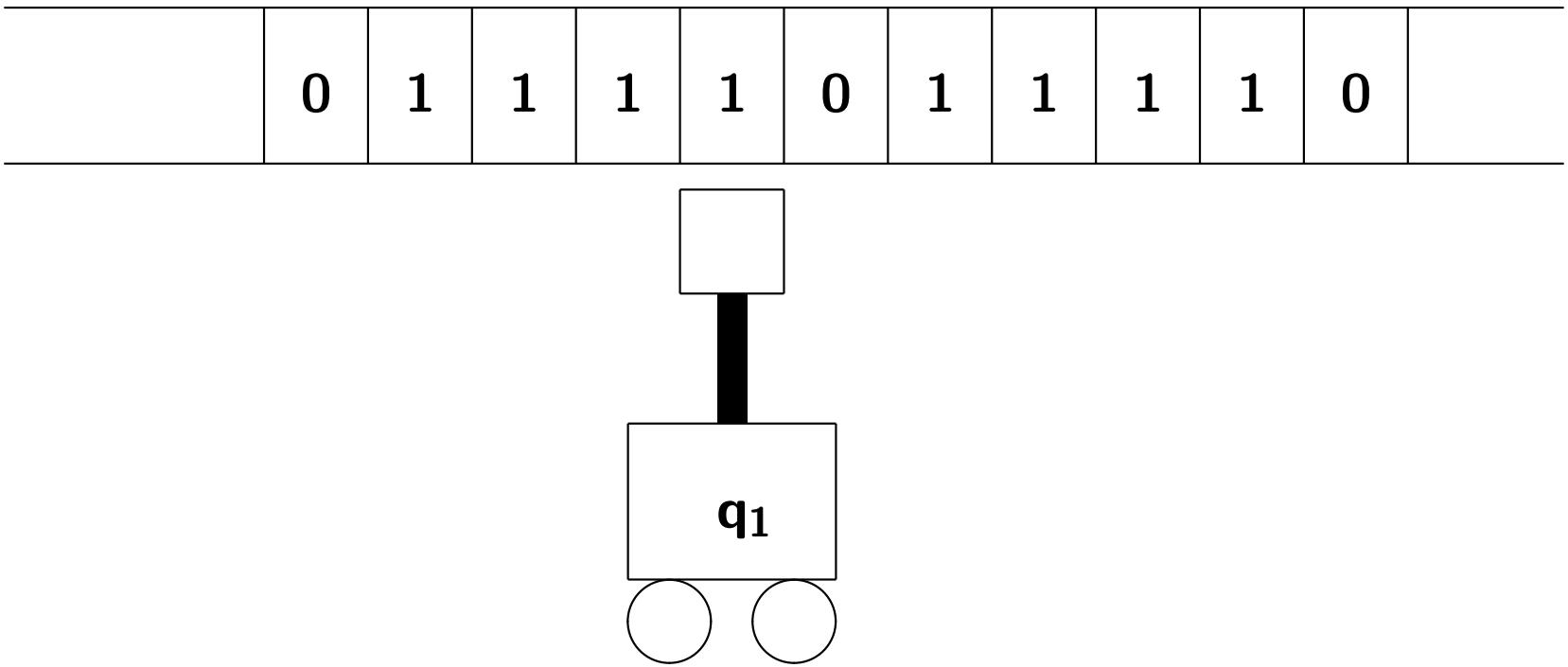
- q — состояние, $q \in Q$,
- x — ленточная буква, $x \in \mathcal{A}$,
- w', w'' — ленточные слова, $w', w'' \in \mathcal{A}^*$.

- q — это то состояние, в котором находится МТ,
- x — это буква, которая записана в той ячейке ленты, которую обозревает считывающая головка МТ,
- w' — это ленточное слово, составленное из символов, записанных **слева** от обозреваемой ячейки,
- w'' — это ленточное слово, составленное из символов, записанных **справа** от обозреваемой ячейки.

По умолчанию считается, что во всех остальных ячейках ленты записаны пустые символы.

АЛГОРИТМИЧЕСКАЯ ПОЛНОТА

Машины Тьюринга



Ленточная конфигурация: $0111q_11011110$

АЛГОРИТМИЧЕСКАЯ ПОЛНОТА

Машины Тьюринга

Ленточная конфигурация вида $u\ q_0\ x\ v$ называется начальной конфигурацией .

Множество всех конфигураций обозначим $Conf_{\mathcal{A},Q}$.

Множество всех начальных конфигураций обозначим $Conf^0_{\mathcal{A},Q}$.

АЛГОРИТМИЧЕСКАЯ ПОЛНОТА

Машины Тьюринга

Командой называется всякая пятерка вида

$$q\ x\ y\ q'\ D, \qquad \text{где}$$

- q, q' — состояния, $q, q' \in Q$,
- x, y — ленточные буквы, $x, y \in \mathcal{A}$,
- D — направление сдвига головки, $D \in \{L, R\}$.

Эту команду нужно понимать так:
если МТ находится в состянии q и обозревает символ x, то записать в обозреваемую ячейку символ y, перейти в состояние q' и сдвинуть считывающую головку на одну ячейку в направлении D.

АЛГОРИТМИЧЕСКАЯ ПОЛНОТА

Машины Тьюринга

Каждая команда K задает отношение перехода $\to_K$ на множестве ленточных конфигураций

$$\alpha \ \to_K \ \beta$$

Оно определяется так:

- если $\alpha = uzqxv$ и $K = qxyq'L$, то $\beta = uq'zyv$,
- если $\alpha = qxv$ и $K = qxyq'L$, то $\beta = q'a_0yv$,
- если $\alpha = uqxzv$ и $K = qxyq'R$, то $\beta = uyq'zv$,
- если $\alpha = uqx$ и $K = qxyq'R$, то $\beta = uyq'a_0$.

АЛГОРИТМИЧЕСКАЯ ПОЛНОТА

Машины Тьюринга

Программа машины Тьюринга — это произвольное множество команд $\pi = \{K_1, K_2, \ldots, K_N\}$. Программа π называется детерминированной , если для любых двух команд этой программы
$K_i = q_i x y q_i' D_i$,
$K_j = q_j z t q_j' D_j$,
выполняется хотя бы одно из двух условий $q_i \neq q_j$, $x \neq z$.

Программа π задает отношение переходов на множестве ленточных конфигураций $\rightarrow_\pi = \bigcup\limits_{K \in \pi} \rightarrow_K$.

Конфигурация α называется заключительной для программы π, если не существует никакой конфигурации β, для которых выполняется $\alpha \rightarrow_\pi \beta$. Это означает, что $\alpha = u\ q\ x\ v$, и в программе π нет ни одной команды $K = qx\ldots$.

АЛГОРИТМИЧЕСКАЯ ПОЛНОТА

Машины Тьюринга

Вычислением машины Тьюринга с программой π на начальной конфигурации $\alpha_0, \alpha_0 \in Conf^0$ называется последовательность конфигураций

$$\pi(\alpha_0) = \alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_i, \alpha_{i+1}, \ldots$$

удовлетворяющая следующим условиям:

- для любого $i, i \geq 0$, верно $\alpha_i \to_\pi \alpha_{i+1}$;
- последовательность $\pi(\alpha_0)$ либо является бесконечной, либо заканчивается заключительной конфигурацией α_N.

В последнем случае α_N называется результатом вычисления . Результат бесконечного вычисления считается неопределенным. Ясно, что вычисление $\pi(\alpha_0)$ детерминированной машины Тьюринга однозначно определяется начальной конфигурацией α_0 и программой π.

МОДЕЛИРОВАНИЕ МАШИН ТЬЮРИНГА ЛОГИЧЕСКИМИ ПРОГРАММАМИ

Чтобы логические программы могли моделировать вычисления машин Тьюринга, нужно выбрать подходящие логические конструкции для представления конфигураций и команд.

Ленточные слова будем представлять списками:
слово $w = z_1 z_2 \ldots z_n$ будет представлено списком
$\mathit{list}(w) = z_1 \,.\, z_2 \,.\, \ldots \,.\, z_n \,.\, \mathbf{nil}$.

Каждая ленточная конфигурация $\alpha = u\ q\ z\ v$ будет представлена парой списков $\mathit{left}(\alpha),\ \mathit{right}(\alpha)$, где
$\mathit{left}(\alpha) = \mathit{list}(u^{-1})$,
$\mathit{right}(\alpha) = q \,.\, z \,.\, \mathit{list}(v)$.

Например,
$\mathit{left}(0111q_1 1011110) = 1 \,.\, 1 \,.\, 1 \,.\, 0 \,.\, \mathbf{nil}$,
$\mathit{right}(0111q_1 1011110) = q_1 \,.\, 1 \,.\, 0 \,.\, 1 \,.\, 1 \,.\, 1 \,.\, 1 \,.\, 0 \,.\, \mathbf{nil}$.

МОДЕЛИРОВАНИЕ МАШИН ТЬЮРИНГА ЛОГИЧЕСКИМИ ПРОГРАММАМИ

Каждой команде $K = q\ b\ c\ q'\ L$ сопоставим пару программных утверждений $D_1(K),\ D_2(K)$:

$$\begin{array}{lrcl} D_1(K): & P(Z \centerdot X, q \centerdot b \centerdot Y, X', Y') & \leftarrow & P(X, q' \centerdot Z \centerdot c \centerdot Y, X', Y'); \\ D_2(K): & P(\mathbf{nil}, q \centerdot b \centerdot Y, X', Y') & \leftarrow & P(\mathbf{nil}, q' \centerdot a_0 \centerdot c \centerdot Y, X', Y'); \end{array}$$

Каждой команде $K = q\ b\ c\ q'\ R$ сопоставим пару программных утверждений $D_1(K),\ D_2(K)$:

$$\begin{array}{lrcl} D_1(K): & P(X, q \centerdot b \centerdot Z \centerdot Y, X', Y') & \leftarrow & P(c \centerdot X, q' \centerdot Z \centerdot X, X', Y'); \\ D_2(K): & P(X, q \centerdot b \centerdot \mathbf{nil}, X', Y') & \leftarrow & P(c \centerdot X, q' \centerdot a_0 \centerdot \mathbf{nil}, X', Y'); \end{array}$$

МОДЕЛИРОВАНИЕ МАШИН ТЬЮРИНГА ЛОГИЧЕСКИМИ ПРОГРАММАМИ

Кроме того, для каждой пары q, z, где $q \in Q$, $z \in \mathcal{A}$, введем факт $D_{q,z}$:

$$D_{q,x} : P(X, q \blacksquare x \blacksquare Y, X, q \blacksquare x \blacksquare Y) \ \leftarrow;$$

Теперь для каждой машины Тьюринга $\pi = \{K_1, K_2, \ldots, K_N\}$ выделим множество T_π всех пар qx, которые **не являются** началами ни одной из команд программы π...

и построим хорновскую логическую программу

$$\mathcal{P}_\pi = \{D_1(K), D_2(K) \ : \ K \in \pi\} \cup \bigcup_{(qx \in T_\pi)} D_{qx}.$$

Можно надеяться, что логическая программа $\mathcal{P}_\pi$ воспроизводит все вычисления машины Тьюринга π.

МОДЕЛИРОВАНИЕ МАШИН ТЬЮРИНГА ЛОГИЧЕСКИМИ ПРОГРАММАМИ

Пример

Пусть $\mathcal{A} = \{0, 1\}$ и пусть $\pi = \{K_1, K_2, K_3\}$, где
$K_1 = \ q_0\ 0\ 1\ q_1\ R,$
$K_2 = \ q_1\ 0\ 1\ q_0\ L,$
$K_3 = \ q_1\ 1\ 1\ q_1\ R.$

Машина Тьюринга с программой π транслируется в логическую программу $\mathcal{P}_\pi$.

МОДЕЛИРОВАНИЕ МАШИН ТЬЮРИНГА ЛОГИЧЕСКИМИ ПРОГРАММАМИ

Пример

Логическая программа $\mathcal{P}_\pi$:

$$
\begin{array}{lcl}
P(X, q_0 \centerdot 0 \centerdot Z \centerdot Y, X', Y') & \leftarrow & P(1 \centerdot X, q_1 \centerdot Z \centerdot Y, X', Y'); \\
P(X, q_0 \centerdot 0 \centerdot \mathbf{nil}, X', Y') & \leftarrow & P(1 \centerdot X, q_1 \centerdot 0 \centerdot \mathit{nil}, X', Y'); \\
 & & \\
P(Z \centerdot X, q_1 \centerdot 0 \centerdot Y, X', Y') & \leftarrow & P(X, q_0 \centerdot Z \centerdot 1 \centerdot Y, X', Y'); \\
P(\mathbf{nil}, q_1 \centerdot 0 \centerdot Y, X', Y') & \leftarrow & P(\mathbf{nil}, q_0 \centerdot 0 \centerdot 1 \centerdot Y, X', Y'); \\
 & & \\
P(X, q_1 \centerdot 1 \centerdot Z \centerdot Y, X', Y') & \leftarrow & P(1 \centerdot X, q_1 \centerdot Z \centerdot Y, X', Y'); \\
P(X, q_1 \centerdot 1 \centerdot \mathbf{nil}, X', Y') & \leftarrow & P(1 \centerdot X, q_1 \centerdot 0 \centerdot \mathit{nil}, X', Y'); \\
 & & \\
P(X, q_0 \centerdot 1 \centerdot Z, X, q_0 \centerdot 1 \centerdot Z) & \leftarrow & ;
\end{array}
$$

МОДЕЛИРОВАНИЕ МАШИН ТЬЮРИНГА ЛОГИЧЕСКИМИ ПРОГРАММАМИ

Лемма 1

Для любой пары ленточных конфигураций $\alpha, \beta \in Conf$ и команды K отношение перехода $\alpha \rightarrow_K \beta$ выполняется тогда и только тогда, когда запрос
$G_\alpha : ?P(left(\alpha), right(\alpha), X, Y)$
и одно из программных утверждений $D_1(K),\ D_2(K)$
имеют SLD-резольвенту $G_\beta :?P(left(\beta), right(\beta), X, Y)$.

Доказательство

Самостоятельно.

МОДЕЛИРОВАНИЕ МАШИН ТЬЮРИНГА ЛОГИЧЕСКИМИ ПРОГРАММАМИ

Лемма 2

Ленточная конфигурация $\alpha \in Conf$ является заключительной для машины Тьюринга π тогда и только тогда, когда запрос $G_\alpha : ?P(left(\alpha), right(\alpha), X, Y)$
и одно из программных утверждений множества $\bigcup\limits_{(qx \in T_\pi)} D_{qx}$
имеют пустой дизъюнкт $\square$ в качестве SLD-резольвенты.

Доказательство

Самостоятельно.

МОДЕЛИРОВАНИЕ МАШИН ТЬЮРИНГА ЛОГИЧЕСКИМИ ПРОГРАММАМИ

Теорема (о моделировании МТ логическими программами)

Для любой машины Тьюринга π и начальной конфигурации α_0 вычисление

$$\alpha_0 \to_\pi \alpha_1 \to_\pi \alpha_2 \to_\pi \cdots \to_\pi \alpha_N$$

завершается заключительной конфигурацией α_N тогда и только тогда, когда запрос

$$G_{\alpha_0} : ?R(\mathit{left}(\alpha_0), \mathit{right}(\alpha_0), X, Y)$$

к хорновской логической программе $\mathcal{P}_\pi$ имеет успешное вычисление с вычисленным ответом

$$\theta = \{X/\mathit{left}(\alpha_N),\ Y/\mathit{right}(\alpha_N)\}.$$

Доказательство

Следует из лемм 1 и 2. □

МОДЕЛИРОВАНИЕ МАШИН ТЬЮРИНГА ЛОГИЧЕСКИМИ ПРОГРАММАМИ

Таким образом, хорновские логические программы обладают не меньшими вычислительными воможностями, чем машины Тьюринга. Значит, логическое программирование — это универсальная модель вычислений, позволяющая вычислять все эффективно вычислимые функции.

Но это также означает, что логическому программированию присущи все те трудности анализа поведения программ, которые присущи и другим универсальным моделям вычислений. Речь идет об алгоритмически неразрешмых задачах.

Например, интересен вопрос о том, можно ли по заданному запросу G к логической программе $\mathcal{P}$ выяснить, имеет ли этот запрос хотя бы одно успешное вычисление. Тогда не пришлось бесполезно тратить время на построение дерева SLD-резолютивных вычислений.

ТЕОРЕМА ЧЕРЧА

Теорема Тьюринга об алгоритмической неразрешимости проблемы останова

Проблема останова машин Тьюринга алгоритмически неразрешима, т. е. не существует алгоритма (машины Тьюринга), способного вычислить следующую функцию

$$F(x,y) = \begin{cases} 1, & \text{если } x \text{ — это начальная ленточная конфигурация,} \\ & \text{если } y \text{ — это список команд МТ } \pi, \\ & \text{и вычисление } \pi(x) \text{ конечно;} \\ 0, & \text{в противном случае.} \end{cases}$$

Доказательство

Известно каждому первокурснику. □

ТЕОРЕМА ЧЕРЧА

Из теоремы о моделировании машин Тьюринга логическими программами и теоремы об алгоритмической неразрешимости проблемы останова машин Тьюринга получаем несколько важных следствий

Следствие 1.

Не существует алгоритма, способного определить по заданному запросу G к хорновской логической программе $\mathcal{P}$,

- является ли дерево SLD-резолютивных вычислений запроса G конечным;
- содержит ли дерево SLD-резолютивных вычислений запроса G хотя бы одно успешное вычисление;
- является ли заданная подстановка θ вычисленным ответом на запрос G.

ТЕОРЕМА ЧЕРЧА

Следствие 2 (Теорема Черча).

Не существует алгоритма, способного определить по заданной замкнутой формуле логики предикатов φ, является ли эта формула общезначимой, т. е. проблема общезначимости "$\models \varphi$?" алгоритмически неразрешима.

Доказательство

Пусть $G : ?C_1, \ldots, C_m$ произвольный запрос к произвольной логической программе $\mathcal{P} = \{D_1, \ldots, D_N\}$.
Тогда...

ТЕОРЕМА ЧЕРЧА

Доказательство

Запрос G к программе $\mathcal{P}$ имеет хотя бы одно успешное SLD-резолютивное вычисление

$$\Longleftrightarrow \qquad \text{(теоремы корректности и полноты)}$$

Запрос G к программе $\mathcal{P}$ имеет хоть один правильный ответ θ

$$\Longleftrightarrow \qquad \text{(определение правильного ответа)}$$

$$\{D_1, \ldots, D_N\} \models \forall z_1 \ldots \forall z_k (C_1 \& \ldots \& C_m)\theta$$

$$\Longleftrightarrow \qquad \text{(теорема о логическом следовании)}$$

$$\models D_1 \& \ldots \& D_N \ \rightarrow \ \forall z_1 \ldots \forall z_k (C_1 \& \ldots \& C_m)\theta.$$

□

ТЕОРЕМА ЧЕРЧА

Теорема Черча об алгоритмической неразрешимости проблемы общезначимости показывает, что ни одна система автоматического доказательства теорем не может гарантировать решение следующих вопросов для произвольных формул:

- является ли заданная формула φ общезначимой?
- является ли заданная формула φ выполнимой?
- является ли заданная формула φ логическим следствием заданного множества формул Г?

КОНЕЦ ЛЕКЦИИ 16.