Теория вычислительных процессов и структур

Лекция №4. Неразрешимые свойства стандартных схема

Содержание лекции

ИТОГ ЛЕКЦИИ:

Одним из первых шагов в исследовании свойств стандартных схем — установление их распознаваемости. Проблемы пустоты, эквивалентности, свободы и тотальности для стандартных схем неразрешимы, более того, первые три проблемы не являются частично разрешимыми.

Часть 1. Предварительные сведения

Понятие функции

Функцией, отображающей множество X по множество Y (обозначается: $F : X \to Y$), называется множество $F \subseteq X \times Y$ такое, что для любых пар $(x, y) \in F$ и $(x', y') \in F$ из равенства $x = x'$ следует, что $y = y'$.

Множество $\{x | (x, y) \in F\}$ — *область определения функции* F (или множество значений ее аргументов); множество $\{y | (x, y) \in F\}$ — *область значений* функции.

Если область определения функции $F : X \to Y$ служит все множество X, то F — *всюду определенная функция*, в противном случае — *частичная*.

Вычислимая функция

По определению, функция F является (*частично*) *вычислимой*, если существует машина Тьюринга T такая, что $F_T = F$.

Говорят, что для функции F существует (*частичный*) *алгоритм* вычисления ее значения, задаваемый машиной Тьюринга.

Понятие протокола машины Тьюринга

Функционирование машины Тьюринга можно описать с помощью *протокола* работы над заданным начальным словом.

Пусть $\#a_1a_2 \dots a_{k-1}a_k \dots a_n\#$ — слово, возникающее на ленте в процессе работы машины после некоторого шага, в результате которого машина находится в состоянии q, а головка оборевает клетку, содержащую k-ый слева символ слова. Тогда слово

$$\#a_1a_2 \dots a_{k-1}qa_k \dots a_n\#$$

называется *конфигурацией машины Тьюринга*.

Последовательность конфигураций, выписанных в том порядке, в котором они следуют в процессе работы машины, называется *протоколом*.

Массовая алгоритмическая проблема

Конечная цель теории схем программ — автоматизация программирования, в том числе автоматический анализ свойств программ и преобразования программ, осуществляемые с помощью других, специальных программ. Поэтому *нас будут интересовать алгоритмы*, которые могли бы по любой предъявленной программе установить, завершит ли она работу или зациклится, дают ли две программы, исходная и оптимизированная, один и тот же результат, является ли произвольная предъявленная программа (на каком-либо языке) синтаксически правильной и т.д.

Изучая свойства схем программ и алгоритмы их установления мы неизбежно сталкиваемся с понятием *массовой алгоритмической проблемы*, которая формулируется следующим образом:

Необходимо указать алгоритм, который бы определял, обладает ли предъявленный объект из некоторого класса объектов интересующим нас свойством; другими словая, принадлежит ли он множеству M всех объектов, обладающих этим свойством.

Разрешимость и частичная разрешимость

Если существует такой частичный алгоритм, то говорят, что *множество M перечислимо*, а поставленная массовая алгоритмическая проблема *частично разрешима*.

Если этот алгоритм к тому же всюду определен, то *множество M разрешимо* и поставленная проблема также *разрешима*.

Существуют неразрешимые проблемы и даже проблемы, которые не являются частично разрешимыми

Обсуждаемые понятия введем формально!

Характеристические функции

Пусть V — алфавит, а $M \subseteq V^*$ — некоторое множество слов в этом алфавите. Тогда

Характеристической функцией множества M называется предикат $F_M : V^* \to \{0,1\}$, всюду определенный на V^*:

$$F_M(\alpha) = \begin{cases} 1, & \text{если } \alpha \in M, \\ 0, & \text{если } \alpha \notin M. \end{cases}$$

Частичная характеристическая функция множества M — это функция $H_M : V^* \to \{1\}$, определеная только для слов из M и имеющая вид $H_M(\alpha) = 1$ для всех $\alpha \in M$.

Множество M называется *разрешимым*, если его характеристическая функция вычислима.

Множество M *перечислимо*, если его частичная характеристическая функция вычислима.

Разрешимость и перечислимость

Разрешимость множества M означает, что существует всегда останавливающаяся машина Тьюринга T_{F_M}, позволяющая для любого слова в алфавите V через конечное число шагов установить, принадлежит ли это слово множеству M или нет.

Перечислимость множества M означает, что существует машина Тьюринга T_{H_M}, которая останавливается в том и только в том случае, если предъявленное слово принадлежит множеству M. Машина T_{H_M} не позволяет точно установить, принадлежит заданное слово множеству M или нет, так как отсутствие ответа к некоторому времени не несет никакой информации о том, появится он позже или нет. Однако, если машина T_{H_M} остановилась, то мы знаем, что предъявленное слово принадлежит M.

Основные теоремы машин Тьюринга

Теорема 1

Множество $M \subseteq V^*$ *разрешимо тогда и только тогда, когда* M *и его дополнение* $\bar{M} = V^* \setminus M$ *перечислимы.*

Теорема 2

Проблема остановки машин Тьюринга неразрешима

Теорема 3

Проблема пустой ленты неразрешима (остановка машины на пустой ленте)

Теорема 4

Проблема зацикливания машин Тьюринга не является частично разрешимой.

Определение автомата

Двухголовочный автомат $A = (V, Q, R, q_0, \#, I)$ имеет одну ленту и две головки, которые могут независимо перемещаться вдоль ленты в одном направлении. Множество состояний $Q = Q_1 \cup Q_2$ разбито на два непересекающихся подномжества; в состояниях Q_1 *активна первая головка*, а в состояниях Q_2 *активная вторая головка*. V — алфавит символов на ленте. R — множество заключительных состояний, $R \subseteq Q$, $q_0 \in Q$ — начальное состояние, $\#$ — пустой символ, а I — программа автомата A, т.е. последовательность команда вида $q\ a \to q'$, где $q, q' \in Q$, $a \in V$, причем для любой пары (q, a) существует единственная команда, начинающаяся этими символами.

Пример двухголовочного автомата...

Приведем пример двухголовочного автомата, который проверяет равенство двух последовательно записанных слов а алфавите $V = \{0, 1\}$. Признаком окончания каждого из слов будет символ $*$, не входящий в V. Автомат должен допускать только слова вида $\alpha * \alpha *$, где $\alpha \in V^*$. Зададим

$A_= = (V \cup \{*\}, Q_1 \cup Q_2, R, q_0^1, \#, I)$, где $Q_1 = \{q_0^1, q_1^1, q_2^1, q_7^1\}$ — множество состояний, в которых активна первая головка; $Q_2 = \{q_3^2, q_4^2, q_5^2, q_6^2\}$ — множество состояний, в которых активна вторая головка; $R = \{q_7^1\}$ — множество, содержащее единственное заключительное состояние.

Граф автомата показан на рисунке слайда 15. На этом рисунке вместо многих "параллельных" дуг с разными пометками, нарисована одна дуга со всеми этими пометками.

Пример двухголовочного автомата

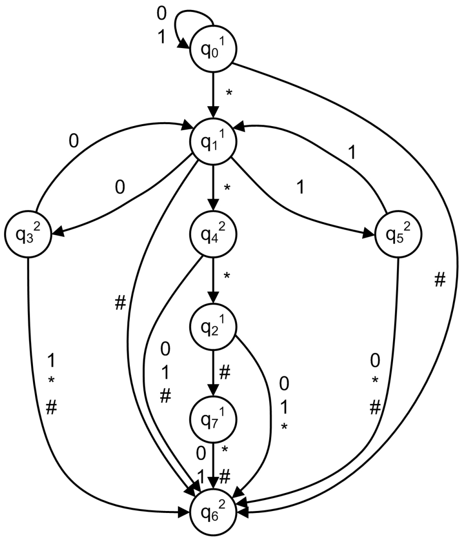


Рис.: Двухголовочный автомат $A_=$

Связь двухголовочных автоматов с машинами Тьюринга

Говорят, что *двухголовочный автомат моделирует работу машины Тьюринга над некоторым начальным словом*, если автомат допускает единственное слово — конечный протокол работы машины над ним.

Лемма 5

Существует алгоритм, который для любой машины Тьюринга и для любого начального слова строит двухголовочный автомат, моделирующий ее работу над этим словом.

Доказательство.

Построение автомата основано на сравнении двух соседних конфигуарций машины Тьюринга. Если конфигурации одинаковы, значит машина зациклилась и слово отверагается. Если есть различия, которые соответствуют какой-то команде машины Тьюринга, то происходит проверка следующей конфигурации и т.д. В противном случае слово также отвергается. □

Теоремы о двухголовочных автоматах

Теорема 6

Проблема пустоты двухголовочных автоматов не является частично разрешимой.

Теорема 7

Проблема эквивалентности двухголовочных автоматов не является частично разрешимой.

Часть 2. Двоичные двухголовочные автоматы

Введение

Для доказательства неразрешимости пустоты и эквивалентности схем покажем, что стандартные схемы могут моделировать (определенным способом) двухголовочные автоматы, что позволит свести проблему пустоты этих автоматов к проблеме пустоты схем.

Такое моделирование можно осуществить более простым способом, если использовать специальный класс двухголовочных автоматов, а именно класс двоичных двухголовочных автоматов (ДДА), работающих со словами над алфавитом $\{0, 1\}$.

Построение ДДА...

Лемма 8

Существует алгоритм преобразования двухголовочных автоматов в двоичные двухголовочные автоматы, сохраняющий пустоту автоматов (построенный двоичный автомат A_b пуст тогда и только тогда, когда пуст исходный автомат A).

Доказательство.

Пусть двухголовочный автомат A над алфавитом $V = \{a_1, a_2, \dots, a_n\}$ имеет множество состояний $Q_A = \{q_1^{i_1}, q_2^{i_2}, \dots, q_k^{i_k}\}$, где верхний индекса равен 1 (активна головка 1) или 2 (активна головка 2).

Шаг 1: Кодировка символов из V^*

код$(\#) = 0$,
код$(a_i) = \underbrace{11\dots1}_{i}0 \ (i = 1, \dots, n)$
код$(\alpha a_i) =$ код(α)код(a_i).
Таким образом, два нуля подряд на ленте автомата A_b приводит к его остановке.

Построение ДДА...

Доказательство (продолжение).

Шаг 2: Структурная замена

1. Каждый фрагмент графа автомата A, соответствующий некоторому состоянию q_j^i из Q_A (показан на слайде 22 литера "а") заменяется фрагментом графа автомата A_b (показан на слайде 22 литера "б").
2. К графу добавляются фрагменты, показанные под литерами "в" и "г".

Таким образом, множество состояний автомата A_b включает:

1. все старые состояния из Q_A,
2. для каждого старого состояния q_j^i n новых состояний, где n — число символов в V,
3. два новых состояния r_1^1 и r_2^1.

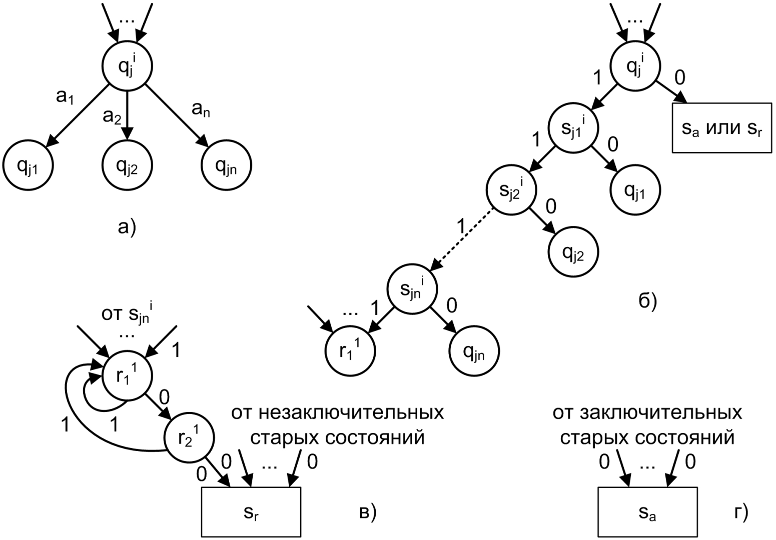


Рис.: Фрагменты двухголовочного автомата A и двоичного автомата A_b: а — фрагмент автомата A; б — соответствующий фрагмент автомата A_b; в — отвергающий фрагмент; г — допускающий фрагмент

Построение ДДА...

Доказательство (продолжение).

Заключительными состояниями автомата A_b являются все заключительные состояния A и только они. Вершина s_a (останов допускающий) и s_r (останов отвергающий) носят вспомогательный характер в графе A_b. Они отмечают тот факт, что автомат прочитал два нуля подряд и остановился в заключительном состоянии (случа s_a) или в незаключительном состоянии (случай s_r). $\square$

Пример ДДА

Двухголовочный автомат A, допускающий только те слова в алфавите $V = \{a, b, c\}$, в которых символ a встречается не меньшее число раз, чем символы b и c, вместе взятые, показан на слайде 25 под литерой "а" (заключительное состояние — q_0^1). Под литерой "б" изображен двоичный автомат, построенный по автомату A (10 — код символа a, 110 — код символа b, 1110 — код символа c).

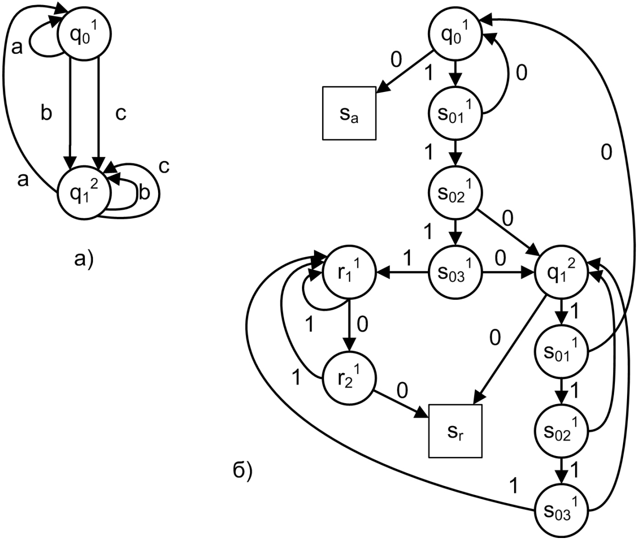


Рис.: Пример ДДА

Теоремы о ДДА

Лемма 9

Проблема пустоты двоичных двухголовочных автоматов не является частично разрешимой.

Часть 3. Моделирование двоичного автомата стандартной схемой

Исходные данные

Зафиксируем класс $\mathcal{S}_1$ стандартных схем в следующем базисе $\mathcal{B}_1$:

1. $\{x_1, x_2\}$ — множество переменных;
2. $\{a, f^{(1)}\}$ — множество функциональных символов;
3. $\{p^{(1)}\}$ — множество предикатных символов;
4. $\{x_1 := f(x_1), x_2 := f(x_2), x_1 := a, x_2 := a, p(x_1), p(x_2), \text{стоп}(x_1, x_2)\}$ — множество операторов.
5. Также предполагается, что все схемы класса $\mathcal{S}_1$ начинаются так
$$\{\text{старт}, x_1 := a, x_2 := a \dots\}$$
все операторы засылки констант встречаются только по одному разу, причем в самом начале.

Из п. 4) и 5) следует, что при любой свободной интерпретации I базиса $\mathcal{B}_1$ в любой программе (S, I), где $S \in \mathcal{S}_1$, переменные x_1 и x_2 могут принимать значения только из следующего множества термов:

$$\{a, fa, ffa, \dots, f^n a, \dots\}$$

Согласованность двоичного слова

Двоичное слово $b_1 b_2 \dots b_n$ согласовано со свбодной интерпретацией базиса $\mathcal{B}_1$, если для любого i $(1 \leq i \leq n)$, $I(p)('f^i a') = b_i$, где p — единственный предикатный символ базиса $\mathcal{B}_1$.

Например
а) 101010100 согласовано с любой свободной интерпретацией I такой, что для всех $i \leq 9$:

$$I(p)('f^i a') = \begin{cases} 1, & \text{если } i \text{ нечетно и меньше 9,} \\ 0, & \text{если } i \text{ четно или равно 9} \end{cases}$$

б) (обратно) свободная интерпретация I такая, что для всех i

$$I(p)('f^i a') = 0,$$

согласована с любым словом, не содержащим 1.

Связь ДДА со схемами

Для того чтобы свести проблему пустоты двоичных двухголовочных автоматов к проблеме пустоты стандартных схем из класса $\mathcal{S}_1$, покажем, что для любого двоичного автомата A можно построить схему S из $\mathcal{S}_1$, которая моделирует автомат A в следующем смысле:

Если на ленту автомата A подано произвольное двоичное слово α, то программа (S, I), где I — любая свободная интерпретация базиса $\mathcal{B}_1$, согласованная с α, останавливается в том и только в том случае, когда автомат допускает слово α.

Лемма 10

Двоичный двухголовочный автомат пуст в том и только в том случае, если пуста моделирующая его стандартная схема.

Построение схемы

Лемма 11

Для любого двоичного двухголовочного автомата можно построить моделирующую его схема из класса $\mathcal{S}_1$.

Доказательство.

Схема строится в два этапа:

Этап 1: Подготовка фрагментов схемы (графически)

Фрагмент графа (схемы) — это его подграф (подмножество вершин вместе с дугами, соединяющими эти вершины).

Пусть граф двоичного двухголовочного автомата A включает вершины состояния $\{q_0^{i_0}, q_1^{i_1}, \dots, q_j^{i_j}, \dots, q_n^{i_n}\}$ и две вспомогательные вершины — останов допускающий (s_a) и останов отвергающий (s_r).

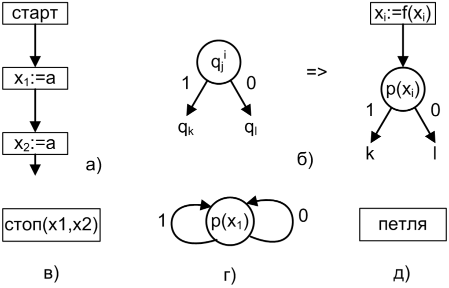
Присвоим им здесь имена q_{n+1} и q_{n+2} соответственно.

Схема *собирается* из $n+4$ фрагментов

$\mathcal{F}_{\text{нач}}, \mathcal{F}_0, \mathcal{F}_1, \dots, \mathcal{F}_j, \dots, \mathcal{F}_n, \mathcal{F}_{n+1}, \mathcal{F}_{n+2}$:

Доказательство (продолжение).

- Фрагмент $\mathcal{F}_{\text{нач}}$ — общий для всех схем из класса $\mathcal{S}_1$ начальный фрагмент (см. на слайде 32 литера "а");
- Фрагменты $\mathcal{F}_j$, $0 \le j \le n$, соответствуют состояниям $q_i^{i_j}$ и строятся по фрагментам графа автомата, связанным с $q_i^{i_j}$ так, как показано на слайде 32 литера "б".
- Фрагмент $\mathcal{F}_{n+1}$ соответствует вершине q_{n+1} (литера "в");
- Фрагмент $\mathcal{F}_{n+2}$ соответствует вершине q_{n+2} (литера "г"), что эквивалентно оператору "петля" (литера "г").



Доказательство (продолжение).

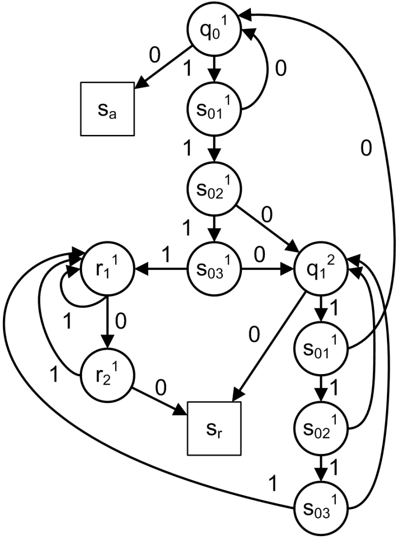
Этап 2: Сборка схемы

Правило сборки: выходная дуга с пометкой j фрагментов $\mathcal{F}_{\text{нач}}, \mathcal{F}_0, \ldots, \mathcal{F}_n$ заводится на начальную вершину фрагмента $\mathcal{F}_j$, где $0 \leq j \leq n+2$.

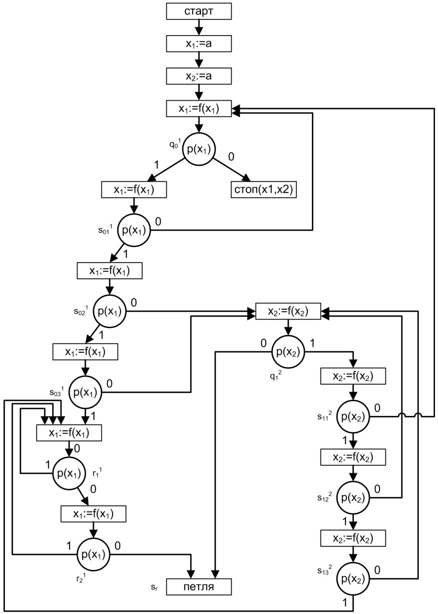
После этого трубется перенумеровать числами-метками подходящим образом все вершины поученного графа, и стандартная схема построена. $\square$

Пример модели...

На слайде 35 приведена стандартная схема, моделирующая двоичнй двухголовочный автомат, изображенный ниже



Пример модели



Часть 4. Теоремы о неразрешимых свойствах стандартных схем

Проблемы пустоты и эквивалентности

Теорема 12

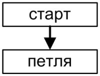
Проблема пустоты стандартных схем не является частично разрешимой.

Теорема 13

Проблема функциональной эквивалентности стандартных схем не является частично разрешимой

Доказательство.

Если бы эта проблема была частично разрешима, то существовал бы частичный алгоритм распознавания пустоты стандартных схем, который состоял бы в установлении эквивалентности заданной схемы заведомо пустой схемы, например как на рисунке ниже. □



Теорема 14

Проблема свбоды стандартных схем не является частично разрешимой

Проблема тотальности

Теорема 15

Проблема тотальности стандартных схем частично разрешима.

Доказательство.

Пусть S — стандартная схема, C — множество всех префиксов допустимых цепочек схемы S. На множестве C введем отношение "$<$": $c_1 < c_2$, если $c_1 \in C$, $c_2 \in C$, $c_1 \neq c_2$ и c_1 является максимальным префиксом c_2, т.е для любого $c_3 \in C$ и $c_3 \neq c_1$ $c_3 \neq c_1$, не может быть, что $c_1 < c_3$ и одновременно $c_3 < c_2$.

Можно определить граф G, описывающий это отношение. Вершинами его являются элементы C, и из c_1 ведет дуга к c_2, если и только если $c_1 < c_2$. Граф G является деревом. Корнем является вершина с префиксом из одной метки 0. Из вершины $c = (0, 1, \dots, k)$ исходит одна дуга, если k — метка преобразователя, одна или две дуги, если k — метка распознавателя, и эта вершина является листом, если k — метка заключительного оператора. Таким образом, листьями дерева G служат допустимые цепочки схемы S. Отметим также, что в C длина префикса c_1 ровно на единицу меньше c_2, если $c_1 < c_2$.

Доказательство (продолжение).

Если G — конечное дерево (кол-во вершин конечно) то все допустимые цепочки конечны и схема S тотальна. В противном случае, схема S не тотальна. □