Листок 1

Модели вычислений и почему они не имеют значения ☺

Детерминированная машина Тьюринга

Детерминированной машиной Тьюринга или просто машиной Тьюринга без уточнения (ДМТ или просто МТ) в различных источниках называются различные объекты. При этом отличия в определениях "незначительны" в любых дальнейших применениях определения. Т.е, теоремы о существовании, возможности моделирования, класс сложности остаются (почти) таковыми без изменения при изменении определения.

В общем виде ДМТ – это тройка (Q, Γ, δ). Здесь

– Q – конечное множество состояний МТ; мы будем предполагать (также это можно включить в определение, сделав из тройки упорядоченное множество большего размера – так и делают во многих источниках), что Q содержит четыре выделенных состояния: q_0 – начальное состояние, q_{accept} – принимающее состояние, q_{reject} – отвергающее состояние, q_{halt} – завершающее состояние.

– Γ – конечное множество, называемое алфавитом МТ; иногда в нём выделяют подмножество, называемое входным алфавитом. Мы будем предполагать, что Γ содержит два выделенных символа: $\triangleright$ для обозначения "левой границы для ленты", Λ в качестве "пустого" символа. Кроме того, будем стандартно предполагать, что 0 и 1 также лежат в алфавите МТ.

– δ – функция перехода, $\delta : Q \times \Gamma \to Q \times \Gamma \times \{-1, +1\}$. Типично на эту функцию накладываются дополнительные ограничения для удобства описания МТ: функция может быть частичной (не определена на некоторых входах), в частности $\delta(q_{halt}, +1)$ (и множество подобных же значений) обычно не определено, поскольку q_{halt} по своему смыслу является завершающим состоянием, из которого МТ уже никуда не переходит. Кроме того, если (q, a) никогда не достигается при вычислениях, её часто опускают и т.д. Положим также $\delta(\cdot, \triangleright) = (\cdot, \triangleright, +1)$, здесь точкой обозначены произвольные значения аргументов – т.е, при считывании левой границы МТ оставляет символ на месте и перемещается вправо.

Неформально: машина Тьюринга (МТ) состоит из бесконечной, разбитой на клетки, ленты и головки. Головка представляет собой конечный автомат. Лента бесконечна в одну сторону и используется для хранения информации, в каждой клетке может быть записан символ или ничего не записано (ничего – это Λ). В самой левой клетке ленты написан символ $\triangleright$, который в соответствии с дополнениями к определению остаётся там на протяжении работы МТ.

Головка — активная часть МТ, на каждом шаге она размещается над одной из клеток ленты и видит её содержимое. Кроме того, на каждом шаге головка находится в одном из состояний. На каждом шаге МТ делает три элементарных действия в зависимости от состояния головки и видимого символа:

— в клетку под головкой записывается некоторый символ,

— состояние головки меняется на некоторое состояние,

— головка сдвигается влево (-1 в качестве третьего выходного аргумента) или вправо ($+1$) на одну клетку.

Вариации МТ в различных источниках:

– множество завершающих состояний не состоит из трёх элементов, а является произвольным подмножеством Q;

– завершающее состояние единственно, при этом распознавание языка делается с помощью ленты, а не состояний;

– лента не односторонняя, а двусторонняя, при этом символа $\triangleright$ не вводится;

– алфавит МТ равен $\{0, 1, \Lambda\}$ (и треугольник в случае односторонней ленты);

– допустимо стоять на месте после смены состояний (множество переходов равно $\{-1, 0, +1\}$).

Все эти вариации "не имеют значения" (модели представляются друг через друга лишь с полиномиальным замедлением).

Многоленточность

Машина Тьюринга является в некотором смысле "естественной" моделью вычислений и определением алгоритма – в её определении напрямую моделируется последовательное выполнение инструкций над конечным явно заданным объектом. Одним из расширений МТ является "распараллеливание", приводящее к понятию многоленточной МТ.

Многоленточной машиной Тьюринга называется тройка (Q, Γ, δ) вместе с натуральным числом $k \geqslant 2$. Здесь множество состояний и алфавит те же самые, что и ранее, а функция перехода имеет сигнатуру $\delta\colon Q \times \Gamma^k \to Q \times \Gamma^k \times \{-1, +1\}^k$.

Число k определяет количество лент у машины, часто при $k \geqslant 3$ одну из лент выделяют как входную – с неё МТ может только читать, но не писать на неё, ещё одну выделяют как выходную – то, что написано на ней, является результатом вычислений МТ, оставшиеся ленты рабочие и используются для собственно проведения вычислений.

Как и у одноленточной, у многоленточной МТ есть вариации определния, "неважные" с точки зрения вычислительной сложности.

Используем мы многоленточную или одноленточную МТ "важно" с одной точки зрения и "неважно" с другой (см. задачи).

Вычисления на МТ

Конкретный способ изменения состояний, букв на ленте и движения головки определяется программой МТ. МТ работает тактами, которые выполняются один за другим. До выполнения программы на ленту написано входное слово (по одной букве в клетке, в случае односторонней ленты в клетке номер нуль записан маркер начала строки, далее написано слово) и, если не указано обратное, головка МТ находится над нулевой клеткой (где находится $\triangleright$) в состоянии q_0. После этого начинается выполнение программы. В результате МТ должна совершить необходимые действия (подразумеваемые конкретной задачей).

В нашем случае завершающих состояний у МТ три штуки, два из них q_{accept} и q_{reject} используются для распознавания языка. МТ распознает язык L, если на любом слове $x \in L$ МТ останавливается в состоянии q_{accept} (неважно при этом, где на ленте находится головка или что написано на ленте), а на любом слове $x \notin L$ МТ останавливается в состоянии q_{reject}.

Завершающее состояние q_{halt} используется для вычислений. МТ вычисляет функцию f, если для любого x:

– если f определена на x, то МТ останавливается на слове x в состоянии q_{halt} (т.е. после некоторого конечного числа шагов её состояние становится равно q_{halt}, при этом q_{accept} и q_{reject} никогда не достигаются, чтобы не мешать), а на ленте при этом написано $f(x)$;

– если f не определена на x, то МТ не останавливается на слове x (т.е. ни одно из состояний q_{accept}, q_{reject}, q_{halt} не достигается никогда).

МТ, распознающая язык, может рассматриваться как МТ, вычисляющая характеристическую функцию этого языка.

Тезис Тьюринга: для любой вычислимой функции существует МТ, вычисляющая эту функцию. Т.е, МТ может реализовать любой алгоритм.

МТ можно кодировать – существует заранее выбранный алфавит (обычно бинарный) и существует вычислимый способ по данной МТ получить её кодировку и по данной строке над алфавитом получить соответствующую МТ (здесь нужно как-то разобраться со сборкой мусора, см. задачи).

Существует универсальная машина Тьюринга $\mathcal{U}$: она принимает два аргумента α и x (т.е. два входных слова написаны на ленте через разделитель), результатом её вычисления будет то же самое (в том числе неостанов), что является результатом вычисления машины с кодировкой α на входе x. Писать будем $\mathcal{U}(\langle M \rangle, x) = M(x)$.

Время работы

Фиксируйте ваше любимое определение МТ.

Пусть МТ M вычисляет функцию f (или распознаёт какой-то язык), пусть $T : \mathbb{N} \to \mathbb{N}$. Скажем, что M работает за время $T(n)$, если для любого $x \in \{0,1\}^*$ машина M завершает работу не более, чем за $T(|x|)$ шагов. Опционально сюда нужно добавить утверждение о минимальности такой функции T – для каждого n существует $x \in \{0,1\}^n$, что M останавливается на x за $T(n)$ шагов.

Скажем, что M работает за время $O(T(n))$, если существует функция $T'(n) = O(T(n))$ такая, что M работает за время $T'(n)$.

Для языков (или, что то же самое, функций $f : \mathbb{N} \to \{0,1\}$) вводится следующее важное определение.

Пусть $T : \mathbb{N} \to \mathbb{N}$. Говорят, что язык $L \in \mathrm{DTIME}(T(n))$, если существует машина Тьюринга M, которая распознает язык L и работает за время $O(T(n))$.

В теоремах о связи различных вариаций МТ, об их времени работы и т.п. в качестве функции, ограничивающей время работы МТ, не может быть взята любая функция. Прежде всего, она должна быть вычислимой (иначе не очень понятно, о каком времени работы идёт речь). Аккуратное определение "хорошей функции" выглядит так. Функция $T : \mathbb{N} \to \mathbb{N}$ называется конструируемой по времени, если для любого n выполняется $T(n) \geqslant n$ и существует машина Тьюринга, которая на входе x печатает на ленте бинарное представление $T(|x|)$ и работает за время $O(T(n))$.

В дальнейшем во всех теоремах мы предполагаем время работы МТ функцией, конструируемой по времени. Все "приличные" функции являются таковыми.

RAM

Машина с произвольным доступом к памяти (random access machine или RAM – не путать с планкой памяти в компьютере) – ещё одна вычислительная модель. Это общий вид регистровой машины с возможностью доступа к регистрам по их адресу. Слово random здесь не несёт никакого оттенка вероятности или случайности.

Память RAM состоит из бесконечного числа регистров, в каждом регистре может храниться произвольное натуральное число. Программа RAM состоит из последовательности команд. Каждая команда – это команда из заранее оговоренного списка команд, который типично включает копирование чисел между регистрами, битовые сдвиги, сложение, вычитание, переход к другой команде в списке по её номеру, условные переходы if-else.

Каждая команда выполняется за 1 такт времени работы. Подобная машина, казалось бы, значительно расширяет возможности вычислений и уменьшает время работы – она умеет складывать любые числа за 1 такт времени. Оказывается, однако, что такая машина может быть моделирована МТ с полиномиальным замедлением.

Задачи

Задача 1.1:

Конфигурацией (многоленточной) машины Тьюринга M назовём набор $\langle x_1, x_2, \dots x_k, q, y_1, \dots, y_k \rangle$. В нём x_i – строка, находящаяся на ленте номер i (нестрого) левее головки, y_i – строка правее головки, а q – текущее состояние. Определите формально процесс вычисления на МТ в терминах последовательности конфигураций.

Задача 1.2: Эффективная кодировка

Чтобы не рассматривать различные входные алфавиты и алфавиты МТ, а также чтобы уметь задавать вопросы о произвольных конструктивно заданных объектах, а не только строках, достаточно научиться кодировать все данные в бинарном алфавите.

Предъявите какую-нибудь эффективную (вычислимую и при этом желательно быструю) процедуру получения строки над алфавитом $\{0, 1\}$ по объекту и наоборот получения объекта по строке для следующим множеств объектов. Заметьте, что декодирование (получение объекта по строке) может быть сопряжено с технической трудностью: не все строки при кодировании становятся образом какого-то объекта – среди строк появляется “мусор”.

Объекты:

а) пары бинарных строк с разделителями;

б) натуральные числа;

в) булевы формулы;

г) рациональные числа (можно ли вещественные?);

д) матрицы;

е) графы;

ё) машины Тьюринга.

Задача 1.3: Это очень просто и тезис Тьюринга не при чём

Докажите, что вычислимость на МТ не меняется, если модифицировать определение МТ:

а) завершающее состояние единственно, при этом распознавание языка делается с помощью ленты, а не состояний;

б) лента не односторонняя, а двусторонняя;

в) алфавит МТ равен $\{0, 1, \Lambda\}$ (и треугольник в случае односторонней ленты);

г) допустимы только движение головки влево или вправо (-1 или 1);

д) МТ многоленточная.

Задача 1.4: Многоленточность важна

1) Покажите, что язык $\mathrm{PAL} = \{w \mid w \in \{0,1\}^*, w - \text{палиндром}\}$ распознается за $T(n) = O(n)$ на многоленточной МТ.

2) Докажите, что любая одноленточная МТ, распознающая PAL, работает за время $T(n) = \Omega(n^2)$

Задача 1.5: Пусть COPY = $\{x\#x \mid x \in \{0,1\}^*\}$.

1) Докажите, что COPY распознаётся на многоленточной МТ за $T(n) = O(n)$.

2) Докажите, что COPY распознаётся на одноленточной МТ за $T(n) = O(n^2)$.

3) Докажите, что любая одноленточная МТ, распознающая COPY, работает за время $T(n) = \Omega\left(n^2\right)$

Задача 1.6: Универсальная машина Тьюринга

Докажите, что существует универсальная машина Тьюринга $\mathcal{U}$:

– многоленточная, с односторонними лентами, алфавит машины есть $\{0, 1, \triangleright, \Lambda\}$;

– она принимает на вход описание любой МТ (одно/многоленточной и с любым алфавитом);

– $\mathcal{U}(\langle M\rangle, x) = M(x)$ для любого слова x

– если M работает за время $T(n)$, то $\mathcal{U}$ работает за время $O(T^2)$, константа при этом зависит только от M, но не от входа x.

Задача 1.7:

Пусть M работает за время $T(n)$ и существует такое n_0, что $T(n_0) < n_0 + 1$. Докажите, что:

1) M не читает символ номер $n_0 + 1$ ни у какого входа;

2) M работает за время $O(1)$;

3) M распознаёт регулярный язык.

Задача 1.8:

Докажите, что $\forall d \in \mathbb{N}_0$ выполняется $\mathrm{DTIME}(O(n^d)) \neq \mathrm{DTIME}(O(n^{d+1}))$

Задача 1.9:

Постройте одноленточную МТ, которая переводит вход 1^n в выход 1^{2n}, т.е. удваивает слово, заданное в унарном алфавите. Время работы $O(n \log n)$.

Задача 1.10:

При описании МТ мы требуем конечности от алфавита и множества состояний. Что, если не накладывать такого ограничения? Опишите формально МТ с бесконечным множеством состояний. Эквивалентна ли она (по вычислительной силе) стандартной МТ?

Задача 1.11: Пусть THENOTREGULAR = $\{0^n1^n \mid n \in \mathbb{N}\}$.

1) Докажите, что THENOTREGULAR распознаётся на многоленточной МТ за $T(n) = O(n)$.

2) Докажите, что THENOTREGULAR распознаётся на одноленточной МТ за $T(n) = O(n \log n)$.

3) Докажите, что любая одноленточная МТ, распознающая THENOTREGULAR, работает за время $T(n) = \Omega\left(n \log n\right)$