

Домашнее Задание по ТРЯПу №11

Павливский Сергей Алексеевич , 873

25.11.2019

Задание 1.

Постройте по грамматике G приведённую грамматику. Все построения должны быть выполнены строго по алгоритму.

Грамматика G задана правилами:

$$S \rightarrow A \mid B \mid C \mid E \mid AG$$

$$A \rightarrow C \mid aABC \mid \varepsilon$$

$$B \rightarrow bABa \mid aCbDaGb \mid \varepsilon$$

$$C \rightarrow BaAbC \mid aGD \mid \varepsilon$$

$$F \rightarrow aBaACbA \mid aGE$$

$$E \rightarrow A$$

Решение

В соответствии с алгоритмом сначала удаляем бесплодные символы , а затем недостижимые символы .

Удаляем бесплодные символы :

1)

$$V_0 = \{a, b\}$$

2)

$$V_1 = V_0 = \{a, b\}$$

$$\varepsilon \in V_1^* \rightarrow V_1 = \{a, b, A\}$$

$$\text{Ан-о} , V_1 = \{a, b, A, B, C\}$$

3)

$$V_2 = V_1 = \{a, b, A, B, C\}$$

Из S, F и E напрямую выводимы $a : a \in V_2^* \rightarrow V_2 = \{a, b, A, B, C, S, F, E\}$

4)

$V_3 = V_2$, и ни каких нетерминалов больше добавить нельзя

. Значит алгоритм заканчивает свою работу, убирая правила использующие нетерминал G.

Преобразованная грамматика :

$$S \rightarrow A \mid B \mid C \mid E$$

$$A \rightarrow C \mid aABC \mid \varepsilon$$

$$B \rightarrow bABa \mid \varepsilon$$

$$C \rightarrow BaAbC \mid \varepsilon$$

$$F \rightarrow aBaac'bA$$

$$E \rightarrow A$$

Теперь убираем недостижимые символы :

1)

$$V_0 = \{S\}$$

2)

$$V_1 = V_0$$

Т.к. $S \rightarrow A \mid B \mid C \mid E$, то $V_1 = \{S, A, B, C, E\}$

3)

$V_2 = V_1$, и ни каких переходов в символы не из V_1 нет,

т.е. алгоритм заканчивает работу, удаляя правила, в которых фигурирует нетерминал F.

Окончательная грамматика :

$$S \rightarrow A \mid B \mid C \mid E$$

$$A \rightarrow C \mid aABC \mid \varepsilon$$

$$B \rightarrow bABa \mid \varepsilon$$

$$C \rightarrow BaAbC \mid \varepsilon$$

$$E \rightarrow A$$

Это и есть требуемая грамматика.

Задание 2.

Написать для грамматики эквивалентную LL(1)-грамматику, построить LL(1)-анализатор и продемонстрировать его работу

на слове baab.

$$S \rightarrow baaA \mid babA$$

$$A \rightarrow \varepsilon \mid Aa \mid Ab$$

Решение

Воспользуемся алгоритмом удаления непосредственной левой рекурсии для нетерминала A :

$$A \rightarrow Aa \mid Ab \mid \varepsilon$$

Преобразуем :

$$A \rightarrow A' \mid \varepsilon$$

$$A' \rightarrow aA' \mid bA' \mid a \mid b$$

Теперь грамматика стала :

$$S \rightarrow baaA \mid babA$$

$$A \rightarrow A' \mid \varepsilon$$

$$A' \rightarrow aA' \mid bA' \mid a \mid b$$

Воспользуемся алгоритмом левой факторизации :

Возьмем наибольший префикс цепочек порождаемых нетерминалом S равный ba

Тогда преобразованием

$$S \rightarrow baaA \mid babA$$

переходит в

$$S \rightarrow baS'$$

$$S' \rightarrow aA \mid bA$$

Теперь сделаем аналогичные действия для A' :

$$A' \rightarrow aA' \mid bA' \mid a \mid b$$

Преобразованием

$$A' \rightarrow aA' \mid a \mid bA' \mid b$$

переходит в

$$A' \rightarrow aA'' \mid bA' \mid b$$

$$A'' \rightarrow A' \mid \varepsilon$$

Аналогичным преобразованием

$$A' \rightarrow aA'' \mid bA''$$

$$A'' \rightarrow A' \mid \varepsilon$$

(после b такой же нетерминал , т.к. в результате преобразования получался бы нетерминал A''' с такими же переходами , что и A'')

Тогда в результате удаления левой рекурсии и левой факторизации из исходной грамматики

$$S \rightarrow baaA \mid babA$$

$$A \rightarrow \varepsilon \mid Aa \mid Ab.$$

получилась

$$S \rightarrow baS'$$

$$S' \rightarrow aA \mid bA$$

$$A \rightarrow A' \mid \varepsilon$$

$$A' \rightarrow aA'' \mid bA''$$

$$A'' \rightarrow A' \mid \varepsilon$$

Как можно видеть, это LL(1) грамматика, так как FIRST от правых частей одинаковых правил не пересекаются, а FOLLOW у всех нетерминалов - это только конец слова (который не может пересекаться с FIRST от правой части нетерминала). Для удобства переобозначим в ней нетерминалы:

$$S \rightarrow baT$$

$$T \rightarrow aA \mid bA$$

$$A \rightarrow B \mid \varepsilon$$

$$B \rightarrow aC \mid bC$$

$$C \rightarrow B \mid \varepsilon$$

Вычислим функции FIRST и FOLLOW.

FIRST:

$$S \{b\}$$

$$T \{a, b\}$$

$$A \{a, b, \varepsilon\}$$

$$B \{a, b\}$$

$$C \{a, b, \varepsilon\}$$

FOLLOW:

$$S \{\$ \}$$

$$T \{\$ \}$$

$$A \{\$ \}$$

$$B \{\$ \}$$

$$C \{\$ \}$$

(т.к. каждый переход порождает не более одного нетерминала, и всегда ставит его в конец порожденной цепочки).

Построим LL(1) анализатор, занумеровав правила вывода:

$$S \rightarrow baT \quad (1)$$

$$T \rightarrow aA \quad (2)$$

	S	T	A	B	C
a	-	2	3	5	7
b	1	9	3	6	7
\$	-	-	4	-	8

$$A \rightarrow B \text{ (3)}$$

$$A \rightarrow \varepsilon \text{ (4)}$$

$$B \rightarrow aC \text{ (5)}$$

$$B \rightarrow bC \text{ (6)}$$

$$C \rightarrow B \text{ (7)}$$

$$C \rightarrow \varepsilon \text{ (8)}$$

$$T \rightarrow bA \text{ (9)}$$

(на пересечении нетерминала и терминала из FIRST (нетерминал) ставим соответствующее правило , из которого получается данный терминал в начале цепочки , порожденной нетерминалом , если же из нетерминала еще выводится пустое слово , то ставим переход по пустому слову на пересечении нетерминала с элементами из FOLLOW (нетерминал)) .

Продемонстрируем работу на слове baab :

$$(baab\$ \mid S\$)$$

$$(baab\$ \mid baT\$)$$

$$(aab\$ \mid aT\$)$$

$$(ab\$ \mid T\$)$$

$$(ab\$ \mid aA\$)$$

$$(b\$ \mid A\$)$$

$$(b\$ \mid B\$)$$

$$(b\$ \mid bC\$)$$

$$(\$ \mid C\$)$$

$$(\$ \mid \$)$$

Задание 3.

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

Решение

Добавим атрибуты $Type$, $currentNumber$, $maxNumber$

$$S \rightarrow \varepsilon$$

$$S[Type] = 0;$$

$$S[currentNumber] = 0;$$

$$S[maxNumber] = 0;$$

$$S \rightarrow 1S00$$

$$S[Type] = 1$$

$$S_0[currentNumber] = ((S_0[Type] + S_1[Type] + 1) \bmod 2) * (S_1[currentNumber] + 1) + ((S_0[Type] + S_1[Type]) \bmod 2) * 1;$$

$$S_0[maxNumber] = \max(S_0[currentNumber], S_1[maxNumber])$$

$$S \rightarrow 0S11$$

$$S[Type] = 2$$

$$S_0[currentNumber] = ((S_0[Type] + S_1[Type] + 1) \bmod 2) * (S_1[currentNumber] + 2) + ((S_0[Type] + S_1[Type]) \bmod 2) * 2;$$

$$S_0[maxNumber] = \max(S_0[currentNumber], S_1[maxNumber])$$

Принцип работы атрибутов таков :

$Type$ - исключительно для определения типа конструкции перехода , так как от нее зависит количество добавляемых единиц ;

$currentNumber$ - количество единиц в рассматриваемой в данный момент подпоследовательности ;

$maxNumber$ - максимальное количество единиц в последовательности , которая была встречена .

Так как вычисление атрибутов идет по дереву разбора снизу вверх , то нетерминалам S с переход по ε ставится в соответствие уникальный тип и значения количеств единиц во встреченных подпоследовательностях (так как не было встречено ни одной , то они равны 0) .

При вычислениях атрибутов S с переходами не по ε , используется следующий набор идей:

- Текущее максимальное значение, это максимум из текущего значения и максимального ранее встреченного, которое, совершая индуктивный переход, равно максимальному для предыдущего перехода

- Если меняется тип перехода, т.е. например рассматривается переход $S \rightarrow 1S00$, а до этого совершался переход $S \rightarrow 0S11$, то гарантировано ранее шедшая последовательность из 1 прервется, так как один переход оставляет с одного края от себя 0, а с другого 1, а другой наоборот. Поэтому после смены типа перехода, `currentNumber` сбрасывается и в него записываются новопришедшие единицы, начинающие новую последовательность, в противном случае `currentNumber` суммируется с предыдущим `currentNumber`, т.к. последовательность единиц не прервалась. Собственно, $S_0[\text{Type}] + S_1[\text{Type}] + 1 \bmod 2 = 1$, если тип текущего перехода совпадает с предыдущим, и сохраняет `currentNumber` предыдущего, увеличивая его на пришедшее количество единиц, и обнуляет слагаемое $S_1[\text{currentNumber}] + 1$ в противном случае, оставляя в `currentNumber` текущего только пришедшие единицы.

Из вышесказанного, атрибуты корректно хранят текущее значение количества символов в генерируемой непрерывной последовательности единиц, максимальное количество единиц среди всех встреченных подпоследовательностей и при этом все корректно завершат вычисление при подъеме к вершине дерева разбора, ч.т.д.

Задание 4.

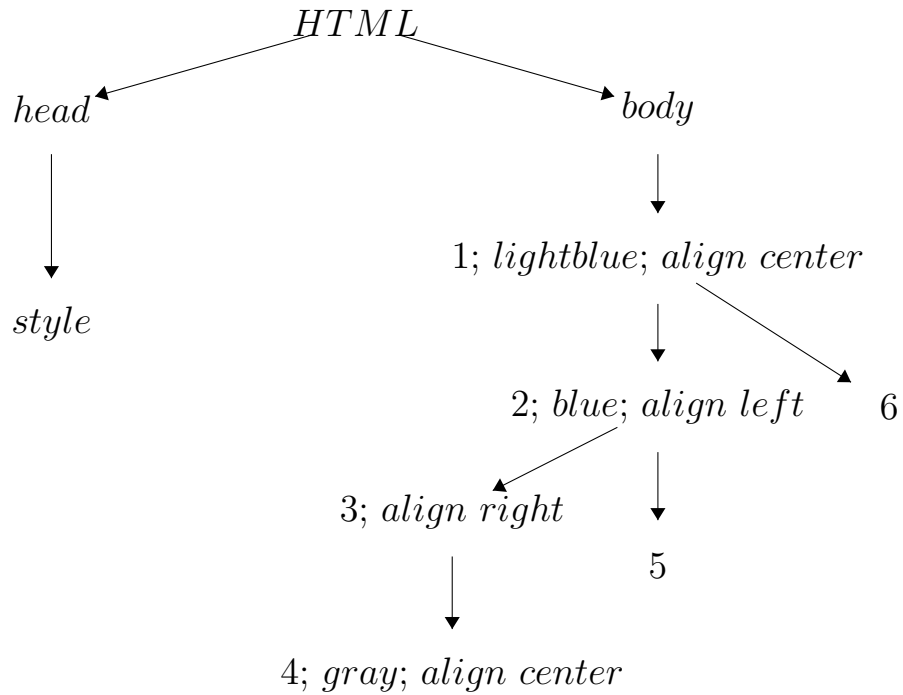
Постройте по коду дерево html документа. Определите значение атрибута `align` у каждого из узлов `div` дерева, а также определите цвет фона элемента (атрибут `background-color`). Проверьте себя, сохранив текст ниже в файле с расширением `.html` и открыв файл в браузере.

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 `<div>`: 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).

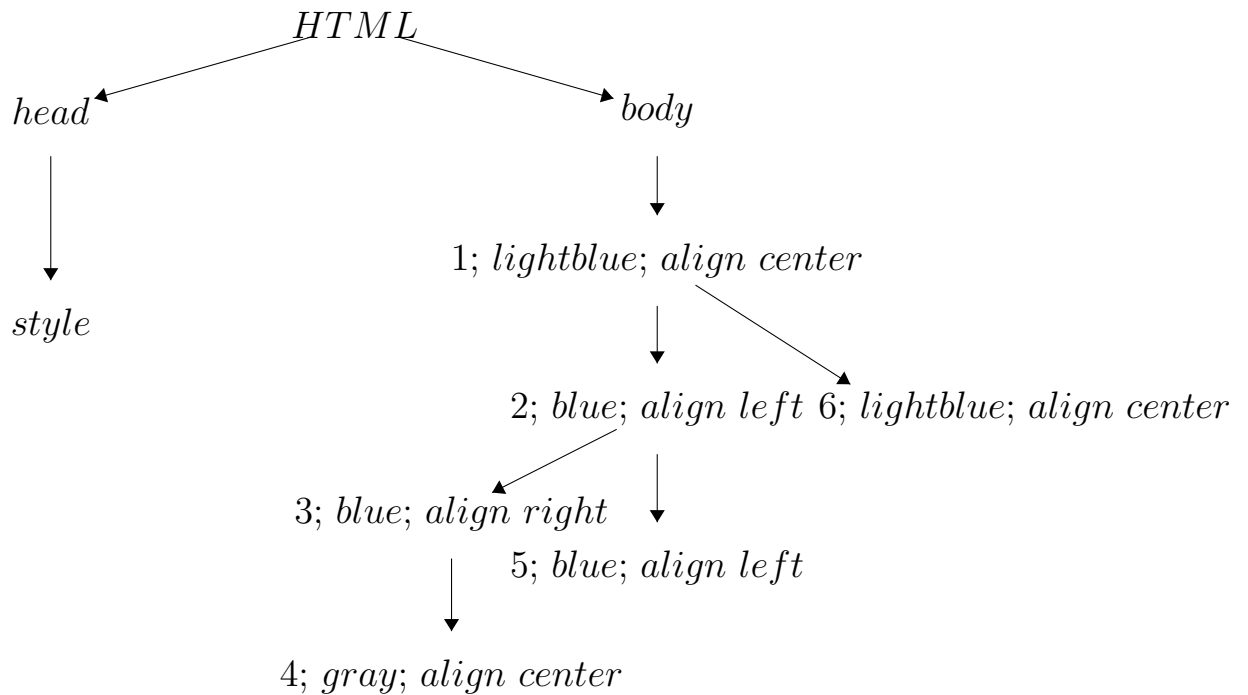
```
<html>
<head>
<style>
div{border: 1px solid black; padding:1px; margin: 1px; width:40%;
height:40%;}
</style>
</head>
<body>
<div style="background-color:lightblue; width:500px; height:500px;"align="center"
<div style="background-color:blue;"align="left»
2
<div align="right»
3
<div style="background-color:gray;"align="center»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


Решение

Построим дерево



Тогда , т.к. атрибуты align и style наследуются в случае вложенности , то дерево становится



Полученные данные сходятся с изображением при запуске кода .