

Контекстно-свободные языки и грамматики

- 1 Контекстно-свободные языки и грамматики
- 2 Преобразования КС-грамматик
- 3 Нормальные формы грамматик

Перейдем от рассмотрения регулярных языков к более широкому классу языков, которые называются контекстно-свободными. Они описываются в виде контекстно-свободных грамматик. Эти грамматики играют главную роль в технологии компиляции с начала 1960-х годов; они превратили непростую задачу реализации синтаксических анализаторов, распознающих структуру программы, из неформальной в рутинную, которую можно решить за один вечер.

Позже контекстно-свободные грамматики стали использоваться для описания форматов документов в виде так называемых определений типа документов (document-type definition — DTD), которые применяются в языке XML (extensible markup language) для обмена информацией в Internet.

В контекстно-свободных грамматиках важным является понятие дерева разбора, изображающего структуру, которую грамматика налагает на цепочки языка. Дерево разбора представляет собой выход синтаксического анализатора языка программирования и одновременно общепринятый способ выражения структуры программы.

Контекстно-свободные языки также описываются с помощью автоматов с магазинной памятью (иными словами автоматы с магазинной памятью являются распознавателями в КС-языках).

Примеры контекстно-свободных языков и грамматик

Рассмотрим следующий пример:

- $\epsilon, 0, 1$ являются палиндромами (базис рекурсии)
- Если w - палиндром, то $0w0$ и $1w1$ - также палиндром (шаг рекурсии). Ни одна цепочка не является палиндромом, если не определяется базисом или шагом рекурсии.

Например, такие цепочки $0110, 11011$ являются палиндромами, а $0011, 0101$ - нет. Воспользуемся леммой о накачке, чтобы показать, что данный язык не является регулярным.

Если бы данный язык был регулярным, то существовала бы константа n , начиная с которой любую цепочку длины больше n можно было бы накачивать. Рассмотрим палиндром $w = 0^n 1 0^n$, представим $w = xyz$, где $xy = 0^n$, эту подцепочку уже можно накачать, в ней y состоит из одного или нескольких нулей, но тогда $0^{n-k}(y^k)^m 1 0^n$ - уже не палиндром.

Данный язык можно задать еще следующей порождающей грамматикой: $P \rightarrow \epsilon, P \rightarrow 0, P \rightarrow 1, P \rightarrow 0P0, P \rightarrow 1P1$. Эта запись по сути повторяет рекурсивное определение. В целом контекстно-свободная грамматика представляет собой формальную запись подобных рекурсивных определений языков. Грамматика состоит из одной или нескольких переменных, которые представляют классы цепочек, или языки. В данном примере нужна только одна переменная, представляющая множество палиндромов.

Устранение бесплодных символов

Определение

Нетерминальный символ является бесплодным, если из него нельзя вывести ни одной цепочки терминальных символов.

В простейшем случае символ является бесплодным, если во всех правилах, где этот символ стоит в левой части, он также встречается и в правой части. Более сложные варианты предполагают зависимости между цепочками бесплодных символом, когда они в любой последовательности вывода порождают друг друга.

Алгоритм устранения бесплодных символов прост: 1. мы начинаем с пустого множества нетерминальных символов $Y = \emptyset$. 2. добавляем в множество Y только такие нетерминалы A , для которых имеем $A \rightarrow \beta$, где β состоит из одних терминальных символов. 3. последовательно пополняем множество Y нетерминальными символами B , из которых следует цепочки терминальных символом, а также символы самого множества Y . 4. выполняем п.3 пока множество Y изменяется.

Определение

Символ называется недостижимым, если он не участвует ни в одной цепочке вывода из целевого символа грамматики.

Очевидно, недостижимые символы можно удалить. Алгоритм удаления недостижимых символов строит множество достижимых символов X . Первоначально в такое множество входит только целевой символ S , затем оно пополняется на основе правил грамматики. Все символы, которые не войдут в данное множество, являются недостижимыми и могут быть исключены в новой грамматике.

Пример устранения бесплодных и недостижимых символов

Пусть грамматика задана правилами:

$$S \rightarrow aAB|E$$

$$A \rightarrow aA|bB$$

$$B \rightarrow ACb|b$$

$$C \rightarrow A|bA|cC|aE$$

$$E \rightarrow cE|aE|Eb|ED|FG$$

$$D \rightarrow a|c|Fb$$

$$F \rightarrow BC|EC|AC$$

$$G \rightarrow Ga|Gb$$

Важно соблюдать последовательность преобразования: сначала удаляются бесплодные символы, потом недостижимые.

Пример устранения бесплодных и недостижимых символов

Удаляем бесплодные символы

- 1 $Y_1 = \{B, D\}$
- 2 $Y_2 = \{B, D, A\}$
- 3 $Y_3 = \{B, D, A, S, C\}$
- 4 $Y_4 = \{B, D, A, S, C, F\}$
- 5 $Y_5 = Y_4$

Окончательно в множестве продукций оставляем только те символы, которые включает в себя множество Y_4 :

$$\begin{aligned}S &\rightarrow aAB \\A &\rightarrow aA|bB \\B &\rightarrow ACb|b \\C &\rightarrow A|bA|cC \\D &\rightarrow a|c|Fb \\F &\rightarrow BC|AC\end{aligned}$$

Пример устранения бесплодных и недостижимых символов

Удаляем недостижимые символы

Итак, теперь из грамматики удалим недостижимые символы:

$$S \rightarrow aAB, A \rightarrow aA|bB, B \rightarrow ACb|b$$

$$C \rightarrow A|bA|cC, D \rightarrow a|c|Fb, F \rightarrow BC|AC$$

- 1 $X_1 = \{S\}$
- 2 $X_2 = \{S, A, B\}$
- 3 $X_3 = \{S, A, B, C\}$
- 4 $X_4 = X_3$

Окончательно оставляем в грамматике только правила, включающие в себя достижимые символы:

$$S \rightarrow aAB$$

$$A \rightarrow aA|bB$$

$$B \rightarrow ACb|b$$

$$C \rightarrow A|bA|cC$$

Неукорачивающиеся и ϵ -свободные грамматики

Определение

Грамматика, которая задается общим правилом $A \rightarrow \beta$, где A - один нетерминальный символ, β - цепочка терминальных и нетерминальных символов, называется контекстно-свободной грамматикой (КС-грамматика).

Определение

КС-грамматика называется неукорачивающей, если она не включает продукции вида $A \rightarrow \epsilon$.

Определение

КС-грамматика называется ϵ -свободной, если она неукорачивающая или в ней существует ровно одна продукция вида $S \rightarrow \epsilon$, где S - начальный нетерминал, и S не встречается в правой части ни одной из продукций.

Очевидно, что любой шаг вывода в неукорачивающей грамматике не может уменьшить длину выводимой цепочки - отсюда и ее название.

Устранение ϵ -правил

Теорема

По любой КС-грамматике может быть построена эквивалентная ϵ -свободная КС-грамматика. По любой КС-грамматике, порождающей язык, не включающий пустую цепочку, может быть построена эквивалентная неукорачивающая грамматика.

Доказательство.

Для любой пустой продукции $A \rightarrow \epsilon$ в грамматике найдем и скопируем все продукции, включающие A в правой части, с выбрасыванием символа A из правых частей этих копий, причем если A входит несколько раз в правую часть продукции, то такое выбрасывание производится во всех возможных сочетаниях. Далее продукция $A \rightarrow \epsilon$ выбрасывается из множества правил. Если A - начальный нетерминал, то выбираем новый начальный нетерминал S и к продукциям грамматики добавляем две новые: $S \rightarrow \epsilon$, $S \rightarrow A$. □

Пример устранения ϵ правил

Пусть задана грамматика начальным нетерминалом A :

$$A \rightarrow \epsilon | AaB | BbB$$

$$B \rightarrow Ba | \epsilon$$

Вводим новый символ $S \rightarrow \epsilon | A$. Удаляем A в правой части продукций во всех сочетаниях: $A \rightarrow AaB | aB | BbB$. Удаляем B в правой части продукций: $B \rightarrow Ba | a$, $A \rightarrow AaB | Aa | aB | a | BbB | bB | Bb | b$.

Окончательно имеем:

$$S \rightarrow \epsilon | A$$

$$A \rightarrow AaB | Aa | aB | a | BbB | bB | Bb | b$$

$$B \rightarrow Ba | a$$

Определение

Продукции вида $A \rightarrow B$ называются сингулярными.

Если в результате цепного применения правил вида $A \rightarrow B$ приходим к выводу, что из $A \Rightarrow A$, то такой вывод называется циклическим.

Любую грамматику можно преобразовать в эквивалентную ей грамматику без сингулярных продукций и циклических выводов. Для этого вместо каждой сингулярной продукции вида $A \rightarrow B$ включаем продукции $A \rightarrow \beta$, такие, что $B \rightarrow \beta$ несингулярна.

Пример устранения циклических символов

Пример 1.

$$S \rightarrow BA$$

$$A \rightarrow C|ac$$

$$B \rightarrow b$$

$$C \rightarrow A$$

В данной грамматике есть две сингулярные продукции: $A \rightarrow C, C \rightarrow A$, из которых фактически следует циклический вывод $A \Rightarrow A$. Заменяем продукцию $C \rightarrow A$ на несингулярную $A \rightarrow \beta: C \rightarrow ac$. Окончательно получаем:

$$S \rightarrow BA$$

$$A \rightarrow ac$$

$$B \rightarrow b$$

$$C \rightarrow ac$$

В результате такого приведения видно, что символ C не выводим (еще говорят недостижим), а значит продукция $C \rightarrow ac$ может быть вообще исключена.

Пример устранения циклических символов

Пример 2.

$$S \rightarrow S + T | S - T | T$$

$$T \rightarrow T * E | T / E | E$$

$$E \rightarrow (S) | a | b$$

Здесь мы видим две сингулярности: $S \rightarrow T$, $T \rightarrow E$. Поскольку для E имеем несингулярную продукцию $E \rightarrow (S) | a | b$, то вместо $T \rightarrow E$, можно написать $T \rightarrow (S) | a | b$, или в целом $T \rightarrow T * E | T / E | (S) | a | b$, которая теперь перестает быть сингулярной, а значит вместо $S \rightarrow T$ можно написать $S \rightarrow T * E | T / E | (S) | a | b$. Окончательно получаем:

$$S \rightarrow S + T | S - T | T * E | T / E | (S) | a | b$$

$$T \rightarrow T * E | T / E | (S) | a | b$$

$$E \rightarrow (S) | a | b$$

Устранение левой рекурсии

Определение

Нетерминальный символ A называется рекурсивным, если в грамматике существует вывод $A \Rightarrow \alpha A \beta$.

Определение

КС-грамматика называется леворекурсивной, если в ней существует вывод $A \Rightarrow A \beta$.

Определение

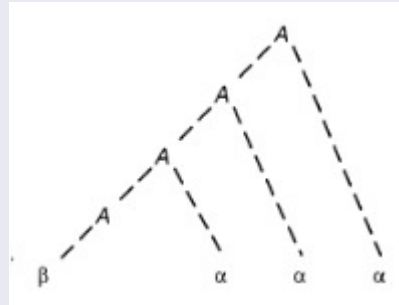
КС-грамматика называется праворекурсивной, если в ней существует вывод $A \Rightarrow \alpha A$.

Полностью исключить рекурсию в КС-грамматике нельзя, но можно полностью исключить либо левую, либо правую.

Многие алгоритмы синтаксического анализа не могут применяться к грамматикам с левой рекурсией, поэтому возникает необходимость ее устранения.

Устранение левой рекурсии

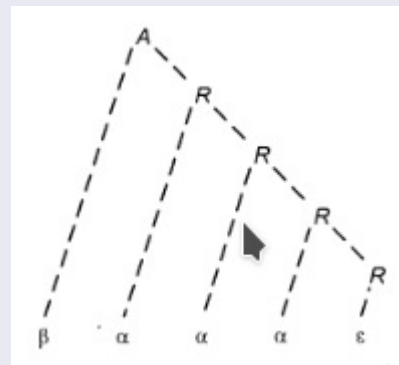
Рассмотрим простейший случай - пусть в грамматике есть продукции $A \rightarrow A\alpha|\beta$. Очевидно, что тогда в дереве вывода присутствует следующий фрагмент:



Тогда такие продукции можно заменить нелеворекурсивными:

$$A \rightarrow \beta R, R \rightarrow \alpha R|\epsilon$$

со следующим деревом вывода:



Пример

Рассмотрим правила грамматики арифметических выражений:

$$E \rightarrow E + T | T$$

Используя рассмотренный подход, получаем:

$$E \rightarrow TR$$

$$R \rightarrow +TR | \epsilon$$

Этот метод обобщается, когда нетерминал имеет несколько альтернатив с левой рекурсией. Более сложный случай непрямой рекурсии требует более сложного алгоритма.

Определение

Приведенные грамматики - это КС-грамматики, которые не содержат недостижимых, бесплодных символов, циклов и ϵ - правил.

Для того, чтобы преобразовать произвольную грамматику к приведенному виду, необходимо:

- 1 удалить все бесплодные символы;
- 2 удалить все недостижимые символы;
- 3 удалить все ϵ - правила;
- 4 удалить цепные правила

Все эти преобразования необходимо выполнять строго в указанном порядке.

Определение

КС-грамматика представлена в нормальной форме Хомского, если она неукорачивающая и каждое ее правило имеет одну из следующих форм:

$$A \rightarrow BC, A \rightarrow a$$

Грамматику $G = \{T, N, S, R\}$ можно привести к форме Хомского $G' = \{T', N', S, R'\}$ следующими преобразованиями:

- В множество продукций R' включаем все продукции вида $A \rightarrow BC, A \rightarrow a$.
- Для каждой продукции из R вида $A \rightarrow X_1X_2 \dots X_k$ включаем в R' множество продукций:

$$A \rightarrow X'_1 < X_2 \dots X_k >, < X_2 \dots X_k > \rightarrow X'_2 < X_3 \dots X_k >, \dots, \\ < X_{k-1} X_k > \rightarrow X'_{k-1} X'_k,$$

где $X'_i = X_i$, если X_i - нетерминал, иначе, если $X_i = a$, вводим новый нетерминал X'_i , а также правило $X'_i \rightarrow a$

$< X_2 \dots X_k >, < X_3 \dots X_k >, \dots < X_{k-1} X_k >$ - новые нетерминалы.

Пример

Пусть дана грамматика:

$$S \rightarrow ByA$$

$$A \rightarrow BS$$

$$A \rightarrow x$$

$$B \rightarrow zA$$

Вместо продукции $S \rightarrow ByA$ вводим $S \rightarrow BD, D \rightarrow EA, E \rightarrow y$

Вместо продукции $B \rightarrow zA$ вводим $B \rightarrow ZA, Z \rightarrow z$

Окончательно получаем следующую грамматику:

$$S \rightarrow BD$$

$$D \rightarrow EA$$

$$E \rightarrow y$$

$$A \rightarrow BS$$

$$A \rightarrow x$$

$$B \rightarrow ZA$$

$$Z \rightarrow z$$

Задание 1

Выбросить все бесполезные productions:

$$S \rightarrow aSbBc$$

$$B \rightarrow cD$$

$$S \rightarrow BdD$$

$$C \rightarrow bCc$$

$$A \rightarrow BcS$$

$$C \rightarrow cDA$$

$$A \rightarrow acb$$

$$D \rightarrow cSD$$

$$B \rightarrow bE$$

$$E \rightarrow ce$$

Задание 2

Построить ϵ -свободную грамматику.

$$S \rightarrow AaAb$$

$$S \rightarrow BbBa$$

$$A \rightarrow \epsilon$$

$$B \rightarrow \epsilon$$

Задание 3

По грамматике построить ациклическую грамматику:

$$S \rightarrow SS|(S)|\epsilon$$

Задание 4

Удалить левую рекурсию:

$$S \rightarrow SS|(S)|AaAb, A \rightarrow AbSa|ab$$

Задание 5

Преобразовать к нормальной форме Хомского:

$$S \rightarrow SbBc$$

$$B \rightarrow cD$$

$$S \rightarrow BdD$$

$$B \rightarrow b$$

$$D \rightarrow DBa$$

$$D \rightarrow c$$