ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Костюк Ю. Л.

ТЕОРИЯ АВТОМАТОВ И ФОРМАЛЬНЫХ ЯЗЫКОВ

Лекция 3

СИНТАКСИЧЕСКИЙ АНАЛИЗ

Дерево порождения для КС-грамматики

Все порождающие правила имеют вид: $A \rightarrow \gamma$,

где A – нетерминальный символ, γ – цепочка из терминальных и нетерминальных символов.

Процесс порождения в КС-грамматике – построение дерева порождения.

Вершины дерева – терминальные и нетерминальные символы.

Корень дерева – начальный нетерминал, вниз от него идут ребра, в концах которых размещаются символы правой части порождающего правила.

От каждой вершины – нетерминала достраиваются вниз ребра с вершинами – символами в правой части примененного порождающего правила.

Процесс продолжается до тех пор, пока среди висячих вершин дерева не останется ни одного нетерминала.

Обход дерева слева направо по висячим вершинам дерева будет цепочкой языка, получившейся в процессе порождения.

Однозначность грамматики и языка

Левое каноническое порождение – если на каждом очередном шаге порождения дерево достраивается от самой левой висячей вершины – нетерминала.

Каждому построенному дереву порождения однозначно соответствует некоторое левое каноническое порождение.

Если для порождения некоторой цепочки языка возможно построение двух (или более) различных деревьев порождения, то грамматика является неоднозначной.

Язык неоднозначен, если для него не существует ни одной однозначной грамматики

Автомат с магазинной памятью (МПА)

МПА задается семеркой множеств:

$$\{\Sigma, Q, q_0, F, \Gamma, Z_0, \delta\},$$

где Σ – множество (алфавит) входных символов;

Q – множество состояний МПА;

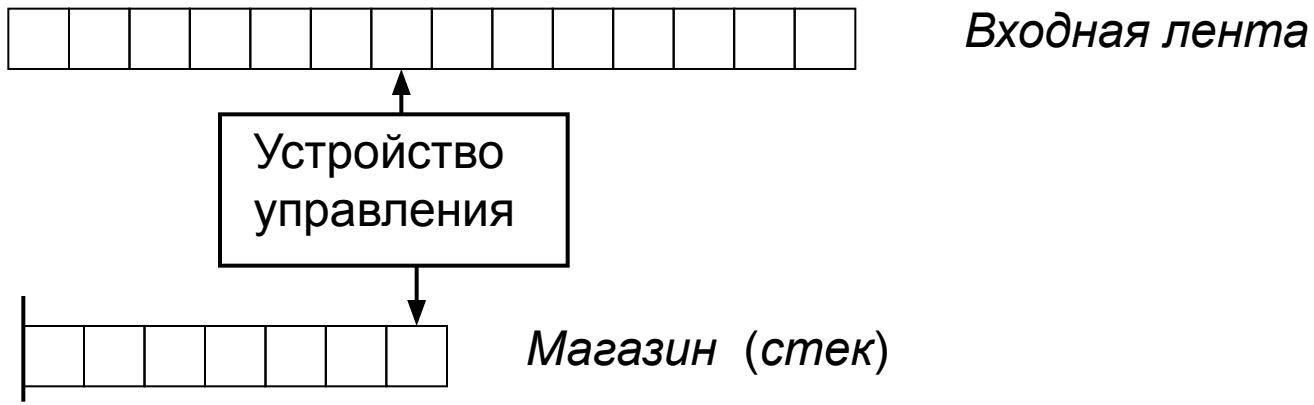
q_0 – начальное состояние, $q_0 \subset Q$;

F – множество заключительных состояний, $F \subseteq Q$;

Γ – множество (алфавит) магазинных символов;

Z_0 – начальный магазинный символ, $Z_0 \subset \Gamma$;

δ – множество правил перехода.



Действие в правиле перехода определяется входным символом, верхним символом магазина и текущим состоянием. Действие может содержать в различных сочетаниях такие шаги:
а) удаление из магазина верхнего символа, б) запись в магазин нескольких символов, в) переход к чтению следующего входного символа, г) изменение текущего состояния.

Запись правила перехода в виде:

$$(a, B, q_j) \rightarrow (\lambda, \gamma B, q_j),$$

где $a \subset \Sigma,\ q_i \subset Q,\ q_j \subset Q$, символ B и символы из цепочки γ все принадлежат алфавиту Г, означает, что в магазин, содержащий верхний символ B, записывается цепочка символов γ, текущее состояние становится q_j , после чего делается переход к чтению следующего входного символа. Правило:

$$(a, B, q_i) \rightarrow (a, \lambda, q_j),$$

означает, что из магазина удаляется верхний символ B, текущее состояние становится q_j , переход к чтению следующего входного символа не производится.

До начала функционирования МПА в магазине имеется начальный магазинный символ, МПА находится в начальном состоянии.

На каждом шаге работы читается очередной символ входной цепочки, начиная с первого, читается верхний символ магазина, и выполняются действия по такому правилу перехода, которое соответствует этим символам и текущему состоянию.

Работа МПА заканчивается, когда не будет ни одного подходящего правила, чтобы можно было продолжать работу.

Если при этом выполнены три условия: цепочка прочитана вся, магазин пуст, МПА находится в заключительном состоянии, то цепочка считается распознанной.

Если же хотя бы одно из этих условий нарушено, цепочка считается нераспознанной. Цепочка считается нераспознанной и в том случае, если МПА никак не может остановиться, т.е. когда он зациклил, и нет продвижения по входной цепочке.

МПА будет детерминированным (ДМПА), если на каждом шаге возможно применение не более одного правила перехода.

Если же на каком-либо шаге можно выполнить действия по двум или более правилам перехода, то МПА будет недетерминированным (НМПА), тогда он должен выполнять действия в соответствии со всеми этими правилами параллельно и независимо, создавая собственные копии.

Если хотя бы одна из копий НМПА ***распознает*** входную цепочку, то считается, что входная цепочка ***распознана*** недетерминированным МПА.

Входная цепочка считается ***нераспознанной***, если ни одна из копий НМПА ***не распознала*** цепочку.

LL-анализ КС-грамматики автоматом с магазинной памятью

LL-анализ – цепочка символов читается слева направо, дерево порождения неявно строится, как при левом каноническом порождении.

Пусть задана КС-грамматика. Построим по ней МПА, выполняющий синтаксический LL-анализ.

Множество входных символов Σ в МПА будет совпадать с множеством терминальных символов грамматики.

Множество состояний Q будет состоять из единственного состояния q_0, оно же будет начальным и заключительным.

Множество магазинных символов Г будет содержать все терминальные и нетерминальные символы грамматики: $\Gamma = \Sigma \cup N$. Начальный магазинный символ Z_0 будет совпадать с начальным символом грамматики S.

Правила перехода δ для МПА задаются следующим образом.

Для каждого порождающего правила вида $A \to \gamma$ соответствует правило перехода:

$$(\lambda, A, q_0) \to (\lambda, \gamma^{-1}, q_0), \qquad (*)$$

где γ^{-1} – правая часть правила γ в обратном порядке.

Для каждого терминального символа a правило перехода:

$$(a, a, q_0) \to (\lambda, \lambda, q_0). \qquad (**)$$

Этот МПА будет повторять левое каноническое порождение.

В общем случае МПА будет недетерминированным, и будет пытаться строить все варианты левого канонического порождения.

Теорема.

НМПА, реализующий LL-анализ некоторой КС-грамматики, допускает входную цепочку тогда и только тогда, когда цепочка порождается этой КС-грамматикой.

Доказательство – методом от «противного» для двух случаев, когда входная цепочка порождается, и когда цепочка не порождается этой КС-грамматикой.

Пример. Грамматика простых арифметических выражений (S – начальный нетерминал, знаки + , * , скобки, переменная a – терминалы):

$$S \to S + T \mid T$$
$$T \to T * F \mid F$$
$$F \to (S) \mid a$$

Вертикальная черта разделяет различные правые части для одного и того же нетерминала в левой части для группы правил.

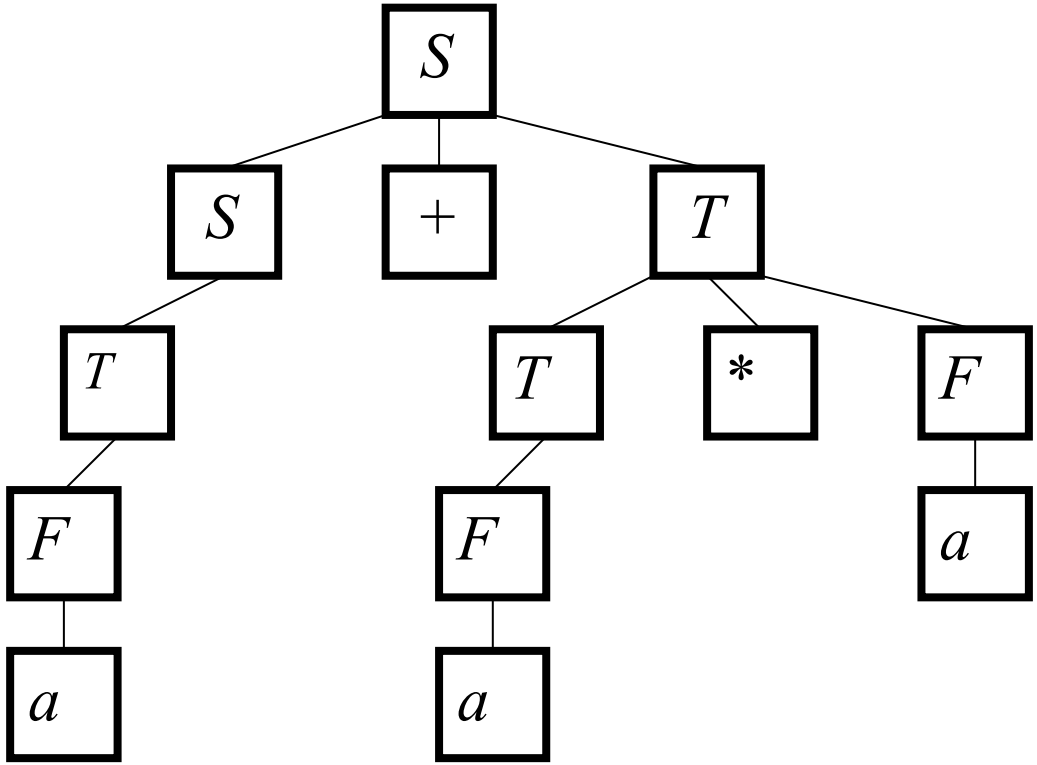
Эта грамматика задает более высокий приоритет для операции * (умножение) и более низкий для операции + (сложение).

Пусть цепочка на входе: $a + a * a$.

В таблице представлен тот *единственный* вариант шагов LL-анализа, который приводит к успешному распознаванию. Всех других возможных (***бесполезных***) вариантов – бесконечно много!

№ шага	Вход	Магазин	Правило
1	$a + a * a$	S	$S \rightarrow S + T$
2	$a + a * a$	$S + T$	$S \rightarrow T$
3	$a + a * a$	$T + T$	$T \rightarrow F$
4	$a + a * a$	$F + T$	$F \rightarrow a$
5	$a + a * a$	$a + T$	
6	$+ a * a$	$+ T$	
7	$a * a$	T	$T \rightarrow T * F$
8	$a * a$	$T * F$	$T \rightarrow F$
9	$a * a$	$F * F$	$F \rightarrow a$
10	$a * a$	$a * F$	
11	$* a$	$* F$	
12	a	F	$F \rightarrow a$
13	a	a	
14	λ	λ	

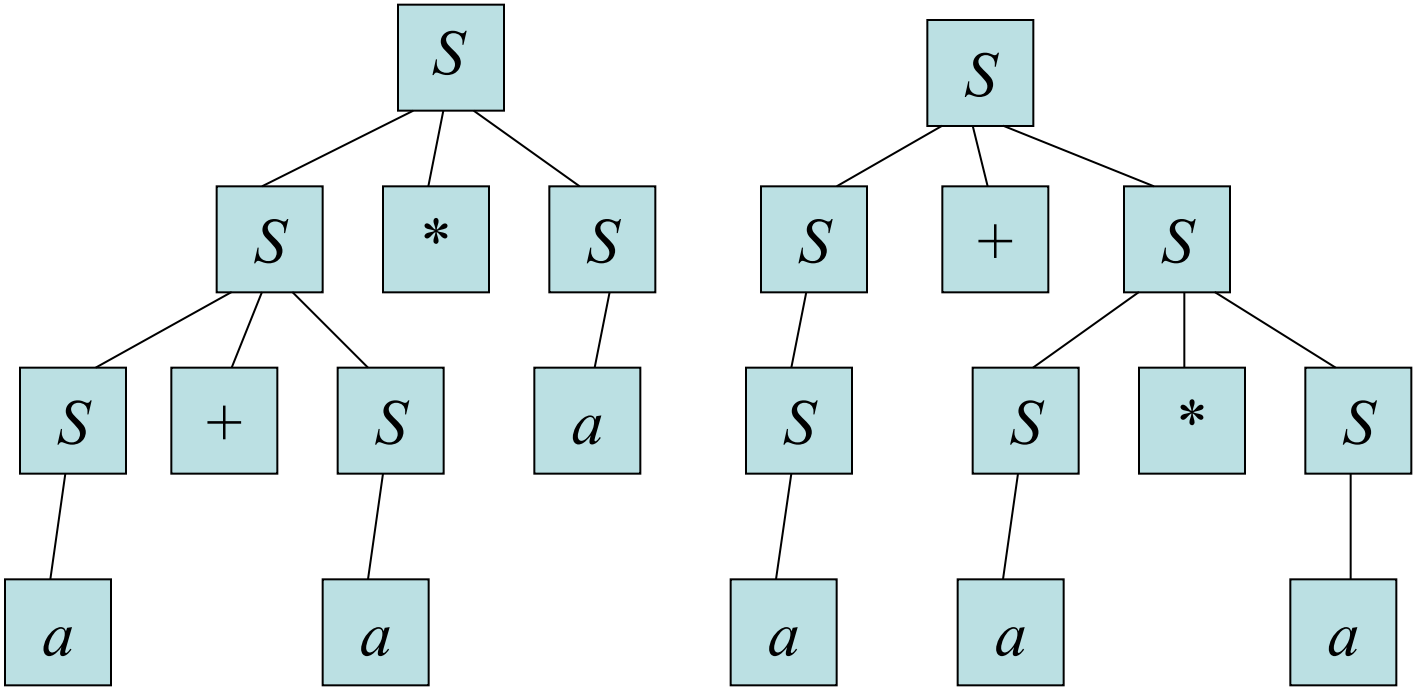
На рисунке изображено ***неявно строящееся*** при анализе дерево порождения цепочки: $a + a * a$.



Пример. Грамматика, эквивалентная грамматике простых арифметических выражений:

$$S \rightarrow S + S \mid S * S \mid (S) \mid a$$

Для цепочки $a + a * a$ существует два дерева порождения, т.е. грамматика неоднозначная!
НМПА построит (неявно) оба дерева.



Детерминированный LL-анализ

Левая рекурсия и ее устранение

Правило порождения вида $A \rightarrow A\gamma$ леворекурсивное. Тогда должно быть также правило вида $A \rightarrow \alpha$, где первый символ цепочки α не совпадает с A, так как в противном случае символ A – бесполезный.

Цепочки символов, порождаемые этими двумя правилами:

α, $\alpha\gamma$, $\alpha\gamma\gamma$ и т.д.

Эти же цепочки можно получить другими правилами:

$A \rightarrow \alpha A'$

$A' \rightarrow \gamma A' | \lambda$,

где A' – новый нетерминал.

Косвенная рекурсия, когда имеется совокупность правил вида:

$$A \to B_1\gamma_1,\ B_1 \to B_2\gamma_2,\ \ldots,\ B_n \to A\gamma_n.$$

Тогда вначале косвенную рекурсию нужно свести к непосредственной, сделав следующую замену этой группы правил на правило:

$$A \to A\gamma_n \ldots\ \gamma_2\ \gamma_1.$$

При этом надо учесть и другие совокупности правил, аналогичные рассмотренным, в которые входят какие-либо из нетерминалов

$$B_1,\ \ldots,\ B_n.$$

Пример. Грамматика простых арифметических выражений:

$$S \to S + T \mid T$$
$$T \to T * F \mid F$$
$$F \to (S) \mid a$$

***После устранения левой рекурсии*:**

$$S \to TU$$
$$U \to + TU \mid \lambda$$
$$T \to FV$$
$$V \to * FV \mid \lambda$$
$$F \to (S) \mid a$$

Нестрогая нормальная форма Грейбах

Нормальная форма Грейбах: правые части всех порождающих правил начинаются с терминала.

Нестрогая нормальная форма Грейбах: правая часть может быть пустой!

Преобразование

В каждом правиле вида $A \to B\alpha$ нетерминал B заменяется совокупностью правых частей правил

$$B \to \gamma_1 \mid \ldots \mid \gamma_n$$

В результате получатся правила:

$$A \to \gamma_1 \alpha \mid \ldots \mid \gamma_n \alpha$$

Когда все подобные замены будут сделаны, все правые части правил будут начинаться с терминальных символов или будут пустыми.

Следствие: ***Любой КС-язык представим грамматикой в нестрогой нормальной форме Грейбах***

Пример. Грамматика простых арифметических выражений с устраненной левой рекурсией:

$$S \to TU$$
$$U \to + TU \mid \lambda$$
$$T \to FV$$
$$V \to * FV \mid \lambda$$
$$F \to (S) \mid a$$

После ее преобразования к нестрогой нормальной форме Грейбах получим:

$$S \to (S)VU \mid aVU$$
$$U \to + TU \mid \lambda$$
$$T \to (S)V \mid aV$$
$$V \to * FV \mid \lambda$$
$$F \to (S) \mid a$$

Детерминированный LL-анализ

КС-грамматика в нестрогой нормальной форме Грейбах допускает детерминированный LL-анализ, если для каждой группы порождающих правил с одним и тем же нетерминалом в левой части правые части будут однозначно различимы по нескольким первым терминальным символам.

LL(1)-анализатор выполняет LL(1)-анализ, если правые части правил грамматики различимы по одному первому терминальному символу.

Преобразование порождающих правил: факторизация

если в грамматике есть правила вида:

$$A \to a\beta\omega_1 \mid a\beta\omega_2 \mid \ldots \mid a\beta\omega_n \qquad (*)$$

Добавим в грамматику нетерминал B и вместо правил (*) включим в грамматику следующие правила:

$$A \to a\beta B \qquad (**)$$

$$B \to \omega_1 \mid \omega_2 \mid \ldots \mid \omega_n \qquad (***)$$

Затем получившиеся правила (***) следует преобразовать к нестрогой нормальной форме Грейбах.

Если это удастся, то для преобразованной грамматики становится возможным LL(1)-анализ.

Построение LL(1)-анализатора

Пусть входная цепочка символов всегда заканчивается символом-ограничителем $\perp$.

Таблица LL(1)-анализатора: столбцы помечены терминальными символами (включая $\perp$), а строки – нетерминалами преобразованной грамматики.

Для каждого из правил вида $A \to a\gamma$, правая часть заносится на пересечение строки, помеченной нетерминалом A, и столбца, помеченного терминалом a. Если же правая часть правила пустая, то во все клетки строки, не занятые другими правилами, записывается λ.

До начала работы в магазин записывается ограничитель $\perp$, а затем начальный нетерминал.

На каждом шаге анализатор проверяет, допустим ли очередной входной символ, и если да, то выполняет одно из двух действий:

1) если в вершине магазина нетерминал, то, в зависимости от того, каков очередной входной символ, этот нетерминал заменяется символами правой части соответствующего правила, причем символы записываются в обратном порядке. Если для очередного входного символа в таблице записано λ, то нетерминал удаляется из магазина, а если в таблице пустая клетка, то анализатор прекращает цикл из-за ошибочного символа во входной цепочке;

2) если в вершине магазина терминал, то он сравнивается с очередным входным символом. При совпадении терминал удаляется из магазина и делается переход к следующему символу входной цепочки. При несовпадении анализатор прекращает цикл из-за ошибочного символа во входной цепочке.

Цикл завершается, когда входная цепочка символов оказалась вся просмотрена, т.е. на входе – символ $\perp$. Если при этом в магазине символ $\perp$, то входная цепочка символов считается распознанной, если нет – то нераспознанной.

Пример. Грамматика простых арифметических выражений, преобразованная к нестрогой нормальной форме Грейбах:

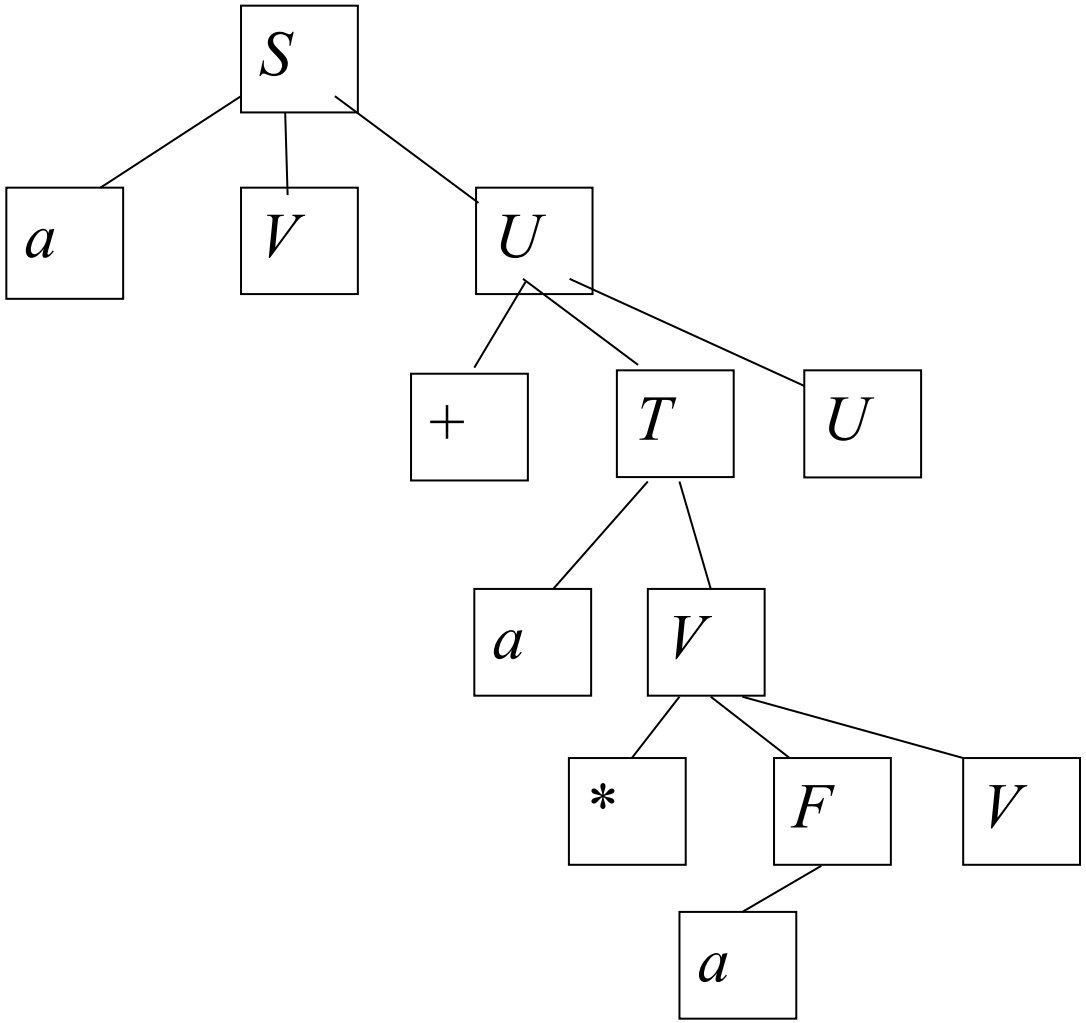
$$S \to (S)VU \mid aVU$$
$$U \to + TU \mid \lambda$$
$$T \to (S)V \mid aV$$
$$V \to * FV \mid \lambda$$
$$F \to (S) \mid a$$

	$+$	$*$	$($	$)$	a	$\perp$
S			$(S)VU$		aVU	
U	$+TU$	λ	λ	λ	λ	λ
T			$(S)V$		aV	
V	λ	$*FV$	λ	λ	λ	λ
F			(S)		a	

Пример. Анализ цепочки $a + a * a \perp$

№ шага	Вход	Магазин	Правило
1	$a + a * a \perp$	$S \perp$	$S \to aVU$
2	$a + a * a \perp$	$aVU \perp$	
3	$+ a * a \perp$	$VU \perp$	$V \to \lambda$
4	$+ a * a \perp$	$U \perp$	$U \to + TU$
5	$+ a * a \perp$	$+ TU \perp$	
6	$a * a \perp$	$TU \perp$	$T \to aV$
7	$a * a \perp$	$aVU \perp$	
8	$* a \perp$	$VU \perp$	$V \to * FV$
9	$* a \perp$	$* FVU \perp$	
10	$a \perp$	$FVU \perp$	$F \to a$
11	$a \perp$	$aVU \perp$	
12	$\perp$	$VU \perp$	$V \to \lambda$
13	$\perp$	$U \perp$	$U \to \lambda$
14	$\perp$	$\perp$	

На рисунке изображено ***неявно строящееся*** при анализе дерево порождения цепочки: *a* + *a* * *a*.



Пример. Анализ ошибочной цепочки $a\ )\ \perp$

№ шага	Вход	Магазин	Правило
1	$a\ )\ \perp$	$S\ \perp$	$S \rightarrow aVU$
2	$a\ )\ \perp$	$aVU\ \perp$	
3	$)\ \perp$	$VU\ \perp$	$V \rightarrow\ \lambda$
4	$)\ \perp$	$U\ \perp$	$U \rightarrow \lambda$
5	$)\ \perp$	$\perp$	<ОШИБКА>