

Задание 12

LR-анализ

Это задание, которое не требуется сдавать на проверку!

Для самопроверки используйте онлайн-конструктор анализаторов (см. описание на сайте). Подробный теоретический материал по теме LR также приведён на сайте в отдельном файле.

1 Подготовка к изучению LR-анализа

LR-анализаторы устроены гораздо сложнее, чем LL-анализаторы. Их построение можно изучить по книге Серебрякова, а за подробностями обратиться к книге Ахо и Ульмана, оригинальной статье Кнута или тексту на странице курса. В этом разделе мы приведём технические конструкции, нужные для построения LR(1)-анализатора.

Зафиксируем пополненную грамматику G . Назовём LR(k)-ситуацией набор

$$[A \rightarrow X_1 X_2 \dots X_i \cdot X_{i+1} \dots X_n, u],$$

где $A \rightarrow X_1 \dots X_n$ — правило G , u — *аванцепочка* — слово длины не больше k (параметр LR(k) анализатора) над алфавитом $\Sigma \cup \{\$\}$ (возможно пустое), а « \cdot » — вспомогательный символ, *курсор*, который встречается ровно один раз.

Определим недетерминированный LR(k)-автомат Кнута для грамматики G . Множество состояний автомата — LR(k)-ситуации грамматики G . Начальное состояние автомата — ситуация¹ $[S' \rightarrow \cdot S, \$]$. Переходы определены следующим образом:

Если состояние — ситуация вида $[A \rightarrow \alpha \cdot B\gamma, u]$, то для каждого правила $B \rightarrow \beta$ и каждого слова $v \in \text{FIRST}_k(\gamma u)$ определён ε переход

$$[A \rightarrow \alpha \cdot B\gamma, u] \xrightarrow{\varepsilon} [B \rightarrow \cdot \beta, v]$$

¹Эта ситуация и ещё одна $[S' \rightarrow S \cdot, \$]$ формально не удовлетворяют определению выше; мы выделяем их в особый случай, что бы не тратить на обсуждение этого технического момента много времени.

Для состояния (ситуации) $[A \rightarrow \alpha \cdot X\gamma, u]$, где $X \in \Sigma \cup N$ определён переход

$$[A \rightarrow \alpha \cdot X\gamma, u] \xrightarrow{X} [A \rightarrow \alpha X \cdot \gamma, u]$$

Множество принимающих состояний несущественно для дальнейшего использования автомата Кнута, будем считать, что оно пусто.

Упражнение 1. Постройте недетерминированный LR(1)-автомат Кнута для грамматики $S \rightarrow Sa \mid b \mid \varepsilon$. Детерминизируйте этот автомат по алгоритму из курса — вы получите детерминированный автомат Кнута, с помощью которого строится управляющая таблица LR(1)-анализатора. При построении автомата Кнута используйте только состояния, достижимые из начального.

2 LR(0)-анализаторы

Параметр k в LR(k) анализаторах — число предпросматриваемых символов, по которым анализатор принимает решение какое из действий «сдвиг» или «свёртка» совершить на текущем такте. На первый взгляд может показаться, что $k \geq 1$, однако существуют и LR(0)-анализаторы. LR(0)-анализатор принимает решение не зная какой символ находится на входной ленте. Как же такое возможно?

Принимать решение без предпросмотра возможно, поскольку не смотря на независимость решений для произвольного символа, переходы между состояниями LR(0)-автомата Кнута по-прежнему зависят от символа помещённого в стек. В управляющей таблице LR(0)-анализатора каждому состоянию соответствует ровно одно действие (независимо от символа входа). В остальном, LR(0)-анализаторы не отличаются от LR(1)-анализаторов, изученных на семинаре.

LR(0)-анализатор принимает слово, если оказался в конфигурации $(I_0 S I_1, \$, p_n p_{n-1} \dots p_1)$, где p_i — номер правила, а состоянию I_1 соответствует действие **Акцепт**.

Упражнение 2. Постройте LR(0)-анализатор для грамматики $S \rightarrow aSb \mid b$. Продемонстрируйте работу анализатора на слове $aabbb$ и постройте по результатам работы дерево разбора. Если затрудняетесь, то воспользуйтесь конструктором анализаторов.

Как понять, является ли грамматика LR(0)- или LR(1)-грамматикой?

(точнее, какое из чисел 0 или 1 является минимальным k). Есть критерий LR(k)-грамматик, записанный, например, в книге Ахо и Ульмана, но он достаточно сложный и мы не будем приводить его здесь. Мы приведём следующий рецепт.

Постройте LR(1)-анализатор и если в управляющей таблице некоторому состоянию I_m , в зависимости от символа, соответствуют два разных правила свёртки (в том числе и **Accept**) или по одному символу должна быть операция сдвиг, а по другому — свёртка, то эта грамматика заведомо не LR(0). Действительно, состояния детерминированного LR(0)-автомата Кнута отличаются от состояний LR(1)-автомата Кнута только в следующем. Каждому состоянию LR(1)-автомата соответствует состояние LR(0)-автомата; последнее содержит ситуацию² $[A \rightarrow \alpha \cdot X\beta]$ тогда и только тогда, когда в первом есть ситуация вида $[A \rightarrow \alpha \cdot X\beta, u]$. Это свойство легко вытекает из процедуры детерминизации. Таким образом, если в некотором состоянии LR(1)-автомата есть две разных операции (по разным символам), то при построении LR(0)-анализатора, в некотором состоянии (множестве ситуаций) потребуется совершить две разных операции.

3 Задачи

Задача 1. Грамматика G задана правилами:

$$S \rightarrow Ab, \quad A \rightarrow aAa, \quad A \rightarrow B, \quad B \rightarrow b.$$

1. Построить LR(1) и LR(0)-анализаторы для грамматики G по алгоритму из курса.
2. Постройте LR(0)-анализатор по LR(1)-анализатору из пункта 1 следующим образом. Сотрите все аванцепочки и постройте управляющую таблицу LR(0)-анализатора по получившемуся автомату Кнута. Верно ли, что полученный LR(0)-анализатор является анализатором для грамматики G ? То есть для любого слова, порождаемого G , анализатор строит корректный правый разбор, а слова, не порождаемые G , анализатор отвергает.

²Мы опускаем аванцепочку в LR(0)-ситуациях, поскольку её длина ноль.

I	a	\$	+	*	S'	S	T	Q	F	W	+	a	*	\$
I ₀	S					1	2		3			4		
I ₁	R (S'→S)													
I ₂	R (Q→)	S					5				6			
I ₃	R (W→)	R (W→)	S								7		8	
I ₄	R (F→a)	R (F→a)	R (F→a)											
I ₅	R (S→TQ)													
I ₆	S					9			3			4		
I ₇	R (T→FW)	R (T→FW)												
I ₈	S								10			4		
I ₉	R (Q→)	S					11				6			
I ₁₀	R (W→)	R (W→)	S								12		8	
I ₁₁	R (Q→+TQ)													
I ₁₂	R (W→*FW)	R (W→*FW)												

Рис. 1: автомат LR(1)-анализатора

Задача 2. Дана грамматика $G = \{ \{A, S\}, \{a, b, c\}, \{ S \rightarrow Aa \mid b \mid \varepsilon; A \rightarrow Ab \mid c \}, S \}$. Является ли грамматика G LR(k)-грамматикой? При положительном ответе на вопрос найти минимальное k и построить соответствующий анализатор. Построить дерево разбора для цепочки $cbba$.

Задача 3. Дана грамматика $G = \{ \{A, S\}, \{a, b, c\}, \{ S \rightarrow Aa \mid b; A \rightarrow Ab \mid c \}, S \}$. Является ли грамматика G LR(k)-грамматикой? При положительном ответе на вопрос найти минимальное k и построить соответствующий анализатор. Продемонстрировать работу анализатора на цепочке $cbbab$.

Задача 4. На рис. 1 приведен автомат LR(1)-анализатора (запись $A \rightarrow$ обозначает правило $A \rightarrow \varepsilon$). В приведенной ниже конфигурации LR-анализатора в первой компоненте (содержимом магазина) опущены состояния автомата. В процессе разбора строки $z \in L(G)$ автомат оказался в конфигурации $\langle cF * FW, +a + a \rangle$. Требуется:

1. Восстановить состояния автомата в содержимом магазина.
2. Восстановить какую-либо из возможных строк $z \in L(G)$, разбор которой мог привести к этой конфигурации.
3. Продемонстрировать процесс разбора на этой строке. Решение обоснуйте.

Задача 5. Грамматика $\text{RExpr} = \langle \{E, T\}, \{a, b, +, *, (,)\}, P, E \rangle$ имеет множество правил P :

$$E \rightarrow T + E \mid TE \mid T \quad T \rightarrow (E) \mid T^* \mid a \mid b$$

и порождает регулярные выражения над алфавитом $\{a, b\}$.

1. Постройте для грамматики RExpr LR(1)-анализатор³.
2. Дополните грамматику RExpr до атрибутной так, чтобы она вычисляла атрибуты `firstpos`, `lastpos` и `nullable` согласно алгоритму их вычисления при построении ДКА по РВ. Считайте атрибуты `firstpos`, `lastpos` и `nullable` у терминалов заданными: на практике перед вычислением атрибутов происходит препроцессинг, во время которого могут быть заданы атрибуты терминалов, если это требуется.

Указание. Вычисление атрибута нужно определять через функцию, которую можно описать на языке программирования или псевдокоде. Пример вычисления атрибута `nullable` для правила $E_0 \rightarrow T + E_1$:

```

E0[nullable] = function( T[nullable], E1[nullable]){
    if( T[nullable] or E1[nullable]){
        return True;
    } else{ return False; }
}

```

- 3*. Добавьте в атрибутную схему вычисление атрибута `followpos`.
4. С помощью анализатора постройте дерево разбора для РВ $(a + ab)^* + ab^*$ и вычислите атрибуты `firstpos`, `lastpos`, `nullable` (`*followpos`) согласно атрибутной схеме (предварительно задав атрибуты у терминалов).

³Анализатор для этой грамматики довольно громоздкий. Постройте его с помощью программы, например на сайте <http://lrk.umeta.ru>, используйте программу для выполнения последующих пунктов.

3.1 Контрольные вопросы

Задача 6. При построении LR(1)–анализатора для грамматики G в одном множестве оказались ситуации $[A \rightarrow .aA\alpha, b]$ и $[B \rightarrow \beta.a, a]$, где α, β некоторые цепочки из $(N \cup T)^*$. Может ли грамматика G быть LR(0)-грамматикой?