Устранение левой рекурсии

Содержание

Определение:

Говорят, что контекстно-свободная (КС) грамматика Γ содержит **непосредственную левую рекурсию** (англ. *direct left recursion*), если она содержит правило вида $A \to A\alpha$.

Определение:

Говорят, что КС-грамматика Γ содержит **левую рекурсию** (англ. *left recursion*), если в ней существует вывод вида $A \Rightarrow^* A\alpha$.

Методы нисходящего разбора не в состоянии работать с леворекурсивными грамматиками. Проблема в том, что продукция вида $A \Rightarrow^* A\alpha$ может применяться бесконечно долго, так и не выработав некий терминальный символ, который можно было бы сравнить со строкой. Поэтому требуется преобразование грамматики, которое бы устранило левую рекурсию.

Устранение непосредственной левой рекурсии

Опишем процедуру, устраняющую все правила вида $A \to A\alpha$, для фиксированного нетерминала A.

1. Запишем все правила вывода из A в виде:
   $A \to A\alpha_1 \mid \ldots \mid A\alpha_n \mid \beta_1 \mid \ldots \mid \beta_m$, где
   - α — непустая последовательность терминалов и нетерминалов ($\alpha \nrightarrow \varepsilon$);
   - β — непустая последовательность терминалов и нетерминалов, не начинающаяся с A.
2. Заменим правила вывода из A на
   $A \to \beta_1 A' \mid \ldots \ \mid \beta_m A' \mid \beta_1 \mid \ldots \mid \beta_m$.
3. Создадим новый нетерминал $A' \to \alpha_1 A' \mid \ldots \mid \alpha_n A' \mid \alpha_1 \mid \ldots \mid \alpha_n$.

Изначально нетерминал A порождает строки вида $\beta\alpha_{i0}\alpha_{i1} \ldots \alpha_{ik}$. В новой грамматике нетерминал A порождает $\beta A'$, а A' порождает строки вида $\alpha_{i0}\alpha_{i1} \ldots \alpha_{ik}$. Из этого очевидно, что изначальная грамматика эквивалентна новой.

Пример

$$A \to S\alpha \mid A\alpha$$

$$S \to A\beta$$

Есть непосредственная левая рекурсия $A \to A\alpha$. Добавим нетерминал A' и добавим правила $A \to S\alpha A'$, $A' \to \alpha A'$.

Новая грамматика:

$$A \to S\alpha A' \mid S\alpha$$

$$A' \to \alpha A' \mid \alpha$$

$$S \to A\beta$$

В новой грамматике нет непосредственной левой рекурсии, но нетерминал A леворекурсивен, так как есть $A \Rightarrow S\alpha A' \Rightarrow A\beta\alpha A'$

Алгоритм устранения произвольной левой рекурсии

Воспользуемся алгоритмом удаления ε-правил. Получим грамматику без ε-правил для языка $L(\Gamma) \setminus \{\varepsilon\}$.

Упорядочим нетерминалы, например по возрастанию индексов, и будем добиваться того, чтобы не было правил вида $A_i \to A_j\alpha$, где $j \leqslant i$. Если данное условие выполняется для всех A_i, то в грамматике нет $A_i \Rightarrow^* A_i$, а значит не будет левой рекурсии.

Пусть $N = \{A_1, A_2, \ldots, A_n\}$ — упорядоченное множество всех нетерминалов.

```
for $A_i \in N$
  for $A_j \in \{N \mid 1 \leqslant j < i\}$
    for $p \in \{P \mid A_i \to A_j\gamma\}$
      удалить продукцию $p$
      for $Q \to x_i \in \{A_j \to \delta_1 \mid \ldots \mid \delta_k\}$
        добавить правило $A_i \to x_i\gamma$
  устранить непосредственную левую рекурсию для $A_i$
```

Если ε присутствовал в языке исходной грамматики, добавим новый начальный символ S' и правила $S' \to S \mid \varepsilon$.

После i итерации внешнего цикла в любой продукции внешнего цикла в любой продукции вида $A_k \to A_l\alpha, k < i$, должно быть $l > k$. В результате при следующей итерации внутреннего цикла растет нижний предел m всех продукций вида $A_i \to A_m\alpha$ до тех пор, пока не будет достигнуто $i \leqslant m$.

После i итерации внешнего цикла в грамматике будут только правила вида $A_i \to A_j\alpha$, где $j > i$. Можно заметить, что неравенство становится строгим только после применения алгоритма устранения непосредственной левой рекурсии. При этом добавляются новые нетерминалы. Пусть A'_i новый нетерминал. Можно заметить, что нет правила вида $\ldots \to A'_i$, где A'_i самый левый нетерминал, а значит новые нетерминалы можно не рассматривать во внешнем цикле. Для строгого поддержания инвариантов цикла можно считать, что созданный на i итерации в процессе устранения непосредственной левой рекурсии нетерминал имеет номер A_{-i} (т.е. имеет номер, меньший всех имеющихся на данный момент нетерминалов).

На i итерации внешнего цикла все правила вида $A_i \to A_j\gamma$ где $j < i$ заменяются на $A_i \to \delta_1\gamma \mid \ldots \mid \delta_k\gamma$ где $A_j \to \delta_1 \mid \ldots \mid \delta_k$. Очевидно, что одна итерация алгоритма не меняет язык, а значит язык получившийся в итоге грамматики совпадает с исходным.

Оценка времени работы

Пусть a_i количество правил для нетерминала A_i. Тогда i итерация внешнего цикла будет выполняться за $O\left(\sum\limits_{A_i \to A_j, j<i} a_j\right) + O(a_i)$, что меньше чем $O\left(\sum\limits_{i=1}^{n} a_j\right)$, значит асимптотика алгоритма $O\left(n\sum\limits_{i=1}^{n} a_j\right)$.

Худший случай

Проблема этого алгоритма в том, что в зависимости от порядка нетерминалов в множестве размер грамматки может получиться экспоненциальным.

Пример грамматики для которой имеет значение порядок нетерминалов

$$A_1 \to 0 \mid 1$$

$A_{i+1} \to A_i 0 \mid A_i 1$ для $1 \leqslant i < n$

Упорядочим множество нетерминалов по возрастанию индексов. Легко заметить, что правила для A_i будут представлять из себя все двоичные вектора длины i, а значит размер грамматики будет экспоненциальным.

Порядок выбора нетерминалов

Определение:

Говорят, что нетерминал X — **прямой левый вывод** (англ. *direct left corner*) из A, если существует правило вида $A \to X\alpha$.

Определение:

Левый вывод (англ. *left corner*) — транзитивное, рефлексивное замыкание отношения «быть прямым левым выводом».

Во внутреннем цикле алгоритма для всех нетерминалов A_i и A_j, таких что $j < i$ и A_j — прямой левый вывод из A_i заменяем все прямые левые выводы A_j из A_i на все выводы из A_j.

Это действие удаляет левую рекурсию только если A_i — леворекурсивный нетерминал и A_j содержится в выводе A_i (то есть A_i — левый вывод из A_j,в то время как A_j — левый вывод из A_i).

Перестанем добавлять бесполезные выводы, которые не участвуют в удалении левой рекурсии, упорядочив нетерминалы так: если $j < i$ и A_j — прямой левый вывод из A_i, то A_i — левый вывод из A_j. Упорядочим их по уменьшению количества различных прямых левых выводов из них.

Так как отношение «быть левым выводом» транзитивно,то если C — прямой левый вывод из B, то каждый левый вывод из C также будет левым выводом из B. А так как отношение «быть левым выводом» рефлексивно, B явлеяется своим левым выводом, а значит если C — прямой левый вывод из B — он должен следовать за B в упорядоченном множестве, если только B не является левым выводом из C.

Пример

Дана грамматика:

$$A \to S\alpha$$

$$S \to S\beta \mid A\gamma \mid \beta$$

Среди правил A непосредственной рекурсии нет, поэтому во время первой итерации внешнего цикла ничего не происходит. Во время второй итерации внешнего цикла правило $S \to A\gamma$ переходит в $S \to S\alpha\gamma$.

Грамматика имеет вид

$$A \to S\alpha$$

$$S \to S\beta \mid S\alpha\gamma$$

Устраняем левую рекурсию для S

$$S \to \beta S_1 \mid \beta$$

$$S_1 \to \beta S_1 \mid \alpha\gamma S_1 \mid \beta \mid \alpha\gamma$$

См. также

- Контекстно-свободные грамматики
- Нормальная форма Хомского
- Удаления ε-правил из грамматики

Источники информации

- *Хопкрофт Д., Мотвани Р., Ульман Д.* — **Введение в теорию автоматов, языков и вычислений**, 2-е изд. : Пер. с англ. — Москва, Издательский дом «Вильямс», 2002. — 528 с. : ISBN 5-8459-0261-4 (рус.)
- *Robert C. Moore* — Removing Left Recursion from Context-Free Grammars (https://aclanthology.org/A00-2033.pdf)