EURASIAN JOURNAL OF ACADEMIC RESEARCH
Innovative Academy Research Support Center
UIF = 8.1 | SJIF = 5.685 www.in-academy.uz

СВОЙСТВА ЗАМКНУТОСТИ КЛАССА КС-ЯЗЫКОВ

Ибрагимов Улугбек Мурадиллоевич
Бухарский инженерно-технологический институт, доцент
Имомов Бекзод марат ўғли
Бухарский инженерно-технологический институт, стажер-перподаватель

https://www.doi.org/10.5281/zenodo.10069299

ARTICLE INFO

Received: 25th October 2023
Accepted: 30th October 2023
Online: 31th October 2023

KEY WORDS

Контекстно-свободные языки, цепочка языка, гомоморфизм, грамматика, терминалы.

ABSTRACT

Свойства замкнутости часто помогает доказать, что некоторые языки не контекстно-свободны; кроме того, они интересны и с теоретической точки зрения. В этой статье мы приведем несколько основных свойств замкнутости класса КС-языков.

Пусть К-класс языков и язык $L \subseteq \sum^*$ принадлежит К. Допустим, что $\sum = \{a_1, a_2, .., a_n\}$ и языки $L_{a_1}, \dots, L_{a_n}$ принадлежат К. Класс К замкнут относительно подстановки, если для любого набора языков $L, L_{a_1}, \dots, L_{a_n}$язык

$$L' = \left\{x_1 \dots x_k | a_{j_1} \dots a_{j_k} \in L_{a_{j_1}}, \dots, x_k \in L_{a_{j_k}}\right\}$$

принадлежит К.

Рассмотирм пример: Пусть $L = \{0^n 1^n | n \geq 1\}, L_0 = \{a\}$ и $L_1 = \{b^m c^m | m \geq 1\}$. Тогда результатом подстановки языков L_0 и L_1 будет язык

$$L^1 = \{a^n b^{m_1} c^{m_1} b^{m_2} c^{m_2} \dots b^{m_n} c^{m_n} | n \geq 1, m_i \geq 1\}$$

Класс КС-языков замкнут относительно подстановки. Пусть $L \sqsubseteq \sum^*$ КС – язык и $\sum = \{a_1, a_2, .., a_n\}$. Пусть $L_a \sqsubseteq \sum_a^*$ - КС-язык для каждого $a \in \sum$ и L' – результат подстановки языков L_a вместо a в цепочке языка L. Пусть $G = (N, \sum, P, S)$ – КС - грамматика языка L и $G_a = (N_a, \sum_a, P_a, a')$ КС грамматика языка L_a. Предполагаем, что N и все N_a попарно не пересекаются. Возьмем G'=(N', ∑', P', S), где

(1) N' $=\cup_{a \epsilon \sum} N_a \cup N$;

(2) $\sum' = \cup_{a \epsilon \sum} \sum_a$

(3) пусть h-гомоморфизм, определенный на $N \cup \sum$ и такой, что h(A)=A для всех A∈N и h(a)=a' для a∈ ∑; положим $P' = \{A \to h(a) | A \to a \in P\}\ U \cup_{a \epsilon \sum} P_a$

Итак, P' состоит из правил всех грамматик G_a и правил грамматики G, в которых все терминалы сделаны нетерминалами со штрихами. Пусть $a_{j_i} \dots\ a_{j_k} \in L$ н $x_i \in L_{a_{j_i}}$ для $1 \leq i \leq k$. Тогда

$S \Rightarrow \dot{G}' a'_{j_i} \dots a'_{j_k} \Rightarrow \dot{G}' x_1 a'_{j_z} \dots a'_{j_k} \Rightarrow \dot{G}' \dots \Rightarrow \dot{G}' x_1 \dots x_k$.

Следовательно, $L' \subseteq L(G')$

Допустим, что $\omega \in L(G')$, и рассмотрим дерево вывода T цепочки ω. Так как N и N_a не пересекаются, каждый лист с меткой, отличной от e, имеет по крайней мере одного предка, помеченного $a^{\wedge\prime}$, где $a \in \sum$. Если удалить из T каждую вершипу. у которой есть предок, отличный от нее и помеченный a' для $a \in \sum$., то получим дерево вывода T' с кропой $a'_{j_i} \dots a'_{j_k}$, где $a'_{j_i} \dots a'_{j_k} \in L$. Если x_i-крона поддерева дерева T, над которыми доминирует i-й лист дерева T', то $\omega = x_1 \dots x_k$ и $x_i \epsilon L_{a_{j_i}}$ Таким образом, L(G') ⊆ L' (G).

Класс КС-языков замкнут относительно (1) объединения, (2) конкатенации, (3) итерации, (4) позитивной итерации и (5) гомоморфизма.

Доказательство. Пусть L_a и L_b-КС-языки.

(1) Подставим L_a вместо a и L_b вместо b в КС-язык {a, b}.

(2) Подставим L_a вместо a и L_b вместо b в {a b}.

(3) Подставим L_a вместо a в a*.

(4) Подставим L_a вместо a в a^+.

(5) Для гомоморфизма h возьмем L_a = {h(a)} и, подставив L_a вместо a в L, получим h (L).

Класс КС-языков замкнут относительно пересечения с регулярными множествами.

Нетрудно показать, что МП-автомат P и параллельно работающий конечный автомат A можно моделировать подходящим МП-автоматом P′. Этот составной МП-автомат P′ прямо моделирует P и изменяет состояние автомата A каждый раз, когда P делает не e-такт. P' допускает цепочку тогда и только тогда, когда ее допускает P и A находится при этом в заключительном состоянии.

В отличие от регулярных множеств КС-языки не образуют булеву алгебру множеств.

Существует много операций, относительно которых замкнут класс КС-языков.

Пример 1. L ={$\omega\omega|\omega\epsilon$ $\{a, b\}$+} не КС-язык. Допустим, что это не так. Тогда по теореме 2.26 язык $L' = L \cap a^+ b^+ a^+ b^+ = \{a^m b^n a^m b^n | m, n \geq 1\}$ контекстно-свободей. Но в упр. 2.6.3 (д) утверждается, что L' не КС-язык.

Пример 2. L ={$\omega\omega|\omega\epsilon$ $\{c, f\}$+} не КС-язык. Пусть h-такой гомоморфизм, что h(c)=a и h(f) = b. Тогда h(L) ={$\omega\omega|\omega\epsilon$ $\{a, b\}$+} не КС-язык (см. предыдущий пример). Так как класс КС-языков замкнут относительно гомоморфизмов, то L не КС-язык.

Пример 3. Алгол не является КС-языком. Рассмотрим следующий класс программ Алгола:

$$L = \{\boldsymbol{begin\ integer}\ \omega;\ \omega := 1;\ \boldsymbol{end} | \omega \in \{c, f\}^+\}$$

Пусть L_A-множество всех правильных программ Алгола и R-регулярное множество, обозначаемое регулярным выражением

begin integer $(c+f)^+$; $(c+f)^+$: = 1; **end**

Тогда $L = L_A \cap R$. Наконец, пусть h-такой гомоморфизм, что h(c)=c, h(f)=f и h(X)=e в остальных случаях. Тогда h(L) ={$\omega\omega|\omega\epsilon$ $\{c, f\}$+}

Следовательно, если L_A—КС—язык, то $h(L_A \cap R)$ —тоже КС—язык. Однако мы знаем, что язык $h(L_A \cap R)$ не контекстно-свободен, и поэтому заключаем, что множество L_A всех правильных программ Алгола не является КС-языком.

Пример 4. показывает, что язык программирования, требующий описания идентификаторов, длина которых может быть произвольно большой, не контекстно-свободен. Однако в компиляторе идентификаторы обычно обрабатываются лексическим анализатором и свертываются в лексемы прежде, чем достигают синтаксического анализатора. Поэтому язык, который должен распознаваться синтаксическим анализатором, обычно можно считать контекстно-свободным.

В Алголе и многих других языках встречаются и другие явления, характерные для не контекстно-свободных языков. Например, каждая процедура имеет одно и то же число аргументов в каждом месте, где она упоминается. Поэтому можно показать, что подаваемый на вход синтаксического анализатора язык не контекстно-свободен, отобразив множество программ с тремя вызовами одной и той же процедуры на не контекстно- свободный язык $\{0^n|0^n|0^n|n \geq 0$. Обычно, однако, для проверки того, что число аргументов процедуры не противоречит ее определению, используется некогорый процесс, не входящий в собственно синтаксический анализ.

Рассмотрим проблему принадлежности для ҚС-грамматик, Нам надо найти алгоритм, который по данной КС-грамматике G=(N, ∑, P, S) и цепочке $\omega \in \sum^*$ выясняет, принадлежит ли ω языку L(G). Получение эффективного алгоритма, решающего эту проблему. С чисто теоретической точки зрения можно сразу сказать, что проблема принадлежности для КС-грамматик разрешима: с помощью преобразований, всегда можно превратить G в эквивалентную приведенную КС-грамматику G', а приведенные КС-грамматики (без учета пустой цепочки) контекстно-зависимы, так что можно применить общий грубый алгоритм, решающий проблему принадлежности.

Рассмотрим проблему эквивалентности для КС-грамматик. К сожалению, эта проблема неразрешима. Мы докажем, что не существует алгоритма, который по двум данным КС-грамматикам G_1 и G_2 мог бы определить, совпадают ли языки L(G_1) и L(G_2). На самом деле будет доказано, что не существует даже алгоритма, который по КС-грамматике G_1 и праволинейной грамматике G_2 выяснял бы, выполняется ли равенство L(G_1) =L(G_2). Как и для большинства других неразрешимых проблем, мы покажем, что если бы проблема эквивалентности для КС-грамматик была разрешима, то была бы разрешима и проблема соответствий Поста. По частному случаю проблемы соответствий мы будем строить два естественно связанных с ним КС-языка.

Пусть C= (x_1, y_1), ..., (x_n, y_n) —частный случай проблемы соответствий над алфавитом $\sum$ и $I = \{1,2, \ldots, n\}, I \cap \sum = \emptyset$. Положим $L_C = \{x_{i_1} x_{i_2} \ldots x_{i_m} i_m i_{m-1} \ldots i_1 | i_1, \ldots, i_m \in I, m \geq 1\}$ и $M_C = \{y_{i_1} y_{i_2} \ldots x_y i_m i_{m-1} \ldots i_1 | i_1, \ldots, i_m \in I, m \geq 1\}$

Пусть C= (x_1, y_1), ..., (x_n, y_n)—частный случай проблемы соответствий над алфавитом $\sum$, где $\sum \cap \{1,2 \ldots, n\} = \emptyset$. Тогда

(1) можно найти расширенные ДМП-автоматы, допускающие L_C и M_C,

(2) $L_C \cap M_C = \emptyset$ тогда и только тогда, когда C не имеет решения.

Легко построить расширенный ДМП-автомат (верхняя часть его магазина расположена справа), который все входные символы, принадлежащие ∑, переносит в магазин, я когда на входе появляются символы из {1, 2, ..., n}, он удаляет из магазина x_i , если на входе появилось число i. Если на верху магазина в этот момент не x_i , то ДМП-автомат останавливается. Кроме того, с помощью управляющего устройства с конечной памятью он проверяет, принадлежит ли входная цепочка множеству $\sum^+\{1, ..., n\}^+$, и допускает ее, если все символы множества ∑ удалены из магазина. Таким образом допускается L_C. Аналогично можно найти расширенный ДМП-автомат, допускающий M_C.

Если язык $L_C \cap M_C$ содержит цепочку $\omega i_m \dots i_1$, где $\omega \epsilon \sum^+$, то ясно, что ω- решающая последовательцость. Если $x_{i_1} \dots x_{i_m} = y_{i_1} \dots y_{i_m} = \omega$, то $\omega i_m \dots i_1 \in L_C \cap M_C$.

Вернемся к проблеме эквивалентности для КС-грамматик. Нам понадобятся еще два языка, связанные с частным случаем проблемы соответствий.

Пусть С= (x_1, y_1), ..., (x_n, y_n)—частный случай проблемы соответствий над алфавитом $\sum$ и I={1, ..., n), причем $\sum \cap I = \emptyset$. Положим $Q_C = \{\omega \# \omega^R | \omega \in \sum^+ I^+\}$, где $\# \notin \sum \cup I$ и $P_C = L_C \# M_C^R$.

Пусть С —определенная выше последовательность. Тогда (1) можно найти расширенные ДМП-автоматы, допускающие Q_C и P_C, (2) $Q_C \cap P_C = \emptyset$ тогда и только тогда, когда С не имеет решения.

ДМП-автомат, допускающий Q_C, построить легко. Что касается P_C, то в силу леммы существует ДМП—автомат, скажем M_1, допускающий L_C. Найти ДМП-автомат M_2, допускающий M_C^R, не намного труднее; он переносит со входа в магазин символы из I и затем сравнивает их с той частью входной цепочнн, которая принадлежит $\sum^+$. Поэтому можио построить ДМН-автомат M_3, моделирующий M_1, потом проверяющий, есть ли #, и, наконец, моделирующий M_2.

(2) Если $uv\#\omega x \in Q_C \cap P_C$, где и, $x \in \sum^+$ и $v, \omega \in I^+$, то $u = x^R$ и $v = \omega^R$, поскольку $uv\#\omega x \in Q_C$. С другой стороны, u-решающая последовательность, так как $uv\#\omega x \in P_C$. Таким образом, С имеет решение. Обратно, если $x_{i_1} \dots x_{i_m} = y_{i_1} \dots y_{i_m}$, то $x_{i_1} \dots x_{i_m} i_m \dots i_1 \# i_1 \dots i_m x_{i_m}^R \dots x_{i_1}^R \in Q_C \cap P_C$.

Пусть С—определенная выше последовательность. Тогда (1) можно найти ҚС-грамматику для языка $\overline{Q_C} \cup \overline{P_C}$ (2) $\overline{Q_C} \cup \overline{P_C} = (\sum \cup I)^*$ тогда и только тогда, когда С не имеет решения.

В силу замкнутости класса детерминированных КС-языков относительно дополнения (теорема 2.23) можно найти ДМП-автоматы для языков $\overline{Q_C}$ и $\overline{P_C}$. В силу эквивалентности КС-грамматик и МІ-автоматов (лемма 2.26) можно найти для этих языков КС-грамматики. В силу замкнутости класса КС-языков относительно объединения можно найти КС грамматику для $\overline{Q_C} \cup \overline{P_C}$ (2) Непосредственное следствие леммы 2.30(2) и закона де Моргана.

Не существует алгоритма, который для КС-грамматики G_1 и праволинейной грамматики G_2 отвечал бы на вопрос: „L (G_1) = L (G_2)?".

Если бы такой алгоритм существовал, то можно было бы следующим образом решать проблему соответствий Поста: (1) По данному частному случаю С построить КС-грамматику G_1, порождающую $\overline{Q_C} \cup \overline{P_C}$ (по лемме 2.31), и праволинейную грамматику G_2, порождающую регулярное множество $(\sum \cup I)^*$, где $\sum$-алфавит последовательности С, n-ее длина и I= {1, ...,n). Возможно, придется сначала переименовать некоторые символы, но это не повлияет на наличие или отсутствие решения.

(2) С помощью гипотетического алгоритма получить ответ па вопрос: „L(G_1) = L(G_2)?" По лемме 2.31 (2) это равенство выполняется тогда и только тогда, когда С не имеет решения.

Так как существуют алгоритмы, преобразующие КС-грамматику в эквивалентный МП-автомат и обратно, то из теоремы следует также неразрешимость таких проблем: распознают ли два МП-автомата (или МП автомат конечный автомат) один и тот же язык; распознает ли данный МП-автомат множество, обозначаемое данным регулярным выражением, и т. д.

References:

1. Ало А., Сети Р., Ульман Дж. Компиляторы: принципы, технологии и инструменты: Пер. с анг. - М.: Издательский дом «Вильямс», 2003. - 768 с.

2. Гордеев А. В., Молчанов А. Ю. Системное программное обеспечение. - СПб. Питер, 2002. - 734 с.

3.Компанеец Р. И., Мальков Е. В., Филатов Н. Е. Системное программирование. Основы построения трансляторов: Учеб, пособие для высших в средних, учебных заведений. - СПб.: КОРОНА принт, 2000. - 256 с.

4. Шоймардонов, Ж. З. Применение классификаторов машинного обучения при определении качества сигнала в параметрических технологических процессах / Ж. З. Шоймардонов ; науч. рук. У. М.Ибрагимов // Молодость. Интеллект. Инициатива : материалы XI. Международной научно-практической конференции студентов и магистрантов, Витебск, 21 апреля 2023 года : в 2 т. - Витебск : ВГУ имени П. М. Машерова, 2023. - Т. 1. - С. 80-81. - Библиогр.: с. 81 (4 назв.).

5. Sh.T. Shukurov, U.M. Ibragimov. Architecture for building the systems of storage and analysis of BIG DATA. "Economics and society" №5(96 2022(www.iupr.ru).

6. Султонова, М., & Ибрагимов, У. (2022). ПРИМУЩЕСТВО ИСПОЛЬЗОВАНИЯ МАШИННОГО ОБУЧЕНИЯ ДЛЯ РАСПОЗНАВАНИЕ ТРАНСПОРТНЫХ СРЕДСТВ. International Bulletin of Applied Science and Technology, 2(11), 95–103.