Модели вычислений

В.А. Захаров, Р.И. Подловченко

Лекция 9.

1. Свойства замкнутости КС-языков
2. Алгоритмические проблемы для КС-языков
3. Алгоритмы синтаксического анализа КС-языков
4. Детерминированные КС-языки
5. LL(k)-грамматики
6. Контекстно-зависимые грамматики

Свойства замкнутости КС-языков

Замкнут ли класс КС-языков относительно теоретико-языковых операций?

Свойства замкнутости КС-языков

Замкнут ли класс КС-языков относительно теоретико-языковых операций?

Утверждение 9.1. Если L_1 и L_2 — КС-языки, то $L_1 \cup L_2$ — КС-язык.

Доказательство. Объединим КС-грамматики языков L_1 и L_2, добавим новый начальный нетерминал S_0 и два новых грамматических правила $S_0 \to S_1$ и $S_0 \to S_2$. В результате получим КС-грамматику для языка $L_1 \cup L_2$.

QED

Здесь и далее мы полагаем, что все языки заданы над одним и тем же алфавитом.

Свойства замкнутости КС-языков

Утверждение 9.2. Класс КС-языков не замкнут относительно операции пересечения множеств.

Свойства замкнутости КС-языков

Утверждение 9.2. Класс КС-языков не замкнут относительно операции пересечения множеств.

Доказательство. Рассмотрим языки
$L_1 = \{a^n b^n c^k : n = m, k \geq 0\}$ и
$L_2 = \{a^k b^n c^n : n = m, k \geq 0\}$.
Нетрудно видеть, что эти языки являются контекстно-свободными.
Нетрудно видеть, что $L_1 \cap L_2 = \{a^n b^n c^n : n \geq 0\}$ — язык, не являющийся контекстно-свободным.

QED

Свойства замкнутости КС-языков

Утверждение 9.2. Класс КС-языков не замкнут относительно операции пересечения множеств.

Доказательство. Рассмотрим языки
$L_1 = \{a^n b^n c^k : n = m, k \geq 0\}$ и
$L_2 = \{a^k b^n c^n : n = m, k \geq 0\}$.
Нетрудно видеть, что эти языки являются контекстно-свободными.
Нетрудно видеть, что $L_1 \cap L_2 = \{a^n b^n c^n : n \geq 0\}$ — язык, не являющийся контекстно-свободным.

QED

Следствие. Класс КС-языков не замкнут относительно операции дополнения множеств.

СВОЙСТВА ЗАМКНУТОСТИ КС-ЯЗЫКОВ

Утверждение 9.3. Класс КС-языков замкнут относительно операции конкатенации.

Свойства замкнутости КС-языков

Утверждение 9.3. Класс КС-языков замкнут относительно операции конкатенации.

Доказательство. Самостоятельно.

Свойства замкнутости КС-языков

Утверждение 9.3. Класс КС-языков замкнут относительно операции конкатенации.

Доказательство. Самостоятельно.

Задача 1.
1) Верно ли, что если L_1 — КС-язык, а L_2 — регулярный язык, то $L_1 \cap L_2$ — КС-язык?
2) Верно ли, что если L — КС-язык, то $L^R = \{w^R : w \in L\}$ — КС-язык?

АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ ДЛЯ КС-ЯЗЫКОВ

Мы уже убедились в том, что проблемы включения $w \in L(G)$? и пустоты $L(G) = \varnothing$? алгоритмически разрешимы для КС-языков, описанных КС-грамматиками.

А как обстоят дела с другими алгоритмическими проблемами для КС-языков?

АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ ДЛЯ КС-ЯЗЫКОВ

Мы уже убедились в том, что проблемы включения $w \in L(G)$? и пустоты $L(G) = \varnothing$? алгоритмически разрешимы для КС-языков, описанных КС-грамматиками.

А как обстоят дела с другими алгоритмическими проблемами для КС-языков?

Теорема 9.4. Проблема непустоты пересечения КС-языков $L(G_1) \cap L(G_2) \neq \varnothing$, представленных КС-грамматиками, алгоритмически неразрешима.

АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ ДЛЯ КС-ЯЗЫКОВ

Доказательство. Пусть $\Sigma \cap \{a, b, c\} = \varnothing$, и пусть имеется некоторый конечный набор слов $U = \langle u_1, u_2, \ldots, u_k \rangle$ в алфавите Σ. Обозначим записью G_U КС-грамматику $(\Sigma \cup \{a, b, c\}, \{S\}, P_U, S)$, в которой множество P_U состоит из правил двух видов:

- $S \to u_i S a^i b, \ 1 \leq i \leq k$,
- $S \to u_i c a^i b, \ 1 \leq i \leq k$.

Нетрудно видеть, что язык $L(G_U)$ состоит из всех слов вида

$$u_{i_1} u_{i_2} \ldots u_{i_m} c a^{i_m} b \ldots b a^{i_2} b a^{i_1} b$$

АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ ДЛЯ КС-ЯЗЫКОВ

А теперь покажем, что Проблема Соответствий Поста сводится к задаче проверки пустоты пересечения КС-языков.

Алгоритмические проблемы для КС-языков

А теперь покажем, что Проблема Соответствий Поста сводится к задаче проверки пустоты пересечения КС-языков.

Рассмотрим произвольное конечное множество пар слов в алфавите $\Sigma, |\Sigma| \geq 2,$

$$\mathcal{P} = \{(u_1, v_1), (u_2, v_2), \ldots, (u_k, v_k)\},$$

разделим эти пары на два набора

$$U = \langle u_1, u_2, \ldots, u_k \rangle, \ V = \langle v_1, v_2, \ldots, v_k \rangle,$$

и построим грамматики G_U и G_V.

Алгоритмические проблемы для КС-языков

Тогда ПСП для системы пар $\mathcal{P}$ имеет решение

Алгоритмические проблемы для КС-языков

Тогда ПСП для системы пар $\mathcal{P}$ имеет решение

$$\Longleftrightarrow$$

$u_{i_1} u_{i_2} \dots u_{i_m} = v_{i_1} v_{i_2} \dots v_{i_m}$ для некоторой последовательности индексов $i_1, i_2, \dots, i_m$

АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ ДЛЯ КС-ЯЗЫКОВ

Тогда ПСП для системы пар $\mathcal{P}$ имеет решение

$$\Longleftrightarrow$$

$u_{i_1} u_{i_2} \dots u_{i_m} = v_{i_1} v_{i_2} \dots v_{i_m}$ для некоторой последовательности индексов $i_1, i_2, \dots, i_m$

$$\Longleftrightarrow$$

$$u_{i_1} \dots u_{i_m} c a^{i_m} b \dots b a^{i_1} b = v_{i_1} \dots v_{i_m} c a^{i_m} b \dots b a^{i_1} b$$

для той же последовательности индексов

Алгоритмические проблемы для КС-языков

Тогда ПСП для системы пар $\mathcal{P}$ имеет решение

$$\Longleftrightarrow$$

$u_{i_1}u_{i_2}\dots u_{i_m} = v_{i_1}v_{i_2}\dots v_{i_m}$ для некоторой последовательности индексов $i_1, i_2, \dots, i_m$

$$\Longleftrightarrow$$

$$u_{i_1}\dots u_{i_m}ca^{i_m}b\dots ba^{i_1}b = v_{i_1}\dots v_{i_m}ca^{i_m}b\dots ba^{i_1}b$$

для той же последовательности индексов

$$\Longleftrightarrow$$

$$L(G_U)\cap L(G_V)\neq\varnothing$$

QED

Алгоритмические проблемы для КС-языков

Пусть $U = \langle u_1, u_2, \ldots, u_k \rangle$ — произвольный конечный набор слов в алфавите Σ. Обратимся еще раз к грамматике $G_U = (\Sigma \cup \{a, b, c\}, \{S\}, P_U, S)$.

Утверждение 9.5. Язык

$$\overline{L(G_U)} = (\Sigma \cup \{a, b, c\})^* \setminus L(G_U)$$

является контекстно-свободным.

Алгоритмические проблемы для КС-языков

Пусть $U = \langle u_1, u_2, \ldots, u_k \rangle$ — произвольный конечный набор слов в алфавите Σ. Обратимся еще раз к грамматике $G_U = (\Sigma \cup \{a, b, c\}, \{S\}, P_U, S)$.

Утверждение 9.5. Язык

$$\overline{L(G_U)} = (\Sigma \cup \{a, b, c\})^* \setminus L(G_U)$$

является контекстно-свободным.

Доказательство. Попробуйте сделать это в качестве упражнения.

АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ ДЛЯ КС-ЯЗЫКОВ

Теорема 9.6. Проблема тотальности КС-языков $L(G) = \Sigma^*$, представленных КС-грамматиками, алгоритмически неразрешима.

Доказательство. Проблема соответствий Поста также сводится к проблеме тотальности КС-языков. Пусть G_U и G_V — ранее построенные КС-грамматики, соответствующие ПСП $\mathcal{P}$. Тогда

ПСП $\mathcal{P}$ не имеет решения $\iff L(G_U) \cap L(G_V) = \varnothing$
$\iff \overline{L(G_U) \cap L(G_V)} = (\Sigma \cup \{a, b, c\})^*$
$\iff \overline{L(G_U)} \cup \overline{L(G_V)} = (\Sigma \cup \{a, b, c\})^*$

Алгоритмические проблемы для КС-языков

Теорема 9.6. Проблема тотальности КС-языков $L(G) = \Sigma^*$, представленных КС-грамматиками, алгоритмически неразрешима.

Доказательство. Проблема соответствий Поста также сводится к проблеме тотальности КС-языков. Пусть G_U и G_V — ранее построенные КС-грамматики, соответствующие ПСП $\mathcal{P}$. Тогда

ПСП $\mathcal{P}$ не имеет решения $\Longleftrightarrow L(G_U) \cap L(G_V) = \varnothing$
$\Longleftrightarrow \overline{L(G_U) \cap L(G_V)} = (\Sigma \cup \{a, b, c\})^*$
$\Longleftrightarrow \overline{L(G_U)} \cup \overline{L(G_V)} = (\Sigma \cup \{a, b, c\})^*$
Остается заметить, что $\overline{L(G_U)} \cup \overline{L(G_V)}$ — КС-язык.

QED

Алгоритмические проблемы для КС-языков

Следствие. Проблема эквивалентности КС-языков $L(G_1) = L(G_2)$, представленных КС-грамматиками, алгоритмически неразрешима.

АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ ДЛЯ КС-ЯЗЫКОВ

Следствие. Проблема эквивалентности КС-языков $L(G_1) = L(G_2)$, представленных КС-грамматиками, алгоритмически неразрешима.

Задача 2. Являются ли алгоритмически разрешимыми следующие проблемы:

1. Для заданной КС-грамматики G выяснить, является ли язык $L(G)$ бесконечным.
2. Для заданной КС-грамматики G выяснить, является ли дополнение языка $L(G)$ бесконечным.
3. Для заданной КС-грамматики G выяснить, является ли язык $L(G)$ регулярным.

Алгоритмические проблемы для КС-языков

Алгоритмическая неразрешимость многих задач анализа КС-языков — это печальная новость для практиков.

Оказывается КС-грамматики чрезвычайно хрупки: очень просто вносить изменения в грамматические правила, стремясь улучшить грамматику, но очень непросто доказать, что эти изменения безопасны.

Алгоритмические проблемы для КС-языков

Алгоритмическая неразрешимость многих задач анализа КС-языков — это печальная новость для практиков.

Оказывается КС-грамматики чрезвычайно хрупки: очень просто вносить изменения в грамматические правила, стремясь улучшить грамматику, но очень непросто доказать, что эти изменения безопасны.

А как обстоит дело со сложностью алгоритмически разрешимых задач анализа КС-грамматик?

Алгоритмические проблемы для КС-языков

Наиболее важной для практических приложений КС-грамматик является задача проверки включения $w \in L(G)$ — задача синтаксического анализа. В теореме 7.13 показано, что для грамматик в нормальной форме Хомского длина грамматического вывода слова w не превосходит величины $2|w|$. Поэтому переборный алгоритм способен решить задачу синтаксического анализа. Однако его сложность экспоненциальна относительно длины анализируемого слова. Можно ли решить задачу синтаксического анализа для КС-грамматик существенно более эффективно?

Алгоритм Кока-Касами-Янгера

Опишем алгоритм, решающий задачу синтаксического анализа для КС-грамматик за время $O(n^3)$, где n — длина проверяемого слова.

Предположим, что грамматика $G = (\Sigma, \mathcal{N}, P, S)$ представлена в нормальной форме Хомского и все нетерминалы грамматики полезны.

Пусть задано слово $w = x_1 x_2 \ldots x_n$. Алгоритм Кока-Касами-Янгера (CKY-алгоритм) заполняет таблицу (матрицу) T размера $n \times n$. В ячейках таблицы (элементах матрицы) записываются множества нетерминалов.

АЛГОРИТМ КОКА-КАСАМИ-ЯНГЕРА

Пример. Пусть $G = (\{a, b\}, \{S, A, B, C\}, P, S)$, где множество P состоит из следующих правил

$$\begin{array}{rcl} S & \to & AB \mid BC, \\ A & \to & BA \mid a, \\ B & \to & CC \mid b, \\ C & \to & AB \mid a. \end{array}$$

Пусть $w = baaba$.

Алгоритм Кока-Касами-Янгера

Вначале для каждого i, $1 \leq i \leq n$ полагаем
$$T[i,i] = \{N : N \to a_i \in P\}.$$
Остальные ячейки $T[i,j], i \neq j,$ заполнены $\varnothing$.

$S \to AB|BC$

$A \to BA|a$

$B \to CC|b$

$C \to AB|a$

	b	a	a	b	a
1	b	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	a	$\varnothing$	$\varnothing$	$\varnothing$
3	$\varnothing$	$\varnothing$	a	$\varnothing$	$\varnothing$
4	$\varnothing$	$\varnothing$	$\varnothing$	b	$\varnothing$
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	a
	1	2	3	4	5

Алгоритм Кока-Касами-Янгера

Вначале для каждого i, $1 \leq i \leq n$ полагаем

$$T[i, i] = \{N : N \to a_i \in P\}.$$

Остальные ячейки $T[i, j], i \neq j$, заполнены $\varnothing$.

$S \to AB|BC$

$A \to BA|a$

$B \to CC|b$

$C \to AB|a$

	b	a	a	b	a
1	B	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	A, C	$\varnothing$	$\varnothing$	$\varnothing$
3	$\varnothing$	$\varnothing$	A, C	$\varnothing$	$\varnothing$
4	$\varnothing$	$\varnothing$	$\varnothing$	B	$\varnothing$
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A, C
	1	2	3	4	5

Алгоритм Кока-Касами-Янгера

Далее для каждой пары (i, j), $1 \le i < j \le n$ полагаем

$$T[i,j] = \{N : N \to N'N'' \in P, \exists k : i \le k < j \\ N' \in T[i,k], N'' \in T[k+1,j]\}.$$

$S \to AB|BC$

$A \to BA|a$

$B \to CC|b$

$C \to AB|a$

	b	a	a	b	a
1	B	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	A, C	$\varnothing$	$\varnothing$	$\varnothing$
3	$\varnothing$	$\varnothing$	A, C	$\varnothing$	$\varnothing$
4	$\varnothing$	$\varnothing$	$\varnothing$	B	$\varnothing$
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A, C
	1	2	3	4	5

АЛГОРИТМ КОКА-КАСАМИ-ЯНГЕРА

Далее для каждой пары $(i,j),\ 1 \leq i < j \leq n$ полагаем

$$T[i,j] = \{N : N \to N'N'' \in P, \exists k : i \leq k < j \\ N' \in T[i,k], N'' \in T[k+1,j]\}.$$

$S \to AB|BC$

$\mathbf{A} \to \mathbf{BA}|a$

$B \to CC|b$

$C \to AB|a$

$j - i = 1$

	b	a	a	b	a
1	$\mathbf{B}$	$\mathbf{A}$	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	$\mathbf{A}, C$	$\varnothing$	$\varnothing$	$\varnothing$
3	$\varnothing$	$\varnothing$	A, C	$\varnothing$	$\varnothing$
4	$\varnothing$	$\varnothing$	$\varnothing$	B	$\varnothing$
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A, C
	1	2	3	4	5

АЛГОРИТМ КОКА-КАСАМИ-ЯНГЕРА

Далее для каждой пары (i, j), $1 \le i < j \le n$ полагаем

$$T[i, j] = \{N : N \to N'N'' \in P, \exists k : i \le k < j \\ N' \in T[i, k], N'' \in T[k+1, j]\}.$$

$\mathbf{S} \to AB|\mathbf{BC}$

$A \to BA|a$

$B \to CC|b$

$C \to AB|a$

$j - i = 1$

	b	a	a	b	a
1	**B**	**S**, A	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	A, **C**	$\varnothing$	$\varnothing$	$\varnothing$
3	$\varnothing$	$\varnothing$	A, C	$\varnothing$	$\varnothing$
4	$\varnothing$	$\varnothing$	$\varnothing$	B	$\varnothing$
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A, C
	1	2	3	4	5

АЛГОРИТМ КОКА-КАСАМИ-ЯНГЕРА

Далее для каждой пары (i, j), $1 \le i < j \le n$ полагаем

$$T[i,j] = \{N : N \to N'N'' \in P, \exists k : i \le k < j \\ N' \in T[i,k], N'' \in T[k+1,j]\}.$$

$S \to AB|BC$

$A \to BA|a$

$\mathbf{B} \to \mathbf{CC}|b$

$C \to AB|a$

$j - i = 1$

	b	a	a	b	a
1	B	S, A	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	$A, \mathbf{C}$	$\mathbf{B}$	$\varnothing$	$\varnothing$
3	$\varnothing$	$\varnothing$	$A, \mathbf{C}$	$\varnothing$	$\varnothing$
4	$\varnothing$	$\varnothing$	$\varnothing$	B	$\varnothing$
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A, C
	1	2	3	4	5

АЛГОРИТМ КОКА-КАСАМИ-ЯНГЕРА

Далее для каждой пары (i, j), $1 \leq i < j \leq n$ полагаем

$$T[i,j] = \{N : N \to N'N'' \in P, \exists k : i \leq k < j \\ N' \in T[i,k], N'' \in T[k+1,j]\}.$$

$\mathbf{S} \to \mathbf{AB}|BC$

$A \to BA|a$

$B \to CC|b$

$\mathbf{C} \to \mathbf{AB}|a$

$j - i = 1$

	b	a	a	b	a
1	B	S, A	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	A, C	B	$\varnothing$	$\varnothing$
3	$\varnothing$	$\varnothing$	$\mathbf{A}, C$	$\mathbf{S}, \mathbf{C}$	$\varnothing$
4	$\varnothing$	$\varnothing$	$\varnothing$	$\mathbf{B}$	$\varnothing$
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A, C
	1	2	3	4	5

Алгоритм Кока-Касами-Янгера

Далее для каждой пары (i, j), $1 \leq i < j \leq n$ полагаем

$$T[i,j] = \{N : N \to N'N'' \in P, \exists k : i \leq k < j \\ N' \in T[i,k], N'' \in T[k+1,j]\}.$$

$S \to AB|BC$

$\mathbf{A} \to \mathbf{BA}|a$

$B \to CC|b$

$C \to AB|a$

$j - i = 1$

	b	a	a	b	a
1	B	S, A	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	A, C	B	$\varnothing$	$\varnothing$
3	$\varnothing$	$\varnothing$	A, C	S, C	$\varnothing$
4	$\varnothing$	$\varnothing$	$\varnothing$	$\mathbf{B}$	$\mathbf{A}$
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\mathbf{A}, C$
	1	2	3	4	5

Алгоритм Кока-Касами-Янгера

Далее для каждой пары (i, j), $1 \leq i < j \leq n$ полагаем

$$T[i,j] = \{N : N \to N'N'' \in P, \exists k : i \leq k < j \\ N' \in T[i,k], N'' \in T[k+1,j]\}.$$

$\mathbf{S} \to AB|\mathbf{BC}$

$A \to BA|a$

$B \to CC|b$

$C \to AB|a$

$j - i = 1$

	b	a	a	b	a
1	B	S, A	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	A, C	B	$\varnothing$	$\varnothing$
3	$\varnothing$	$\varnothing$	A, C	S, C	$\varnothing$
4	$\varnothing$	$\varnothing$	$\varnothing$	$\mathbf{B}$	$\mathbf{S}, A$
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$A, \mathbf{C}$
	1	2	3	4	5

АЛГОРИТМ КОКА-КАСАМИ-ЯНГЕРА

Далее для каждой пары (i, j), $1 \leq i < j \leq n$ полагаем

$$T[i,j] = \{N : N \to N'N'' \in P, \exists k : i \leq k < j \\ N' \in T[i,k], N'' \in T[k+1,j]\}.$$

$S \to AB|BC$

$A \to BA|a$

$B \to CC|b$

$C \to AB|a$

$j - i = 1$

	b	a	a	b	a
1	B	S, A	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	A, C	B	$\varnothing$	$\varnothing$
3	$\varnothing$	$\varnothing$	A, C	S, C	$\varnothing$
4	$\varnothing$	$\varnothing$	$\varnothing$	B	S, A
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A, C
	1	2	3	4	5

АЛГОРИТМ КОКА-КАСАМИ-ЯНГЕРА

Далее для каждой пары (i, j), $1 \leq i < j \leq n$ полагаем

$$T[i,j] = \{N : N \to N'N'' \in P, \exists k : i \leq k < j \\ N' \in T[i,k], N'' \in T[k+1,j]\}.$$

$S \to AB|BC$

$A \to BA|a$

$B \to CC|b$

$C \to AB|a$

$j - i = 2$

	b	a	a	b	a
1	B	S, A	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	A, C	B	$\varnothing$	$\varnothing$
3	$\varnothing$	$\varnothing$	A, C	S, C	$\varnothing$
4	$\varnothing$	$\varnothing$	$\varnothing$	B	S, A
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A, C
	1	2	3	4	5

АЛГОРИТМ КОКА-КАСАМИ-ЯНГЕРА

Далее для каждой пары (i, j), $1 \leq i < j \leq n$ полагаем

$$T[i,j] = \{N : N \to N'N'' \in P, \exists k : i \leq k < j \\ N' \in T[i,k], N'' \in T[k+1,j]\}.$$

$$S \to AB|BC$$
$$A \to BA|a$$
$$B \to CC|b$$
$$C \to AB|a$$

$j - i = 2$

	b	a	a	b	a
1	**B**	S, A	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	A, C	**B**	$\varnothing$	$\varnothing$
3	$\varnothing$	$\varnothing$	A, C	S, C	$\varnothing$
4	$\varnothing$	$\varnothing$	$\varnothing$	B	S, A
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A, C
	1	2	3	4	5

Алгоритм Кока-Касами-Янгера

Далее для каждой пары (i, j), $1 \leq i < j \leq n$ полагаем

$$T[i,j] = \{N : N \to N'N'' \in P, \exists k : i \leq k < j \\ N' \in T[i,k], N'' \in T[k+1,j]\}.$$

$S \to AB|BC$

$A \to BA|a$

$B \to CC|b$

$C \to AB|a$

$j - i = 2$

	b	a	a	b	a
1	B	$\mathbf{S}, \mathbf{A}$	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	A, C	B	$\varnothing$	$\varnothing$
3	$\varnothing$	$\varnothing$	$\mathbf{A}, \mathbf{C}$	S, C	$\varnothing$
4	$\varnothing$	$\varnothing$	$\varnothing$	B	S, A
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A, C
	1	2	3	4	5

Алгоритм Кока-Касами-Янгера

Далее для каждой пары (i, j), $1 \le i < j \le n$ полагаем

$$T[i,j] = \{N : N \to N'N'' \in P, \exists k : i \le k < j \\ N' \in T[i,k], N'' \in T[k+1,j]\}.$$

$S \to AB|BC$

$A \to BA|a$

$\mathbf{B} \to \mathbf{CC}|b$

$C \to AB|a$

$j - i = 2$

	b	a	a	b	a
1	B	S, A	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	$A, \mathbf{C}$	B	B	$\varnothing$
3	$\varnothing$	$\varnothing$	A, C	$S, \mathbf{C}$	$\varnothing$
4	$\varnothing$	$\varnothing$	$\varnothing$	B	S, A
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A, C
	1	2	3	4	5

Алгоритм Кока-Касами-Янгера

Далее для каждой пары (i, j), $1 \le i < j \le n$ полагаем

$$T[i,j] = \{N : N \to N'N'' \in P, \exists k : i \le k < j \\ N' \in T[i,k], N'' \in T[k+1,j]\}.$$

$S \to AB|BC$

$A \to BA|a$

$B \to CC|b$

$C \to AB|a$

$j - i = 2$

	b	a	a	b	a
1	B	S, A	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	A, C	$\mathbf{B}$	B	$\varnothing$
3	$\varnothing$	$\varnothing$	A, C	S, C	$\varnothing$
4	$\varnothing$	$\varnothing$	$\varnothing$	$\mathbf{B}$	S, A
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A, C
	1	2	3	4	5

Алгоритм Кока-Касами-Янгера

Далее для каждой пары $(i,j),\ 1 \le i < j \le n$ полагаем

$$T[i,j] = \{N : N \to N'N'' \in P, \exists k : i \le k < j \\ N' \in T[i,k], N'' \in T[k+1,j]\}.$$

$S \to AB|BC$

$A \to BA|a$

$B \to CC|b$

$C \to AB|a$

$j - i = 2$

	b	a	a	b	a
1	B	S, A	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	A, C	B	B	$\varnothing$
3	$\varnothing$	$\varnothing$	**A, C**	S, C	$\varnothing$
4	$\varnothing$	$\varnothing$	$\varnothing$	B	**S, A**
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A, C
	1	2	3	4	5

АЛГОРИТМ КОКА-КАСАМИ-ЯНГЕРА

Далее для каждой пары (i, j), $1 \le i < j \le n$ полагаем

$$T[i, j] = \{N : N \to N'N'' \in P, \exists k : i \le k < j \; N' \in T[i, k], N'' \in T[k+1, j]\}.$$

$S \to AB|BC$

$A \to BA|a$

$\mathbf{B} \to \mathbf{CC}|b$

$C \to AB|a$

$j - i = 2$

	b	a	a	b	a
1	B	S, A	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	A, C	B	B	$\varnothing$
3	$\varnothing$	$\varnothing$	A, C	$S, \mathbf{C}$	B
4	$\varnothing$	$\varnothing$	$\varnothing$	B	S, A
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$A, \mathbf{C}$
	1	2	3	4	5

АЛГОРИТМ КОКА-КАСАМИ-ЯНГЕРА

Далее для каждой пары (i, j), $1 \le i < j \le n$ полагаем

$$T[i,j] = \{N : N \to N'N'' \in P, \exists k : i \le k < j \\ N' \in T[i,k], N'' \in T[k+1,j]\}.$$

$S \to AB|BC$

$A \to BA|a$

$B \to CC|b$

$C \to AB|a$

$j - i = 3$

	b	a	a	b	a
1	B	S, A	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	A, C	B	B	$\varnothing$
3	$\varnothing$	$\varnothing$	A, C	S, C	B
4	$\varnothing$	$\varnothing$	$\varnothing$	B	S, A
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A, C
	1	2	3	4	5

АЛГОРИТМ КОКА-КАСАМИ-ЯНГЕРА

Далее для каждой пары (i, j), $1 \le i < j \le n$ полагаем

$$T[i,j] = \{N : N \to N'N'' \in P, \exists k : i \le k < j \\ N' \in T[i,k], N'' \in T[k+1,j]\}.$$

$S \to AB|BC$

$A \to BA|a$

$B \to CC|b$

$C \to AB|a$

$j - i = 3$

	b	a	a	b	a
1	B	S, A	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	A, C	B	B	S, C, A
3	$\varnothing$	$\varnothing$	A, C	S, C	B
4	$\varnothing$	$\varnothing$	$\varnothing$	B	S, A
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A, C
	1	2	3	4	5

АЛГОРИТМ КОКА-КАСАМИ-ЯНГЕРА

Далее для каждой пары (i, j), $1 \leq i < j \leq n$ полагаем

$$T[i,j] = \{N : N \to N'N'' \in P, \exists k : i \leq k < j \\ N' \in T[i,k], N'' \in T[k+1,j]\}.$$

$S \to AB|BC$

$A \to BA|a$

$B \to CC|b$

$C \to AB|a$

$j - i = 4$

	b	a	a	b	a
1	B	S, A	$\varnothing$	$\varnothing$	$\varnothing$
2	$\varnothing$	A, C	B	B	S, C, A
3	$\varnothing$	$\varnothing$	A, C	S, C	B
4	$\varnothing$	$\varnothing$	$\varnothing$	B	S, A
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A, C
	1	2	3	4	5

АЛГОРИТМ КОКА-КАСАМИ-ЯНГЕРА

Далее для каждой пары (i, j), $1 \leq i < j \leq n$ полагаем

$$T[i,j] = \{N : N \to N'N'' \in P, \exists k : i \leq k < j \\ N' \in T[i,k], N'' \in T[k+1,j]\}.$$

$S \to AB|BC$

$A \to BA|a$

$B \to CC|b$

$C \to AB|a$

$j - i = 4$

	b	a	a	b	a
1	B	S, A	$\varnothing$	$\varnothing$	S, A
2	$\varnothing$	A, C	B	B	S, C, A
3	$\varnothing$	$\varnothing$	A, C	S, C	B
4	$\varnothing$	$\varnothing$	$\varnothing$	B	S, A
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A, C
	1	2	3	4	5

АЛГОРИТМ КОКА-КАСАМИ-ЯНГЕРА

Если по окончании заполнения таблицы $S \in T[1, n]$, то $w \in L(G)$

$S \to AB|BC$

$A \to BA|a$

$B \to CC|b$

$C \to AB|a$

$baaba \in L(G)$

	b	a	a	b	a
1	B	S, A	$\varnothing$	$\varnothing$	$\mathbf{S}$ A
2	$\varnothing$	A, C	B	B	S, A
3	$\varnothing$	$\varnothing$	A, C	S, C	B
4	$\varnothing$	$\varnothing$	$\varnothing$	B	S, A
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A, C
	1	2	3	4	5

АЛГОРИТМ КОКА-КАСАМИ-ЯНГЕРА

Секрет этого алгоритма, построенного методом динамического программирования, прост.

Индукцией по $k = j - i$ нетрудно показать, что $N \in T[i,j] \iff N \xrightarrow{G}_* x_i x_{i+1} \ldots x_j$.

АЛГОРИТМ КОКА-КАСАМИ-ЯНГЕРА

$$B \xrightarrow{G} CC \xrightarrow{G} aC \xrightarrow{G} aAB \xrightarrow{G} aaB \xrightarrow{G} aab$$

$S \to AB|BC$

$A \to BA|a$

$B \to CC|b$

$C \to AB|a$

	b	a	a	b	a
1	B	S, A	$\varnothing$	$\varnothing$	S, A
2	$\varnothing$	A, C	B	B	S, A
3	$\varnothing$	$\varnothing$	A, C	S, C	B
4	$\varnothing$	$\varnothing$	$\varnothing$	B	S, A
5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A, C
	1	2	3	4	5

АЛГОРИТМ КОКА-КАСАМИ-ЯНГЕРА

Секрет этого алгоритма, построенного методом динамического программирования, прост.

Индукцией по $k = j - i$ нетрудно показать, что
$N \in T[i,j] \iff N \xrightarrow{G}_* x_i x_{i+1} \dots x_j$.

Таким образом,
$w = x_1 x_2 \dots x_n \in L(G) \iff S \in T[1,n]$.

QED

Алгоритм Кока-Касами-Янгера

CKY-алгоритм имеет сложность $O(n^3|G|)$, где $n = |w|$, а $|G|$ — суммарное число символов в правилах грамматики G. Лесли Веллиант придумал, как реализовать этот алгоритм посредством умножения булевых матриц.

Однако для практических применений CKY-алгоритм слишком затратный по времени. Более эффективные процедуры синтаксического анализа можно построить, обратившись к детерминированным магазинным автоматам.

Но детерминированные магазинные автоматы существенно слабее недетерминированных.

Детерминированные КС-языки

КС-язык называется детерминированным , если существует детерминированный магазинный автомат, распознающий этот язык.

Утверждение 9.7. Класс детерминированных КС-языков замкнут относительно операции дополнения.

ДЕТЕРМИНИРОВАННЫЕ КС-ЯЗЫКИ

КС-язык называется детерминированным, если существует детерминированный магазинный автомат, распознающий этот язык.

Утверждение 9.7. Класс детерминированных КС-языков замкнут относительно операции дополнения.

Доказательство. Вспомните, что детерминированный магазинный автомат распознает язык $L, L \subseteq \Sigma^*$, если $L(\mathcal{B}) = \{w \dashv : w \in L\}$. Граничный маркер $\dashv$ помогает очистить магазин тому автомату, который распознает дополнение $\Sigma^* \setminus L$.

QED

Детерминированные КС-языки

Утверждение 9.8. Множество слов $L = \Sigma^* \setminus \{a^k b^k c^k : k \geq 0\}$ является КС-языком.

ДЕТЕРМИНИРОВАННЫЕ КС-ЯЗЫКИ

Утверждение 9.8. Множество слов $L = \Sigma^* \setminus \{a^k b^k c^k : k \geq 0\}$ является КС-языком.

Доказательство. $L = \bigcup\limits_{\ell=0}^{6} L_\ell$, где

$$L_0 = \Sigma^* \setminus L(a^* b^* c^*),$$
$$L_1 = \{a^i b^{i+j} c^k : i \geq 0, j > 0, k \geq 0\},$$
$$L_2 = \{a^{i+j} b^j c^k : i > 0, j \geq 0, k \geq 0\},$$

а как устроены языки L_3 — L_6 вы можете догадаться и сами.

Легко видеть, что язык L_0 — регулярный, а L_1 — L_6 — КС-языки QED

Детерминированные КС-языки

Из Утверждений 9.7 и 9.8 следует

Теорема 9.9. Существуют КС-языки, которые не распознаются детерминированными магазинными автоматами.

Детерминированные КС-языки

Однако для практических нужд искусственные языки конструируются так, чтобы они распознавались детерминированными магазинными автоматами.

Как же должна быть устроена грамматика такого языка?

ДЕТЕРМИНИРОВАННЫЕ КС-ЯЗЫКИ

Однако для практических нужд искусственные языки конструируются так, чтобы они распознавались детерминированными магазинными автоматами.

Как же должна быть устроена грамматика такого языка?

Задача 3. Существует ли алгоритм, который по заданной КС-грамматике распознает, является ли КС-язык, порождаемый этой грамматикой, детерминированным?

ДЕТЕРМИНИРОВАННЫЕ КС-ЯЗЫКИ

Предположим, что нужно проверить включение $w \in L(G)$. Для этого можно построить левосторонний грамматический вывод

$$S \xrightarrow{G} \alpha_1 \xrightarrow{G} \alpha_2 \xrightarrow{G} \cdots \xrightarrow{G} \alpha_{n-1} \xrightarrow{G} w.$$

Детерминированные КС-языки

Предположим, что нужно проверить включение $w \in L(G)$. Для этого можно построить левосторонний грамматический вывод

$$S \xrightarrow{G} \alpha_1 \xrightarrow{G} \alpha_2 \xrightarrow{G} \cdots \xrightarrow{G} \alpha_{n-1} \xrightarrow{G} w.$$

Чтобы этот вывод мог построить детерминированный магазинный автомат, он должен уметь выбирать правила грамматики, используя буквы слова w как «подсказки», а грамматика G должна обеспечить однозначность этой «подсказки».

LL(k)-ГРАММАТИКИ

КС-грамматика $G = (\Sigma, \mathcal{N}, P, S)$ называется разделенной, если

1. правая часть любого правила начинается терминалом,
2. правые части правил для одного и того же нетерминала начинаются разными терминалами.

Пример.

$$\begin{array}{lcl} S & \to & aSS \mid bAS \mid cBBa \\ A & \to & aBSS \mid bba \mid cAbA \\ B & \to & abc \mid bbB \mid cBbaS \end{array}$$

LL(k)-ГРАММАТИКИ

Если КС-грамматика является разделенной, то алгоритм, предложенный в доказательстве Теоремы 8.2, позволяет построить детерминированный магазинный автомат, который распознает язык, порождаемый этой грамматикой.

LL(k)-ГРАММАТИКИ

Если КС-грамматика является разделенной, то алгоритм, предложенный в доказательстве Теоремы 8.2, позволяет построить детерминированный магазинный автомат, который распознает язык, порождаемый этой грамматикой.

К сожалению, на практике разделенные грамматики возникают нечасто. Вот типичный пример БНФ-формул, не подпадающих под определение разделенной грамматики:

«список» ::= «пусто» | «элемент» «список»

«пусто» ::= ε

LL(k)-ГРАММАТИКИ

Более общее определение «однозначной подсказки» при выборе правил грамматического вывода можно сформулировать так.

Для каждого слова $w = x_1 x_2 \dots x_m$ и $k, k > 0$, определим функцию

$$Pref_k(w) = \begin{cases} x_1 x_2 \dots x_k, \text{ если } |w| \geq k, \\ w, \text{ если } |w| < k. \end{cases}$$

LL(k)-ГРАММАТИКИ

Грамматика G называется LL(k)-грамматикой, если для любых двух левосторонних грамматических выводов

$$
\begin{array}{ccccccc}
S & \xrightarrow{G}_{*} & wN\alpha & \xrightarrow{G} & w\beta'\alpha & \xrightarrow{G}_{*} & wu' \\
S & \xrightarrow{G}_{*} & wN\alpha & \xrightarrow{G} & w\beta''\alpha & \xrightarrow{G}_{*} & wu''
\end{array}
$$

равенство $Pref_k(u') = Pref_k(u'')$ влечет равенство $\beta' = \beta''$.

LL(k)-ГРАММАТИКИ

Грамматика G называется LL(k)-грамматикой, если для любых двух левосторонних грамматических выводов

$$
\begin{array}{ccccccc}
S & \xrightarrow{G}_* & wN\alpha & \xrightarrow{G} & w\beta'\alpha & \xrightarrow{G}_* & wu' \\
S & \xrightarrow{G}_* & wN\alpha & \xrightarrow{G} & w\beta''\alpha & \xrightarrow{G}_* & wu''
\end{array}
$$

равенство $Pref_k(u') = Pref_k(u'')$ влечет равенство $\beta' = \beta''$.

Иными словами, чтобы узнать, каким правилом $N \to \beta$ воспользоваться для вывода из нетерминала N заданного слова u, нужно прочитать первые k букв этого слова.

LL(k)-ГРАММАТИКИ

Сценарий работы магазинного автомата, соответствующего LL(k)-грамматике

Магазин:

$\mathbf{S}$

Слово на ленте:

$\mathbf{a}_1\ \mathbf{a}_2\ \mathbf{a}_3\ \mathbf{a}_4\ \mathbf{a}_5\ \mathbf{a}_6\ \mathbf{a}_7\ \mathbf{a}_8\ \ldots$

LL(k)-ГРАММАТИКИ

Сценарий работы магазинного автомата, соответствующего LL(k)-грамматике

Магазин:

S

Слово на ленте:

$\boxed{a_1\ a_2\ a_3}\ a_4\ a_5\ a_6\ a_7\ a_8\ \ldots$

Устанавливается поле зрения автомата: прочитываются первые **k** букв слова.

LL(k)-ГРАММАТИКИ

Сценарий работы магазинного автомата, соответствующего LL(k)-грамматике

Магазин:

S

Слово на ленте:

$\boxed{\mathbf{a_1\ a_2\ a_3}}\ \mathbf{a_4\ a_5\ a_6\ a_7\ a_8}\ \ldots$

Вычисляется грамматическое правило:

$Tab_{G,k}:\ (S, a_1a_2a_3) \Rightarrow (S \to a_1ABa_5)$

LL(k)-ГРАММАТИКИ

Сценарий работы магазинного автомата, соответствующего LL(k)-грамматике

Магазин:

$\mathbf{a_1 A B a_5}$

Слово на ленте:

$\boxed{\mathbf{a_1\ a_2\ a_3}}\ \mathbf{a_4\ a_5\ a_6\ a_7\ a_8}\ \ldots$

Выбранное правило применяется: изменяется содержимое магазина

LL(k)-ГРАММАТИКИ

Сценарий работы магазинного автомата, соответствующего LL(k)-грамматике

Магазин:

$AB a_5$

Слово на ленте:

$a_1 \boxed{a_2\ a_3\ a_4}\ a_5\ a_6\ a_7\ a_8\ \ldots$

Поле зрения автомата смещается:

сдвиг по ленте и удаление букв из магазина

LL(k)-ГРАММАТИКИ

Сценарий работы магазинного автомата, соответствующего LL(k)-грамматике

Магазин:

$\mathbf{ABa_5}$

Слово на ленте:

$\mathbf{a_1} \boxed{\mathbf{a_2\ a_3\ a_4}} \mathbf{a_5\ a_6\ a_7\ a_8} \ \ldots$

Вычисляется грамматическое правило:

$Tab: \ (A, a_2a_3a_4) \Rightarrow (A \to Ba_2a_3C)$

LL(k)-ГРАММАТИКИ

Сценарий работы магазинного автомата, соответствующего LL(k)-грамматике

Магазин:

$\mathbf{B a_2 a_3 C B a_5}$

Слово на ленте:

$\mathbf{a_1 \boxed{a_2\ a_3\ a_4}\ a_5\ a_6\ a_7\ a_8\ \ldots}$

Выбранное правило применяется: изменяется содержимое магазина

LL(k)-ГРАММАТИКИ

Сценарий работы магазинного автомата, соответствующего LL(k)-грамматике

Магазин:

$\mathbf{B a_2 a_3 C B a_5}$

Слово на ленте:

$\mathbf{a_1} \boxed{\mathbf{a_2\ a_3\ a_4}} \mathbf{a_5\ a_6\ a_7\ a_8} \ \ldots$

Вычисляется грамматическое правило:

$Tab_{G,k}: \ (B, a_2 a_3 a_4) \Rightarrow (B \rightarrow \varepsilon)$

LL(k)-ГРАММАТИКИ

Сценарий работы магазинного автомата, соответствующего LL(k)-грамматике

Магазин:

$\mathbf{a_2 a_3 C B a_5}$

Слово на ленте:

$\mathbf{a_1} \boxed{\mathbf{a_2\ a_3\ a_4}}\ \mathbf{a_5\ a_6\ a_7\ a_8}\ \ldots$

Выбранное правило применяется: изменяется содержимое магазина

LL(k)-ГРАММАТИКИ

Сценарий работы магазинного автомата, соответствующего LL(k)-грамматике

Магазин:

CBa_5

Слово на ленте:

$a_1\ a_2\ a_3\ \boxed{a_4\ a_5\ a_6}\ a_7\ a_8\ \ldots$

Поле зрения автомата смещается:
сдвиг по ленте и удаление букв из магазина, **и т. д.**

LL(k)-ГРАММАТИКИ

Как видно из описанного сценария работы магазинного автомата, чтобы построить такой автомат для заданной КС-грамматики G, нужно уметь

1. определить наименьшее k, для которого грамматика G принадлежит класса LL(k),

LL(k)-ГРАММАТИКИ

Как видно из описанного сценария работы магазинного автомата, чтобы построить такой автомат для заданной КС-грамматики G, нужно уметь

1. определить наименьшее k, для которого грамматика G принадлежит класса LL(k),
2. построить таблицу $Tab_{G,k}$.

Оценим, насколько трудно решать обе эти задачи.

LL(k)-ГРАММАТИКИ

Грамматику, принадлежащую классу LL(k) для некоторого k, будем называть LL-грамматикой. КС-язык, порождаемый LL(k)-грамматикой (LL-грамматикой), будем также называть LL(k)-языком (соответственно LL-языком).

LL(k)-ГРАММАТИКИ

Грамматику, принадлежащую классу LL(k) для некоторого k, будем называть LL-грамматикой. КС-язык, порождаемый LL(k)-грамматикой (LL-грамматикой), будем также называть LL(k)-языком (соответственно LL-языком).

1. Некоторые детерминированные КС-языки не являются LL-языками.

Задача 4. [Трудная] Доказать, что КС-язык

$$L = \{a^n 0 b^n : n \geq 0\} \cup \{a^n 1 b^{2n} : n \geq 0\}$$

является детерминированным, но не является LL-языком.

LL(k)-ГРАММАТИКИ

2. Непросто узнать, является ли заданный КС-язык LL-языком.

LL(k)-ГРАММАТИКИ

2. Непросто узнать, является ли заданный КС-язык LL-языком.

Задача 5. [Трудная] Доказать, что не существует алгоритма, способного для заданной грамматики G выяснить, является ли эта грамматика LL(k)-грамматикой для некоторого k.

LL(k)-ГРАММАТИКИ

2. Непросто узнать, является ли заданный КС-язык LL-языком.

Задача 5. [Трудная] Доказать, что не существует алгоритма, способного для заданной грамматики G выяснить, является ли эта грамматика LL(k)-грамматикой для некоторого k.

Задача 6. [Трудная] Доказать, что не существует алгоритма, способного для заданной грамматики G выяснить, является ли порождаемый этой грамматикой язык LL-языком.

LL(k)-ГРАММАТИКИ

3. Однако не все так плохо. Свойство КС-грамматики принадлежать классу LL является полуразрешимым.

Задача 7. Построить алгоритм, который для заданной КС-грамматики G и заданного k, проверяет, является ли G LL(k)-грамматикой.

LL(k)-ГРАММАТИКИ

3. Однако не все так плохо. Свойство КС-грамматики принадлежать классу LL является полуразрешимым.

Задача 7. Построить алгоритм, который для заданной КС-грамматики G и заданного k, проверяет, является ли G LL(k)-грамматикой.

Посмотрим, как решать эту задачу для случая $k = 1$.

Далее будем, не оговаривая этого особо, полагать, что рассматриваемые грамматики не содержат бесполезных правил, т.е. правил, которые не участвуют в выводах терминальных слов.

LL(k)-ГРАММАТИКИ

Пусть задана КС-грамматика $G = (\Sigma, \mathcal{N}, P, S)$.
Введем две характеристики правил
грамматического вывода $N \to \alpha$.

$$\begin{aligned} First_G(\alpha) &= \{a : a \in \Sigma, \alpha \xrightarrow{G}_* a\beta\}; \\ After_G(N) &= \{b : b \in \Sigma, S \xrightarrow{G}_* \beta N b\gamma\}. \end{aligned}$$

LL(k)-ГРАММАТИКИ

Пусть задана КС-грамматика $G = (\Sigma, \mathcal{N}, P, S)$. Введем две характеристики правил грамматического вывода $N \to \alpha$.

$$
\begin{aligned}
First_G(\alpha) &= \{a : a \in \Sigma, \alpha \xrightarrow{G}_* a\beta\}; \\
After_G(N) &= \{b : b \in \Sigma, S \xrightarrow{G}_* \beta N b \gamma\}.
\end{aligned}
$$

$First_G(\alpha)$ — это множество всех терминалов, которыми начинаются строки, выводимые в грамматике G из строки α .

$After_G(N)$ — это множество всех терминалов, которые могут встретиться в строках, выводимые в грамматике G из начального нетерминала S непосредственно справа от нетерминала N .

LL(k)-ГРАММАТИКИ

Пример. Пусть задана КС-грамматика $G = (\Sigma, \mathcal{N}, P, S)$ состоит из правил

$$\begin{array}{lcl} S & \to & AcSdA \mid BdScB \\ A & \to & aA \mid \varepsilon \\ B & \to & bB \mid \varepsilon \end{array}$$

LL(k)-ГРАММАТИКИ

Пример. Пусть задана КС-грамматика $G = (\Sigma, \mathcal{N}, P, S)$ состоит из правил

$$
\begin{aligned}
S &\rightarrow AcSdA \mid BdScB \\
A &\rightarrow aA \mid \varepsilon \\
B &\rightarrow bB \mid \varepsilon
\end{aligned}
$$

Тогда

$$
\begin{array}{lcllcl}
First_G(AcSdB) & = & \{a, c\}; & First_G(BdScB) & = & \{b, d\}; \\
After_G(S) & = & \{d, c\}; & & & \\
First_G(aA) & = & \{a\}; & After_G(A) & = & \{c\}; \\
First_G(bB) & = & \{b\}; & After_G(B) & = & \{d\};
\end{array}
$$

LL(k)-ГРАММАТИКИ

Для каждого правила грамматического вывода $N \to \alpha$ положим

$$Dir_G(N \to \alpha) = \begin{cases} First_G(\alpha) \cup After_G(N), & \text{если } \alpha \xrightarrow{G}_{*} \varepsilon, \\ First_G(\alpha), & \text{иначе }. \end{cases}$$

LL(k)-ГРАММАТИКИ

Для каждого правила грамматического вывода $N \to \alpha$ положим

$$Dir_G(N \to \alpha) = \begin{cases} First_G(\alpha) \cup After_G(N), \text{ если } \alpha \xrightarrow{G}_* \varepsilon, \\ First_G(\alpha), \text{ иначе }. \end{cases}$$

Теорема 9.10. КС-грамматика G является LL(1)-грамматикой тогда и только тогда, когда для любого нетерминала N и для любой пары правил $N \to \alpha_1$ и $N \to \alpha_2$ верно соотношение

$$Dir_G(N \to \alpha_1) \cap Dir_G(N \to \alpha_2) = \varnothing.$$

Доказательство. Следует непосредственно из определения LL(1)-грамматики. QED

LL(k)-ГРАММАТИКИ

Пример. Пусть КС-грамматика $G = (\Sigma, \mathcal{N}, P, S)$ состоит из правил

$$
\begin{aligned}
S &\to AcSdA \\
S &\to BdScB \\
A &\to aA \\
A &\to \varepsilon \\
B &\to bB \\
B &\to \varepsilon
\end{aligned}
$$

LL(k)-ГРАММАТИКИ

Пример. Пусть КС-грамматика $G = (\Sigma, \mathcal{N}, P, S)$ состоит из правил

$$
\begin{array}{llll}
S & \to & AcSdA & Dir_G = \{a, c\}; \\
S & \to & BdScB & Dir_G = \{b, d\}; \\
A & \to & aA & Dir_G = \{a\}; \\
A & \to & \varepsilon & Dir_G = \{c\}; \\
B & \to & bB & Dir_G = \{b\}; \\
B & \to & \varepsilon & Dir_G = \{d\};
\end{array}
$$

Значит, G — LL(1)-грамматика.

LL(k)-ГРАММАТИКИ

Множества $Dir_G(N \to \alpha)$ можно вычислить итерационным алгоритмом сравнительно просто за время $O(|\alpha||G|)$.

Задача 8. Построить алгоритмы для вычисления множеств $First_G(\alpha)$ и $After_G(N)$.

LL(k)-ГРАММАТИКИ

Множества $Dir_G(N \to \alpha)$ можно вычислить итерационным алгоритмом сравнительно просто за время $O(|\alpha||G|)$.

Задача 8. Построить алгоритмы для вычисления множеств $First_G(\alpha)$ и $After_G(N)$.

Если грамматика G является LL(1)-грамматикой, то располагая множествами $Dir_G(N \to \alpha)$ для каждого правила, нетрудно построить и функцию $Tab(x, N)$ осуществляющую однозначный выбор правил синтаксического разбора.

LL(k)-ГРАММАТИКИ

$$Tab_G(N, x) = \begin{cases} N \to \alpha, \text{ если } x \in Dir_G(N \to \alpha), \\ \textbf{error}, \text{ если } x \notin Dir_G(N \to \beta) \\ \qquad\qquad \text{для всех правил } N \to \beta \in P \,. \end{cases}$$

LL(k)-ГРАММАТИКИ

$$Tab_G(N, x) = \begin{cases} N \to \alpha, \text{ если } x \in Dir_G(N \to \alpha), \\ \textbf{error}, \text{ если } x \notin Dir_G(N \to \beta) \\ \qquad\qquad \text{для всех правил } N \to \beta \in P \ . \end{cases}$$

Таким образом, справедлива

Теорема 9.11. Если КС-грамматика G является LL(1)-грамматикой, то $L(G)$ — детерминированный КС-язык.

КОНТЕКСТНО-ЗАВИСИМЫЕ ГРАММАТИКИ

Правила вывода контекстно-зависимой (КЗ) грамматики $G = (\Sigma, \mathcal{N}, P, S)$ имеют вид $\theta N \eta \to \theta \alpha \eta$, где $N \in \mathcal{N}$, $\eta, \theta \in (\Sigma \cup \mathcal{N})^*$, $\alpha \in (\Sigma \cup \mathcal{N})^+$. Т.к. $|\alpha| > 0$, КЗ-грамматики также называют нестираемыми грамматиками.

Контекстно-зависимые грамматики

Правила вывода контекстно-зависимой (КЗ) грамматики $G = (\Sigma, \mathcal{N}, P, S)$ имеют вид $\theta N \eta \to \theta \alpha \eta$, где $N \in \mathcal{N}$, $\eta, \theta \in (\Sigma \cup \mathcal{N})^*$, $\alpha \in (\Sigma \cup \mathcal{N})^+$. Т.к. $|\alpha| > 0$, КЗ-грамматики также называют нестираемыми грамматиками.

Утверждение 9.12. Если L — КС-язык и $\varepsilon \notin L$, то L — КЗ-язык.

Доказательство. Следует из Утверждения 7.11 об устранении ε-правил из КС-грамматик.

При этом некоторые КЗ-языки (например, $\{a^n b^n c^n : n \geq 0\}$ и $\{ww : w \in \{a, b\}^*\}$) не являются КС-языками.

КОНТЕКСТНО-ЗАВИСИМЫЕ ГРАММАТИКИ

Утверждение 9.13. Проблема включения $w \in L(G)$ для КЗ-грамматик алгоритмически разрешима.

Доказательство. Это следует из свойства «нестираемости» Попробуйте самостоятельно .

Контекстно-зависимые грамматики

Утверждение 9.13. Проблема включения $w \in L(G)$ для КЗ-грамматик алгоритмически разрешима.

Доказательство. Это следует из свойства «нестираемости» Попробуйте самостоятельно .

Утверждение 9.14. Класс КЗ-языков замкнут относительно операций объединения, пересечения, дополнения, конкатенации.

Казалось бы, КЗ-грамматики весьма перспективны для практического использования!

КОНТЕКСТНО-ЗАВИСИМЫЕ ГРАММАТИКИ

Машина Тьюринга M называется линейно-ограниченной, если для любого ленточного слова w и для любой конфигурации α верно $q_1 w0 \xrightarrow{M}_* \alpha \implies |\alpha| = |q_1 w0|$.

КОНТЕКСТНО-ЗАВИСИМЫЕ ГРАММАТИКИ

Машина Тьюринга M называется линейно-ограниченной, если для любого ленточного слова w и для любой конфигурации α верно $q_1 w0 \xrightarrow{M}_* \alpha \implies |\alpha| = |q_1 w0|$.

Утверждение 9.15. Для любой КЗ-грамматики G можно построить такую линейно-ограниченная МТ M_G, что $L(G) = L(M_G)$. Для любой линейно-ограниченной МТ M_G можно построить такую КЗ-грамматики G_M, что $L(M) = L(G_M)$.

Следствие. Проблема включения $w \in L(G)$ для КЗ-грамматик PSPACE-полна.

КОНТЕКСТНО-ЗАВИСИМЫЕ ГРАММАТИКИ

Задача 9 [Трудная]. Доказите, что проблема пустоты $L(G) = \varnothing$ для КЗ-грамматик алгоритмически неразрешима.

КОНТЕКСТНО-ЗАВИСИМЫЕ ГРАММАТИКИ

Задача 9 [Трудная]. Докажите, что проблема пустоты $L(G) = \varnothing$ для КЗ-грамматик алгоритмически неразрешима.

Это означает, что задача синтаксического анализа для КЗ-грамматик непроста. Но именно КЗ-грамматики позволяют описывать синтаксис естественных языков. Поэтому интересны классы языков, занимающих промежуточное положение между КС и КЗ-языками.

Так, например, студент 3-го курса ф-та ВМК МГУ Александр Охотин в 2002 г. придумал новый тип грамматик — конъюнктивные грамматики , — обобщающий класс КС-грамматик.

Контекстно-зависимые грамматики

А есть ли в этой области нерешенные задачи?

КОНТЕКСТНО-ЗАВИСИМЫЕ ГРАММАТИКИ

А есть ли в этой области нерешенные задачи? Вот одна из них.

Как известно, проблема эквивалентности неразрешима для недетерминированных магазинных автоматов. Но в 1997 г. было доказано, что эта проблема разрешима для детерминированных магазинных автоматов.

Открытая проблема. Какова сложность проблемы эквивалентности для детерминированных магазинных автоматов?

ТЕХНОЛОГИЯ ПРИМЕНЕНИЯ

Вам поручили разработать синтаксический анализатор искусственного языка

ТЕХНОЛОГИЯ ПРИМЕНЕНИЯ

Вам поручили разработать
синтаксический анализатор искусственного языка

Имеет ли грамматика
этого языка тип LL(k)?

ТЕХНОЛОГИЯ ПРИМЕНЕНИЯ

Вам поручили разработать синтаксический анализатор искусственного языка

Имеет ли грамматика этого языка тип LL(k)?

ДА

Постройте функцию $Tab(x, N)$

ПОЗДРАВЛЯЕМ!

Вы получили быстрый синтаксический анализатор!

ТЕХНОЛОГИЯ ПРИМЕНЕНИЯ

Вам поручили разработать
синтаксический анализатор искусственного языка

Имеет ли грамматика
этого языка тип LL(k)?

НЕТ

Постарайтесь привести
правила грамматики
к форме LL(k)

ТЕХНОЛОГИЯ ПРИМЕНЕНИЯ

Вам поручили разработать
синтаксический анализатор искусственного языка

Имеет ли грамматика
этого языка тип LL(k)?

НЕТ

Постарайтесь привести
правила грамматики
к форме LL(k)

УДАЛОСЬ!

Постройте функцию
$Tab(x, N)$

ПОЗДРАВЛЯЕМ!

Вы получили быстрый
синтаксический анализатор!

ТЕХНОЛОГИЯ ПРИМЕНЕНИЯ

Вам поручили разработать
синтаксический анализатор искусственного языка

Имеет ли грамматика
этого языка тип LL(k)?

НЕТ

Постарайтесь привести
правила грамматики
к форме LL(k)

НЕ УДАЛОСЬ

Воспользуйтесь
алгоритмом CKY

Увы, Ваш синтаксический анализатор
иногда будет работать очень медленно.

КОНЕЦ ЛЕКЦИИ 9