

Тотальная выводимость в порождающих грамматиках

Карлов Борис Николаевич

Тверской государственный университет

25 июня 2025 года

- 1 Порождающие грамматики
- 2 Проблема выводимости
- 3 Сложность тотальной выводимости
- 4 Тотальная выводимость в КС-грамматиках
- 5 Дальнейшие исследования

- 1 Порождающие грамматики
- 2 Проблема выводимости
- 3 Сложность тотальной выводимости
- 4 Тотальная выводимость в КС-грамматиках
- 5 Дальнейшие исследования

Определение

Алфавит — множество символов.

Примеры

- Русский алфавит: а, б, в, . . . , э, ю, я.
- Латинский алфавит: а, b, с, . . . , x, у, z.
- Десятичные цифры: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- Множество лексем: имена переменных, константы, ключевые слова, скобки и т. п.
- Множество словоформ русского языка.

Определение

Слово (цепочка) — конечная последовательность СИМВОЛОВ.

Примеры

- «грамматика» — слово в русском алфавите
- «grammar» — слово в латинском алфавите
- «2021» — слово в алфавите цифр
- ε — пустое слово (не содержит символов)
- «Летом здесь играют дети» — цепочка в алфавите русских слов (предложение).

Определение

Язык — произвольное множество слов.

Примеры

- Все слова из букв a и b , которые одинаково читаются в обоих направлениях (палиндромы).
- Все слова из цифр, представляющие числа кратные трём.
- Арифметические выражения.
- Программы на языке C++.
- Русский язык.
- Σ^* — множество всех слов в алфавите Σ .
- Σ^+ — множество всех непустых слов в алфавите Σ .

- Список всех слов, если язык конечный.
- Автомат — устройство, которое читает слово и определяет, принадлежит ли оно языку.
- Грамматика — набор правил, определяющих, как устроены слова.

Определение (Н. Хомский)

Порождающая грамматика — четвёрка $G = (N, \Sigma, P, S)$, где

- N — алфавит **нетерминалов**;
- Σ — алфавит **терминалов**, $\Sigma \cap N = \emptyset$;
- P — множество **правил**;
- $S \in N$ — **начальный (главный)** нетерминал.

Определение

Правило — строка вида $\alpha \rightarrow \beta$, где $\alpha, \beta \in (\Sigma \cup N)^*$, α содержит хотя бы один нетерминал.

Определение

- Выводимость за один шаг: из x можно получить y , заменив левую часть правила на правую:

$$x = u\alpha v, \quad y = u\beta v$$

для некоторых $u, v \in (\Sigma \cup N)^*$, $\alpha \rightarrow \beta \in P$.

- Обозначение: $x \Rightarrow_G y$ или $x \Rightarrow y$.

Определение

- Выводимость за несколько шагов: из x можно получить y , применив последовательность правил:

$$x \Rightarrow_G w_1 \Rightarrow_G w_2 \Rightarrow_G \dots \Rightarrow_G w_n \Rightarrow_G y.$$

- Обозначение: $x \Rightarrow_G^* y$ или $x \Rightarrow^* y$.

Определение

- Язык, **порождаемый** грамматикой G , — множество терминальных слов, выводимых из S :

$$L(G) = \{ w \in \Sigma^* : S \Rightarrow_G^* w \}.$$

Набор правил

$$\alpha \rightarrow \beta_1, \quad \alpha \rightarrow \beta_2, \quad \dots, \quad \alpha \rightarrow \beta_n$$

записывается в виде

$$\alpha \rightarrow \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$$

Пример 1

Язык

L_1 — все слова вида $\underbrace{aa \dots aa}_{n \text{ раз}} \underbrace{bb \dots bb}_{n \text{ раз}}, n > 0$:

$$L_1 = \{ a^n b^n : n > 0 \} = \{ ab, aabb, aaabbb, \dots \}$$

Грамматика для L_1

$$S \rightarrow aSb \mid ab$$

Выводы слов ab , $aabb$ и $aaabbb$

- $S \Rightarrow ab$
- $S \Rightarrow aSb \Rightarrow aabb$
- $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaabbb$

Пример 2

Язык

L_2 — арифметические выражения, построенные из констант, скобок, операций $+$ и $*$

Терминалы

- $(,)$ — скобки
- $+, *$ — операции
- c — константа

Нетерминалы

- E — выражение (англ. expression), начальный нетерминал
- T — слагаемое (англ. term)
- F — множитель (англ. factor)

Пример 2

КС-грамматика для L_2

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow c \mid (E)$$

Вывод выражения $2 + 3 * 5$

$$\begin{aligned} E &\Rightarrow E + T \Rightarrow T + T \Rightarrow F + T \Rightarrow \\ &\Rightarrow F + T * F \Rightarrow F + F * F \Rightarrow 2 + F * F \Rightarrow \\ &\Rightarrow 2 + 3 * F \Rightarrow 2 + 3 * 5 \end{aligned}$$

Определение

- Наиболее общий тип грамматик не имеет ограничений на вид правил.
- Пример правила:

$$aBAb \rightarrow Ca$$

Теорема

Класс языков, порождаемых грамматиками без ограничений на правила, совпадает с классом рекурсивно перечислимых языков.

Определение

- Грамматика **неукорачивающая**, если для любого правила $\alpha \rightarrow \beta$ выполняется $|\alpha| \leq |\beta|$.
- Пример правила:

$$aBAa \rightarrow bCaA$$

Определение

- Грамматика **контекстно-зависимая** (КЗ-грамматика), если любое правило имеет вид $\alpha A \beta \rightarrow \alpha \gamma \beta$, где $A \in N$, $\alpha, \beta \in (\Sigma \cup N)^*$, $\gamma \in (\Sigma \cup N)^+$.
- Пример правила:

$$aBAa \rightarrow aCAa$$

Теорема

Следующие классы языков совпадают:

- порождаемые неукорачивающими грамматиками;
- порождаемые контекстно-зависимыми грамматиками;
- распознаваемые недетерминированными машинами Тьюринга в линейной памяти, то есть $\text{NSPACE}(n)$.

Определение

- Грамматика **контекстно-свободная** (КС-грамматика), если любое правило имеет вид $A \rightarrow \alpha$, где $A \in N$, $\alpha \in (\Sigma \cup N)^*$.
- Пример правила:

$$A \rightarrow bCaA$$

Теорема

Класс контекстно-свободных языков совпадает с классом языков, распознаваемых недетерминированными автоматами с магазинной памятью.

Определение

- Грамматика **праволинейная**, если любое правило имеет вид $A \rightarrow xB$ или $A \rightarrow x$, где $A, B \in N$, $x \in \Sigma^*$.
- Пример правила:

$$A \rightarrow baaB$$

Теорема

Класс языков, порождаемых праволинейными грамматиками, совпадает с классом регулярных языков.

- 1 Порождающие грамматики
- 2 Проблема выводимости
- 3 Сложность тотальной выводимости
- 4 Тотальная выводимость в КС-грамматиках
- 5 Дальнейшие исследования

Определение

Вход:

- грамматика $G = (N, \Sigma, P, S)$;
- слово $w = a_1 a_2 \dots a_n \in \Sigma^*$.

Вопрос:

- существует ли вывод $S \Rightarrow_G^* w$?

Практическое значение

- Дана грамматика для языка программирования и программа.
- Нужно проверить, является ли программа правильной.

Вычислительная сложность

Тип грамматики	Сложность проблемы
Произвольная	m -полная
Неукорачивающая	PSPACE-полная
Контекстно-зависимая	PSPACE-полная
Контекстно-свободная	Полиномиально разрешима

Вычислительная сложность

- Алгоритм Кока – Янгера – Касами.
 - Применим только к грамматикам с правилами вида $A \rightarrow BC, A \rightarrow a$.
 - Сложность: $O(n^3)$.
- Алгоритм Эрли.
 - Сложность: $O(n^3)$ для произвольных КС-грамматик.
 - Сложность: $O(n^2)$ для однозначных КС-грамматик.
- Алгоритм Валианта.
 - Сложность: $O(M(n))$, где $M(n)$ — время умножения двух булевых матриц размера $n \times n$.

Определение

Пусть

- $G = (N, \Sigma, P, S)$ — грамматика;
- R — мультимножество правил;
- M — мультимножество нетерминалов.

Вывод $S \Rightarrow_G^* w$ называется

- **R -тотальным**, если каждое правило применяется в нём не меньшее число раз, чем его кратность в R ;
- **M -тотальным**, если каждый нетерминал встречается в нём не меньшее число раз, чем его кратность в M .

Частные случаи

- $R = P$: каждое правило применяется хотя бы один раз.
- $M = N$: каждый нетерминал встречается хотя бы один раз.

Пример

- Грамматика с правилами

$$S \rightarrow aSb \mid ab$$

порождает язык

$$L = \{ a^n b^n : n > 0 \}.$$

- Для слова $aabb$ существует P -тотальный вывод:

$$S \Rightarrow aSb \Rightarrow aabb$$

- Для слова ab не существует P -тотального вывода, так как единственный вывод имеет вид

$$S \Rightarrow ab$$

Определение

Вход:

- грамматика $G = (N, \Sigma, P, S)$;
- слово $w = a_1 a_2 \dots a_n \in \Sigma^*$;
- мультимножество правил R (или нетерминалов M).

Вопрос:

- существует ли R -тотальный (или M -тотальный) вывод $S \Rightarrow_G^* w$?

Происхождение проблемы

- В классической логике высказывание «из $A_1, \dots, A_n$ следует B » истинно, если либо среди посылок $A_1, \dots, A_n$ есть хотя бы одна ложная, либо заключение B истинно.
- Некоторые истинные высказывания оказываются неправильными в «бытовом» понимании.
- Пример: если Земля круглая, то $2 < 3$.

- В релевантной логике высказывание «из $A_1, \dots, A_n$ следует B » истинно, если все посылки $A_1, \dots, A_n$ использованы в доказательстве B .
- В некоторых релевантных вариантах исчисления Ламбека использование всех посылок соответствует тотальности вывода в КС-грамматике.

- 1 Порождающие грамматики
- 2 Проблема выводимости
- 3 Сложность тотальной выводимости**
- 4 Тотальная выводимость в КС-грамматиках
- 5 Дальнейшие исследования

Варианты проблемы

- Общий вариант: на вход подаётся и грамматика, и слово.
- Частный случай, вариант 1: грамматика зафиксирована, на вход подаётся только слово.
- Частный случай, вариант 2: слово зафиксировано, на вход подаётся только грамматика.

Теорема

Проблема R -тотальной выводимости

- m -полна для произвольных порождающих грамматик;
- PSPACE-полна для контекстно-зависимых и неукорачивающих грамматик.

Полнота имеет место уже при $R = P$.

Идея доказательства

- Проблема «обычной» выводимости сводится к проблеме тотальной выводимости.
- В слово добавляются технические фрагменты для применения «лишних» правил.

Теорема

Проблема R -тотальной выводимости для КС-грамматик является NP-полной. Полнота имеет место уже при $R = P$.

Идея доказательства

- NP-трудность доказывается путём сведения задачи 3-РАЗБИЕНИЕ.
- Алгоритм аккуратно угадывает вывод и хранит его в сжатом виде.

Теорема

Пусть

- $G = (N, \Sigma, P, S)$ — неукорачивающая грамматика;
- $w \in \Sigma^*$ — терминальное слово;
- R — мультимножество правил;
- $|w| = n$, $|\Sigma| = k$, $|N| = m$, $|P| = p$;
- ℓ_1 — суммарная длина кратностей из R ;
- ℓ_2 — суммарная длина правил из P .

Тогда R -тотальную выводимость слова w в грамматике G можно проверить недетерминированным алгоритмом за время

$$O(np(k + m)^n + \ell_1 + \ell_2).$$

Теорема

Пусть

- $G = (N, \Sigma, P, S)$ — неукорачивающая грамматика;
- $w \in \Sigma^*$ — терминальное слово;
- M — мультимножество нетерминалов;
- $|w| = n$, $|\Sigma| = k$, $|N| = m$, $|P| = p$;
- ℓ_1 — суммарная длина кратностей из M ;
- ℓ_2 — суммарная длина правил из P .

Тогда M -тотальную выводимость слова w в грамматике G можно проверить недетерминированным алгоритмом за время

$$O(n(k + m)^{2n} + \ell_1 + \ell_2).$$

Теорема

Если произвольная, неукорачивающая или контекстно-зависимая грамматика фиксирована, то сложность проблемы R -тотальной выводимости совпадает со сложностью обычной выводимости.

Идея доказательства

- Существует грамматика, порождающая m -полный язык.
- Существует КЗ-грамматика, порождающая PSPACE-полный язык.

Теорема

Если $|w| = 1$, то проблемы R -тотальной и M -тотальной выводимости слова w в неукорачивающей грамматике G принадлежат классу P .

Идея доказательства

- Так как грамматика неукорачивающая, то правила имеют вид $A \rightarrow B$ или $A \rightarrow a$.
- Построить ориентированный граф с вершинами $\Sigma \cup N$ и рёбрами R .
- Найти путь из S в w , проходящий через рёбра из R заданное число раз (выделение сильно связанных компонент, поиск самого длинного пути в ациклическом графе).

Теорема (С. М. Дудаков)

Проблема определения P -тотальной выводимости слова ε по КС-грамматике G является NP-полной.

Теорема (С. М. Дудаков)

Для любого фиксированного слова w проблема определения P -тотальной выводимости слова w по КС-грамматике G является NP-полной.

Теорема

Пусть

- Σ — произвольный алфавит;
- $w \in \Sigma^*$ — произвольное фиксированное слово длины не меньше 2.

Проблема определения M -тотальной выводимости слова w по неукорачивающей грамматике $G = (N, \Sigma, P, S)$ является NP-полной. Полнота имеет место уже при $M = N$.

Теорема

Пусть

- Σ — произвольный алфавит;
- $w \in \Sigma^*$ — произвольное фиксированное слово длины не меньше 2.

Проблема определения R -тотальной выводимости слова w по неукорачивающей грамматике $G = (N, \Sigma, P, S)$ и мультимножеству правил R является NP-полной. Полнота имеет место, даже если R — обычное множество.

Теорема

Проблемы R -тотальной и M -тотальной выводимости фиксированного слова w длины не меньше 2 в контекстно-зависимой грамматике G являются NP-полными.

Идея доказательства

- Перестроить неукорачивающие грамматики из предыдущих теорем в контекстно-зависимые стандартным способом.
- Добавить правила, использующие все новые нетерминалы.

- 1 Порождающие грамматики
- 2 Проблема выводимости
- 3 Сложность тотальной выводимости
- 4 Тотальная выводимость в КС-грамматиках**
- 5 Дальнейшие исследования

Трудность 1

- Кратности элементов хранятся в двоичной записи.
- Числовое значение экспоненциально велико по сравнению с длиной входа.
- Алгоритм, непосредственно угадывающий вывод, имеет сложность не меньше экспоненциальной.

Трудность 2

- Может оказаться, что любой тотальный вывод должен содержать циклы.
- Пример грамматики G :

$$S \rightarrow T \mid a, \quad T \rightarrow S$$

- Для слова a существует тотальный вывод:

$$S \Rightarrow T \Rightarrow S \Rightarrow a$$

- Любой тотальный вывод слова в G должен содержать цикл $S \Rightarrow^+ S$.
- Непонятно, как ограничить длину вывода.

Трудность 3

- Даже при отсутствии циклов тотальный вывод может иметь экспоненциальную длину.
- Пример грамматики G :

$$A_1 \rightarrow A_2 A_2$$

$$A_2 \rightarrow A_3 A_3$$

...

$$A_{m-1} \rightarrow A_m A_m$$

$$A_m \rightarrow \varepsilon$$

- Тотальный вывод $A_1 \Rightarrow^* \varepsilon$ имеет длину $2^m - 1$.

Идея 1

- Пусть существует цикл $A \Rightarrow^+ A$, в котором встречается правило ρ .
- Если в выводе $S \Rightarrow^* w$ встречается нетерминал A , то правило ρ может быть использовано сколь угодно много раз (вставляем цикл).
- Для каждого нетерминала A найдём множество правил $C(A)$, которые могут использоваться в циклах $A \Rightarrow^+ A$ (аналогично устранению цепных правил).
- Угадаем вывод без циклов, а потом расширим его циклами.

Идея 2

- Сначала будем строить тотальные выводы пустого слова.
- Если $\alpha \Rightarrow^* \varepsilon$, то вывод сохранится при любой перестановке символов из α .
- Нужно хранить не слово α , а список пар вида $(A, |\alpha|_A)$ для $A \in N$.

Алгоритм

- Строим дерево вывода по уровням.
- Строим дерево высоты не более $|N|$, чтобы избежать циклов большой длины.
- Каждый уровень храним в сжатом виде.
- Если есть пара (A, n) и правила $A \rightarrow \beta_1 \mid \dots \mid \beta_k$, то угадать разложение $n = n_1 + \dots + n_k$ и применить n_i раз правило $A \rightarrow \beta_i$.
- Если получилось пустое слово, то расширить вывод циклами.
- Проверить, что вывод R -тотальный.

Алгоритм

- Угадать дерево вывода (не обязательно полное) без циклов и без применений ε -правил.
- Для каждого нетерминала в кроне угадать мультимножество $R' \subseteq R$ и R' -тотальный вывод пустого слова.
- Расширить получившееся дерево циклами.
- Проверить, что получился R -тотальный вывод слова w .

- 1 Порождающие грамматики
- 2 Проблема выводимости
- 3 Сложность тотальной выводимости
- 4 Тотальная выводимость в КС-грамматиках
- 5 Дальнейшие исследования

- Какова сложность проблемы P -тотальной выводимости в неукорачивающих грамматиках?
- Какова сложность проблемы R -тотальной выводимости в случае фиксированной КС-грамматики?
- Какова сложность проблемы R -тотальной выводимости в случае праволинейных грамматик?
- Какова сложность аналогичных проблем, если заданы верхние границы на число использований правил или вхождений нетерминалов?
- Найти широкие классы грамматик, для которых проблема тотальной выводимости полиномиально разрешима.

Спасибо за внимание!