Алгоритмическая сложность вычисления рекуррент

Артём Парфенов

Международная лаборатория теоретической информатики НИУ ВШЭ

Научная школа МЛ ТИ в Вороново,
9 мая 2024 г.

Содержание

Формулировка рекурренты

Рекуррентой общего вида назовём функцию $P : \mathbb{N} \to \Sigma, |\Sigma| = k < \infty$

$$P(x) = f(P(x - a_1), P(x - a_2), \ldots, P(x - a_d)), \text{ где}$$

- f задана таблицей значений размера k^d,
- a_i заданы в двоичной/унарной записи.

Замечание

Пусть $n = max_i(a_i)$. При этом, если задать $P(0), \ldots, P(n-1)$, функция $P(x)$ будет определена однозначно.

Задача с короткими сдвигами

Формулировка задачи

Вход: начальные данные заданы массивом значений, a_i заданы в унарной записи, x — аргумент функции — задан в бинарной записи, $\beta \subseteq \Sigma$ — множество финальных состояний.
Вопрос: Верно ли, что $P(x) \in \beta$?

Задача с короткими сдвигами

Формулировка задачи

Вход: начальные данные заданы массивом значений, a_i заданы в унарной записи, x — аргумент функции — задан в бинарной записи, $\beta \subseteq \Sigma$ — множество финальных состояний.
Вопрос: Верно ли, что $P(x) \in \beta$?

Насколько трудна такая задача?

Машина Тьюринга

Машина Тьюринга M это кортеж $(Q, \Gamma, \sigma, q_s, q_f)$.
Состоит она из:

1. бесконечной в две стороны ленты, в ячейках которой могут быть записаны символы конечного алфавита Γ
2. головки, которая может двигаться вдоль ленты, обозревая в каждый данный момент времени одну из ячеек
3. оперативной памяти, которая имеет конечный размер (другими словами, состояние оперативной памяти — это элемент некоторого конечного множества Q)
4. Таблицы переходов $\sigma : \Gamma \times Q \to \Gamma \times Q \times \{-1, 0, 1\}$.

Функция перехода

Функция перехода задана на конечном множестве, потому можно задать ее таблицей значений. $\sigma(a, q) = (a', q', s)$ означает, что если головка МТ находит- ся над ячейкой, содержащей символ a, а состояние МТ равно q, то на очередном такте работы МТ записывает в текущую ячейку символ a', изменяет состояние на q' и сдвигает головку на s ячеек (вправо или влево в зависимости от знака).

УМТ

Определение

Универсальной машиной Тьюринга U называют МТ, которая может по описанию произвольной машины M и входу I может симулировать работу M на входе I.

Существование этой машины можно проверить в качестве упражнения 🙂

Время

Определение

Пусть $f : \{0,1\}^* \to \{0,1\}^*$, $T : \mathbb{N} \to \mathbb{N}$. Тогда МТ M вычисляет функцию f за время $T(n)$, если $\forall x \in \{0,1\}^*$ M, начав на входе x, останавливается после не более, чем $T(|x|)$ тактов работы.

DTIME

Пусть $T : \mathbb{N} \to \mathbb{N}$. Определим $DTIME(T(n))$ как множество всех булевых функций, вычислимых за время $O(T(n))$.

P и NP

Определение

$\mathrm{P} = \bigcup_{c \geqslant 1} \mathrm{DTIME}(n^c)$

Про класс NP можно думать как про класс задач, решение которых можно проверить эффективно. (но совсем не факт, что можно при этом решить эффективно)

NP

Язык $L \subseteq \{0,1\}^*$ лежит в NP если существует полином $p : \mathbb{N} \to \mathbb{N}$ и полиномиальная МТ такие что $\forall x \in \{0,1\}^*$

$$x \in L \iff \exists u \in \{0,1\}^{p(|x|)} : M(x,u) = 1$$

Такой u называют сертификатом.

PSPACE

Определение

Пусть $S : \mathbb{N} \to \mathbb{N}$, $L \subseteq \{0,1\}^*$. Тогда язык $L \in \mathrm{SPACE}(S(n))$ если найдется МТ M решающая L, такая что на любом входе $x \in \{0,1\}^*$ общее количество не пустых ячеек ленты во время работы ведет себя как $O(S(|x|))$.

PSPACE

Определение

Пусть $S : \mathbb{N} \to \mathbb{N}$, $L \subseteq \{0,1\}^*$. Тогда язык $L \in \mathrm{SPACE}(S(n))$ если найдется МТ M решающая L, такая что на любом входе $x \in \{0,1\}^*$ общее количество не пустых ячеек ленты во время работы ведет себя как $O(S(|x|))$.

Упражнение

$\mathrm{DTIME}(S(n)) \subseteq \mathrm{SPACE}(S(n))$

Полнота и сводимость

m–сводимость

Пусть два языка $A, B \subseteq \{0,1\}^*$. $A \leqslant_P B$ если существует полиномиально вычислимая функция $f : \{0,1\}^* \to \{0,1\}^*$, такая что $\forall x \in \{0,1\}^* : x \in A \iff f(x) \in B$.

Полнота и сводимость

m–сводимость

Пусть два языка $A, B \subseteq \{0,1\}^*$. $A \leqslant_P B$ если существует полиномиально вычислимая функция $f : \{0,1\}^* \to \{0,1\}^*$, такая что $\forall x \in \{0,1\}^* : x \in A \iff f(x) \in B$.

Трудность

Язык L PSPACE–трудный, если $\forall A \in \mathrm{PSPACE} : A \leqslant_P L$.

Полнота и сводимость

m–сводимость

Пусть два языка $A, B \subseteq \{0,1\}^*$. $A \leqslant_P B$ если существует полиномиально вычислимая функция $f : \{0,1\}^* \to \{0,1\}^*$, такая что $\forall x \in \{0,1\}^* : x \in A \iff f(x) \in B$.

Трудность

Язык L PSPACE–трудный, если $\forall A \in \mathrm{PSPACE} : A \leqslant_P L$.

Полнота

Язык L PSPACE–полный, если $L \in \mathrm{PSPACE}$ и L — PSPACE–трудный.

Такие разные остановки

Проблема остановки

Функция двух параметров M, x равна единице если и только если M останавливается на входе x за конечное число тактов.

Такие разные остановки

Проблема остановки

Функция двух параметров M, x равна единице если и только если M останавливается на входе x за конечное число тактов.

Ограниченная проблема остановки

Пусть нам теперь дана МТ M, вход x и размер зоны 1^n. Верно ли, что M на входе x работает на памяти n.

Число конфигураций ДМТ

Лемма

Пусть M — детерминированная МТ, которая имеет q состояний и работает на памяти размера L в алфавите Γ. Тогда число различных конфигураций работы M равно $q \cdot L \cdot |\Gamma|^L$

Число конфигураций ДМТ

Лемма

Пусть M — детерминированная МТ, которая имеет q состояний и работает на памяти размера L в алфавите Γ. Тогда число различных конфигураций работы M равно $q \cdot L \cdot |\Gamma|^L$

Доказательство.

Заметим, что конфигурация состоит из состояния, положения головки и ленты. У нас есть q возможных состояний, на ленту можно смотреть как на слово длины L в алфавите Γ — то есть возможных состояний ленты $|\Gamma|^L$. Наконец, головка может быть над любой ячейкой ленты, то есть возможных позиций всего L. Итого, возможных конфигураций $q \cdot L \cdot |\Gamma|^L$. □

Рекуррента в конечном алфавите с короткими сдвигами

Теорема (М.Н. Вялый — П., 2024)

Задача вычисления рекурренты с короткими в конечном алфавите сдвигами PSPACE*–полна.*

Идея доказательства

Во-первых, заметим, что для вычисления рекурренты нам нужно поддерживать зону размером с длину максимального сдвига и счетчик. Так как очередное значение зависит от только тех предыдущих, что не дальше, чем $\max_i a_i$, то нам действительно необходимо помнить лишь их и счетчик. Так как сдвиги, заданные массивом чисел в унарной записи, являются частью входа, необходимая нам, использованная память полиномиально ограничена размером входа.

Переопределим остановку

Вход: Машина Тьюринга M, размер зоны L в унарной записи.
Рассматривается работа машины M, начиная с такой конфигурации: $\#\underbrace{\Lambda\ldots\Lambda}_{L} q_0 \underbrace{\Lambda\ldots\Lambda}_{L}\#$, где q_0 — начальное состояния машины, а Λ — пробельный символ.
Вопрос: верно ли, что машина M останавливается?

m–сводимость

Вход задачи состоит из двух частей: описание машины, остановку которой мы хотим проверять и размера зоны L, заданного в унарной записи.
Пусть МТ $M = (Q, \Gamma, \sigma, q_0, A)$, где Q - множество состояний, $A \subseteq Q$ - множество финальных состояний, σ - функция перехода, q_0 - стартовое состояние, Γ - алфавит.

m–сводимость

Вход задачи состоит из двух частей: описание машины, остановку которой мы хотим проверять и размера зоны L, заданного в унарной записи.
Пусть МТ $M = (Q, \Gamma, \sigma, q_0, A)$, где Q - множество состояний, $A \subseteq Q$ - множество финальных состояний, σ - функция перехода, q_0 - стартовое состояние, Γ - алфавит.

Зафиксируем алфавит Σ для рекурренты, который будет состоять из пар (q, α), где $q \in Q \cup \epsilon$ и $\alpha \in \Gamma$: состояние и символ из исходного алфавита. ϵ — пустой символ, означающий, что головки над ячейкой нет. Он все еще конечен. Сводящая функция должна построить по входу задачи остановки рекурренту.

Продолжение сводимости

Первым делом нужно трансформировать начальную конфигурацию в начальные данные рекурренты (по сути переписать старый алфавит в новый), на это явно хватит полиномиального времени — алфавит - это часть описания M, также как и список состояний, поэтому алфавит новый $|\Sigma| \leqslant |M|^2$. Сдвиги будут иметь вид: $L-1, L, L+1$, поэтому нужно задать $L+1$ первых значений рекурренты. То есть начальную конфигурацию и первый символ второй конфигурации, равный $\#$. Для вычисления каждого символа начальных данных рекурренты, нам нужно потратить не более чем линейное от длины входа время (переписать значение).

Продолжение

Очередное значение рекурренты
$P(i) = \psi(P(i-1-L), P(i-L), P(i+1-L)$.
$\psi(P(i-1-L), P(i-L), P(i+1-L)) = P'$ определим так:

1. Если в первой координате $P(i-1-L), P(i-L), P(i+1-L)$, всюду ϵ, тогда положим $P' = P(i-L)$

Продолжение

Очередное значение рекурренты
$P(i) = \psi(P(i-1-L), P(i-L), P(i+1-L)$.
$\psi(P(i-1-L), P(i-L), P(i+1-L)) = P'$ определим так:

1. Если в первой координате $P(i-1-L), P(i-L), P(i+1-L)$, всюду ϵ, тогда положим $P' = P(i-L)$
2. Если $P(i-L) = (q, \alpha)$, $q \neq \epsilon$, тогда смотрим на $\sigma(P(i-L)) = (q', \alpha', m)$, m - движение головки. Если m не нулевое, тогда $P' = (\epsilon, \alpha')$. Если $m = 0$, $P' = (q', \alpha')$.

Продолжение

Очередное значение рекурренты
$P(i) = \psi(P(i-1-L), P(i-L), P(i+1-L)$.
$\psi(P(i-1-L), P(i-L), P(i+1-L)) = P'$ определим так:

1. Если в первой координате $P(i-1-L), P(i-L), P(i+1-L)$, всюду ϵ, тогда положим $P' = P(i-L)$
2. Если $P(i-L) = (q, \alpha)$, $q \neq \epsilon$, тогда смотрим на $\sigma(P(i-L)) = (q', \alpha', m)$, m - движение головки. Если m не нулевое, тогда $P' = (\epsilon, \alpha')$. Если $m = 0$, $P' = (q', \alpha')$.
3. Если $P(i-1-L) = (q, \alpha)$, $q \neq \epsilon$, то тогда $P(i-L) = (\epsilon, \beta)$. Пусть $\sigma(P(i-1-L)) = (q', \alpha', +1)$. Тогда $P' = (q', \beta)$.

Продолжение

Очередное значение рекурренты
$P(i) = \psi(P(i-1-L), P(i-L), P(i+1-L)$.
$\psi(P(i-1-L), P(i-L), P(i+1-L)) = P'$ определим так:

1. Если в первой координате $P(i-1-L), P(i-L), P(i+1-L)$, всюду ϵ, тогда положим $P' = P(i-L)$
2. Если $P(i-L) = (q, \alpha)$, $q \neq \epsilon$, тогда смотрим на $\sigma(P(i-L)) = (q', \alpha', m)$, m - движение головки. Если m не нулевое, тогда $P' = (\epsilon, \alpha')$. Если $m = 0$, $P' = (q', \alpha')$.
3. Если $P(i-1-L) = (q, \alpha)$, $q \neq \epsilon$, то тогда $P(i-L) = (\epsilon, \beta)$. Пусть $\sigma(P(i-1-L)) = (q', \alpha', +1)$. Тогда $P' = (q', \beta)$.
4. Если $P(i+1-L) = (q, \alpha)$, $q \neq \epsilon$, то тогда $P(i-L) = (\epsilon, \beta)$. Пусть $\sigma(P(i+1-L)) = (q', \alpha', -1)$. Тогда $P' = (q', \beta)$.

Продолжение

Очередное значение рекурренты
$P(i) = \psi(P(i-1-L), P(i-L), P(i+1-L))$.
$\psi(P(i-1-L), P(i-L), P(i+1-L)) = P'$ определим так:

1. Если в первой координате $P(i-1-L), P(i-L), P(i+1-L)$, всюду ϵ, тогда положим $P' = P(i-L)$
2. Если $P(i-L) = (q, \alpha)$, $q \neq \epsilon$, тогда смотрим на $\sigma(P(i-L)) = (q', \alpha', m)$, m - движение головки. Если m не нулевое, тогда $P' = (\epsilon, \alpha')$. Если $m = 0$, $P' = (q', \alpha')$.
3. Если $P(i-1-L) = (q, \alpha)$, $q \neq \epsilon$, то тогда $P(i-L) = (\epsilon, \beta)$. Пусть $\sigma(P(i-1-L)) = (q', \alpha', +1)$. Тогда $P' = (q', \beta)$.
4. Если $P(i+1-L) = (q, \alpha)$, $q \neq \epsilon$, то тогда $P(i-L) = (\epsilon, \beta)$. Пусть $\sigma(P(i+1-L)) = (q', \alpha', -1)$. Тогда $P' = (q', \beta)$.
5. Иначе, $P' = P(i-L)$.

Продолжение

Очередное значение рекурренты
$P(i) = \psi(P(i-1-L), P(i-L), P(i+1-L))$.
$\psi(P(i-1-L), P(i-L), P(i+1-L)) = P'$ определим так:

1. Если в первой координате $P(i-1-L), P(i-L), P(i+1-L)$, всюду ϵ, тогда положим $P' = P(i-L)$
2. Если $P(i-L) = (q, \alpha)$, $q \neq \epsilon$, тогда смотрим на $\sigma(P(i-L)) = (q', \alpha', m)$, m - движение головки. Если m не нулевое, тогда $P' = (\epsilon, \alpha')$. Если $m = 0$, $P' = (q', \alpha')$.
3. Если $P(i-1-L) = (q, \alpha)$, $q \neq \epsilon$, то тогда $P(i-L) = (\epsilon, \beta)$. Пусть $\sigma(P(i-1-L)) = (q', \alpha', +1)$. Тогда $P' = (q', \beta)$.
4. Если $P(i+1-L) = (q, \alpha)$, $q \neq \epsilon$, то тогда $P(i-L) = (\epsilon, \beta)$. Пусть $\sigma(P(i+1-L)) = (q', \alpha', -1)$. Тогда $P' = (q', \beta)$.
5. Иначе, $P' = P(i-L)$.
6. Функцию ψ можно задать таблицей значений размером $|\Sigma|^3$.

Продолжение

Утверждение (Без доказательства)

Рекуррента генерирует последовательность слов в алфавите Σ, кодирующую последовательность конфигураций M при работе из указанной начальной конфигурации.

Продолжение

Утверждение (Без доказательства)

Рекуррента генерирует последовательность слов в алфавите Σ, кодирующую последовательность конфигураций M при работе из указанной начальной конфигурации.

Тогда, исходная машина M останавливается если и только если $t+1$ значение рекурренты принадлежит множеству β финальных состояний рекурренты, которое устроено как $(q \in Q, q_a \in A)$. Следуя лемме 1 выберем значение, большее числа конфигураций M и сводимость построена.

Булева рекуррента

Булевой рекуррентой назовём рекурренту с алфавитом $\Sigma = \{0, 1\}$

Теорема (М.Н. Вялый — П., 2024)

Аналогичная задача для булевой рекурренты также PSPACE*–полна.*

Буквальное повторение предыдущего рассуждения не сработает. Теперь у нас вместо символов языка биты, по конкретному биту нужно как–то восстановить символ языка, чтобы воспользоваться функцией перехода.

Кодировка

Кодировка τ

Сопоставим каждому символу $a_i \in \Sigma = \{a_1, \dots a_k\}$ слово в двоичном алфавите $\tau(a_i)$ по следующему правилу:
$a_i \to 1100 w_i 0011, w_i = 00 \dots \underbrace{1}_{i} \dots 00$. Слову $a = a_1 a_2 \dots a_n$ сопоставим двоичное слово $\tau(a) = \tau(a_1)\tau(a_2)\dots\tau(a_n)$.
Длина кода символа $l = k + 8$.

Лемма (Без доказательства)

Кодировка τ позволяет для произвольной позиции бита x по его окрестности размера $2l$ однозначно восстановить исходный символ алфавита Σ.

EXP

Определение

$EXP = \bigcup_{c \geqslant 0} DTIME(2^{n^c})$, где n — длина входа задачи.

Полнота

Язык A называется EXP-полным, если $A \in EXP$ и $\forall B \in EXP$

$$B \leqslant_p A$$

Рекуррента в конечном алфавите с произвольными сдвигами

Формулировка

Вход: начальные данные заданы булевой схемой $C(y_0, \ldots, y_{n-1})$, значение которой $P(y)$, где y — число, двоичная запись которого равна $y_0, \ldots, y_{n-1}$.
Вопрос: Верно ли, что $P(x) = 1$?

Рекуррента в конечном алфавите с произвольными сдвигами

Формулировка

Вход: начальные данные заданы булевой схемой $C(y_0, \ldots, y_{n-1})$, значение которой $P(y)$, где y — число, двоичная запись которого равна $y_0, \ldots, y_{n-1}$.
Вопрос: Верно ли, что $P(x) = 1$?

Теорема (М.Н. Вялый — П., 2024)

Задача вычисления рекурренты с произвольными сдвигами EXP*-полна.*

От чего сводить?

Еще одна остановка

По данной детерминированной МТ M и числу m, представленному в двоичной записи, определить, верно ли, что M останавливается на пустой строке не более чем за m шагов.

От чего сводить?

Еще одна остановка

По данной детерминированной МТ M и числу m, представленному в двоичной записи, определить, верно ли, что M останавливается на пустой строке не более чем за m шагов.

Утверждение

Такая задача EXP*–полна.*

Доказательство

Пусть n — общая длина записи M и m. Тогда $m \leqslant 2n$, каждая конфигурация M занимает не более чем $2n$ клеток, так что каждый шаг M можно выполнить не более чем за 2^{2n} шагов. Отсюда работу M на ϵ можно промоделировать не более чем за 2^{3n} шагов.

Доказательство

Пусть n — общая длина записи M и m. Тогда $m \leqslant 2n$, каждая конфигурация M занимает не более чем $2n$ клеток, так что каждый шаг M можно выполнить не более чем за 2^{2n} шагов. Отсюда работу M на ϵ можно промоделировать не более чем за 2^{3n} шагов.

Пусть L — произвольная задача из EXP, решаемая некоторой МТ M за время 2^{n^k} . Эта задача сводится к нашей задаче так. По данной строке w, сперва строится машина Тьюринга M_w, которая, получив на входе пустую строку, записывает на ленту w, после чего работает, как M. Далее на выходную ленту выдаётся пара из машины M_w и числа $m = 2^{|w|^k}$ — оно потребует всего $|w|^k$ битов.