Задание 11

Атрибутные грамматики

1 Приведённые КС-грамматики

Вообще говоря, не все нетерминалы из описания КС-грамматики могут встретиться в выводе некоторого слова. Такие нетерминалы могут возникнуть в ходе различных алгоритмических преобразований – позже мы встретимся с такими преобразованиями. Для удобства, в частности для корректности работы многих алгоритмов, необходимо от таких бесполезных нетерминалов избавиться.

Выделяют два типа бесполезных нетерминалов. Нетерминал A называется *бесплодным*, если язык $L(G_A) = \{w \mid A \Rightarrow^* w\}$ пуст. Нетерминал A называется *недостижимым*, если ни одна цепочка вида $\alpha A \beta$ не выводится из S. Грамматика G называется *приведённой*, если она не содержит недостижимых и бесплодных нетерминалов.

Для того, чтобы удалить все бесплодные символы нужно действовать по следующему алгоритму:

- Множество $V_0 = T$.
- Множество V_{i+1} строим по V_i следующим образом. Положим в начале $V_{i+1} = V_i$. Если для правила $A \to \alpha$ справедливо $\alpha \in V_i^*$, то добавим нетерминал A в множество V_{i+1}.
- Как только $V_{i+1} = V_i$, объявляем $N = V_i \setminus T$, удаляем из P все правила, которые содержат нетерминалы не из V_i и заканчиваем работу.

Упражнение 1. Доказать корректность данного алгоритма.

Чтобы удалить все недостижимые символы нужно действовать по следующему алгоритму:

- Множество $V_0 = S$
- Множество V_{i+1} строим по V_i следующим образом. Положим в начале $V_{i+1} = V_i$. Если $A \in V_i$ и $A \to \alpha B \beta$, то добавим нетерминал B в множество V_{i+1}.
- Как только $V_{i+1} = V_i$, объявляем $N = V_i$, удаляем из P все правила, которые содержат нетерминалы не из V_i и заканчиваем работу.

Упражнение 2. Доказать корректность данного алгоритма.

Для того чтобы по грамматике G построить приведённую грамматику G', необходимо сначала удалить все бесплодные символы, а потом удалить все недостижимые символы. Действовать надо именно в таком порядке, потому что иначе после удаления бесплодных символов могут появится новые недостижимые символы, а после удаления недостижимых, новые бесплодные появиться не могут

2 Атрибутные грамматики

Следующий за синтаксическим анализом этап в процессе компиляции является генерация кода. В основе этого этапа лежат вычисления по дереву разбора, которые описывают с помощью атрибутных грамматик. Мы не будем детально изучать эту тему, а изучим лишь частный случай атрибутных грамматик (с синтезируемыми атрибутами).

Определение 1. КС-грамматика G называется *атрибутной с ситнезируемыми атрибутами*, если каждому нетерминалу поставлен в соответствие набор переменных (атрибутов), и при этом для каждого правила грамматики

$$X_0 \to u_0 X_1 u_1 X_2 \dots u_{n-1} X_n u_n,\ X_i \in N,\ u_i \in \Sigma^*$$

Задан набор правил вычисления некоторых атрибутов

$$X_0[\mathsf{attr}] = f(\xi_1, \xi_2, \dots, \xi_m),$$

где $\xi_i = X_k[\mathsf{attr}_j]$ – значение атрибута attr_j для нетерминала X_k , а f – некоторая функция. Набор правил вычислений атрибутов называют *атрибутной схемой.*

Замечание 1. В каждом правиле нетерминалы занумерованы; это нужно для корректного определения атрибутной схемы, если в правило входит несколько одинаковых нетерминалов. В случае, когда в правиле нетерминалы не повторяются, мы опускаем нумерацию для простоты обозначений.

Пример 1. Грамматика G задана правилами

$$S \to 1D \mid 0 \mid 1,\ D \to 1D \mid 0D \mid 1 \mid 0$$

и порождает язык двоичных записей натуральных чисел. Определим атрибутную схему для этой грамматики

$S \to 0$	$S \to 1$	$S \to 1D$	$D \to 1$	$D \to 0$
$S[\mathsf{val}] = 0$	$S[\mathsf{val}] = 1$	$S[\mathsf{val}] = D[\mathsf{ord}] + D[\mathsf{val}]$	$D[\mathsf{val}] = 1$	$D[\mathsf{val}] = 0$
			$D[\mathsf{ord}] = 2$	$D[\mathsf{ord}] = 2$

$D \to 0$	$D_0 \to 1D_1$	$D_0 \to 0D_1$
$D[\mathsf{val}] = 0$	$D_0[\mathsf{val}] = D_1[\mathsf{ord}] + D_1[\mathsf{val}]$	$D_0[\mathsf{val}] = D_1[\mathsf{val}]$
$D[\mathsf{ord}] = 2$	$D_0[\mathsf{ord}] = 2 \times D_1[\mathsf{ord}]$	$D_0[\mathsf{ord}] = 2 \times D_1[\mathsf{ord}]$

В случае, если правило содержит несколько одинаковых нетерминалов мы нумеруем их вхождение и различаем атрибуты как в случае двух последних правил. Нетерминал S имеет единственный атрибут val, а нетерминал D – атрибуты val и ord. Приведённая атрибутная схема вычисляет значение числа по его двоичной записи. Атрибут ord равен 2^l, где l – длина слова выведенного из D, атрибут val равен значению числа, двоичная запись которого выведена из нетерминала.

Приведём на рис. 1 пример вычисления атрибутов для слова 1101. Атрибуты вычисляются снизу вверх.

Больше примеров можно найти в книге Серебрякова. Обратите внимание, что в ней приедено более общее определение атрибутных грамматик; определение атрибутных грамматик в этом задании эквивалентно определению синтезируемых атрибутных грамматик из книги Серебрякова, хотя там оно и имеет другой вид. Особенно рекомендую ознакомиться с примером 5.4 (главы 5): в нём иллюстрируется как с помощью атрибутных грамматик реализовать алгоритм символического вычисления производной от функции.

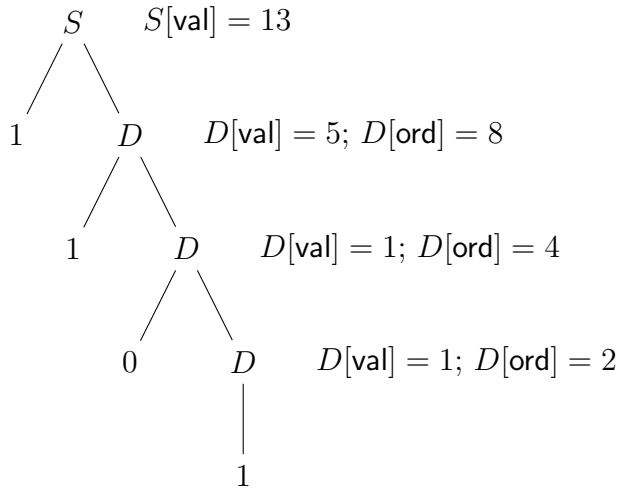

Рис. 1: вычисление атрибутов.

2.1 Атрибутные грамматики и HTML

Язык разметки web-страниц HTML был разработан так, что бы его код было легко разбирать интерпретатором. Код на HTML представляет собой правильное скобочное выражение, в котором скобки имеют имена и называются тегами: <tag> ... </tag>. Внутри открывающей скобки <tag> могут быть атрибуты; в общем случае (открывающий) тег имеет вид <name attr1="value" attr2="value" ... attrN=" value">, т. е. сначала идёт имя тега, а потом перечисляются атрбиуты.

При интерпретации HTML-кода правильная скобочная последовательность тегов интерпретируется как дерево: если один тег вложен в другой, то внешний тег — родитель внутреннего. Так, в случае кода

<tag1> ... <t2>....</t2> <t3>....</t3>...</tag1>

тег tag1 является родителем тегов t2 и t3.

Рассмотрим тег , который является контейнером для разделения содержимого страницы. Будем рассматривать два атрибута: align и style. В случае вложенных тегов, атрибут align либо задан, например , либо наследуется от родителя. Атрибут style позволяет задавать (CSS) стиль элемента и устроен более сложно, чем align. Мы сосредоточимся только на задании с помощью атрибута style цвета

фона у контейнера . В случае вложенных , если цвет фона не задан, то он наследуется у родителей.

3 Задачи

Задача 1. Постройте по грамматике G приведённую грамматику. Все построения должны быть выполнены строго по алгоритму. Грамматика G задана правилами:

$$S \to A \mid B \mid C \mid E \mid AG$$
$$A \to C \mid aABC \mid \varepsilon$$
$$B \to bABa \mid aCbDaGb \mid \varepsilon$$
$$C \to BaAbC \mid aGD \mid \varepsilon$$
$$F \to aBaaCbA \mid aGE$$
$$E \to A$$

Задача 2. Написать для грамматики эквивалентную LL(1)-грамматику, построить LL(1)-анализатор и продемонстрировать его работу на слове $baab$.

$$S \to baaA \mid babA \qquad A \to \varepsilon \mid Aa \mid Ab.$$

Задача 3. Дополните грамматику $S \to 0S11$, $S \to 1S00$, $S \to \varepsilon$ до атрибутной так, чтобы вычислялась максимальная длина непрерывной последовательности из единиц в порождаемом слове.

Задача 4. Постройте по коду на рис. 2 дерево html документа. Определите значение атрибута align у каждого из узлов div дерева, а также определите цвет фона элемента (атрибут background-color). Проверьте себя, сохранив текст ниже в файле с расширением .html и открыв файл в браузере.

Комментарий. В HTML-документе код документа окружается тегом <html>, внутри тега <head> находятся вспомогательные данные (такие как заголовки), а содержимое документа находится в теге <body>. Внутри тега <style> описывается стиль элементов документа, в нашем коде там указан базовый стиль для тегов : наличие границы, отступы и размер в процентах относительно размера тега-родителя. Браузер интерпретирует документ HTML как дерево, точнее модель документа называется DOM (Document Object Model).

```
<html>
<head>
  <style>
   div{border: 1px solid  black;  padding:1px;
        margin: 1px; width:40%; height:40%;}
  </style>
</head>
<body>
<div style="background-color:lightblue; width:500px;
       height:500px;" align="center">1
  <div style="background-color:blue;" align="left">
    2
    <div align="right">
      3
      <div style="background-color:gray;" align="center">
        4
      </div>
    </div>
    <div>
      5
    </div>
  </div>
  <div>
    6
  </div>
</div>
</body>
</html>
```

Рис. 2: HTML-код для задачи 4