Глава 3. Нисходящий синтаксический анализ

Фаза синтаксического анализа реализуется частью компилятора, которая называется *синтаксическим анализатором* (или *парсером*). Парсер исследует последовательность токенов, формируемую лексическим анализатором для проверки, соответствует ли она синтаксису языка, создавая промежуточное представление (в общем случае древовидное в виде дерева разбора), описывающее грамматическую структуру потока токенов. На практике редко требуется явное построение дерева разбора, оно больше носит концептуальный характер.

Формальной основой синтаксического анализа являются КС-грамматики. Входной поток токенов рассматривается как строка языка, состоящая из токенов. Для КС-грамматики, описывающей синтаксис языка, токены (точнее, коды токенов, идентифицирующие соответствующие лексические классы) рассматриваются как терминалы. Множество терминалов КС-грамматики – это множество лексических классов (токенов). Парсер должен определить, принадлежит ли входная строка данному языку.

На практике в компиляторах в основном применяют нисходящие (предиктивные) и восходящие методы синтаксического анализа. В настоящей главе рассматриваются нисходящие методы, восходящие методы будут рассмотрены в главе 4.

Предварительно рассмотрим общие вопросы распознавания КС-языков.

3.1. Автомат с магазинной памятью

Распознавателем КС-языка является *МП-автомат (магазинный автомат*), эквивалентный конечному автомату, к которому добавлена память магазинного типа (стек).

В функции МП-автомата входит:

а) чтение входного символа, замещение верхнего символа стека строкой символов (возможно пустой) и изменение состояния или

б) все то же самое, но без чтения входного символа.

Формально МП-автомат определяется как семерка $M = (K, T, \Gamma, \delta, k_0, Z_0, F)$, где K – конечное множество состояний; T – конечный входной алфавит; Γ – конечный *стековый* (*магазинный*) алфавит; $\delta : K \times (T \cup \{\varepsilon\}) \times \Gamma \to 2^{K \times \Gamma^*}$ – функция переходов, которая отображает множество $K \times (T \cup \{\varepsilon\}) \times \Gamma$ на множество конечных подмножеств множества $K \times \Gamma^*$; $k_0 \in K$ – начальное состояние автомата; $Z_0 \in \Gamma$ – начальный символ стека, который первоначально находится в вершине стека; $F \subseteq K$ – множество конечных (заключительных, финальных, принимающих) состояний.

Запись функции $\delta(k, a, A) = (k', \gamma)$, где $k, k' \in K$, $a \in T \cup \{\varepsilon\}$, $A \in \Gamma$, $\gamma \in \Gamma^*$ означает, что в текущем состоянии автомата k с элементом A в вершине стека при чтении символа a осуществляется переход в состояние k' и в стек вместо элемента A заносится строка γ (элемент A замещается строкой γ).

Конфигурация МП-автомата в любой момент времени описывается тройкой $(k, x, \alpha) \in K \times T^* \times \Gamma^*$, где $k \in K$ – текущее состояние автомата; $x \in T^*$ – необработанная часть входной строки (первый символ строки x является очередным входным символом, если $x = \varepsilon$, то считается, что входная строка полностью прочитана); $\alpha \in \Gamma^*$ – текущее содержимое стека, причем самый левый символ строки α считается верхним символом стека (если $\alpha = \varepsilon$, то стек считается пустым).

Такт работы МП-автомата определяется бинарным отношением $\vdash$, определенным на множестве конфигураций. Если $(k', \gamma) \in \delta(k, a, A)$, где $k, k' \in K$, $a \in T$, $A \in \Gamma$, $\gamma \in \Gamma^*$, автомат переходит из конфигурации $c = (k, ax, A\alpha)$, $x \in T^*$, $\alpha \in \Gamma^*$ в конфигурацию $c' = (k', x, \gamma\alpha)$, т. е. $(k, ax, A\alpha) \vdash (k', x, \gamma\alpha)$. Если $\gamma = \varepsilon$, то просто удаляется символ из вершины стека.

Если $(k', \gamma) \in \delta(k, \varepsilon, A)$, то $(k, ax, A\alpha) \vdash (k', ax, \gamma\alpha)$. т. е. в этом такте, называемом ε-*тактом*, автомат переходит из одной конфигурации в другую без чтения входного символа (ε-*переход*). ε-такты возможны и в случае, если входная строка полностью прочитана, но если стек пуст, следующий такт невозможен.

Начальной конфигурацией МП-автомата называется конфигурация вида (k_0, x, Z_0), т. е. автомат находится в начальном состоянии k_0, на входе автомата распознаваемая строка x, стек содержит начальный символ Z_0.

Заключительной конфигурацией МП-автомата называется конфигурация вида $(k_f, \varepsilon, \alpha)$, где $k_f \in F$ – одно из конечных состояний, входная строка прочитана до конца, а в стеке содержится заранее определенная строка $\alpha \in \Gamma^*$ (часто $\alpha = \varepsilon$).

Можно определить транзитивное ($\vdash^+$) и рефлексивно-транзитивное ($\vdash^*$) замыкания отношения $\vdash$. Запись $c \vdash^+ c'$ означает, что конфигурация c' достижима (выводима) из конфигурации c за один или более тактов, а запись $c \vdash^* c'$ – за нуль или более тактов.

Говорят, что строка $x \in T^*$ *допускается* (*принимается*) МП-автоматом $M = (K, T, \Gamma, \delta, k_0, Z_0, F)$, если $(k_0, x, Z_0) \vdash^* (k_f, \varepsilon, \alpha)$ для некоторых $k_f \in F$ и $\alpha \in \Gamma^*$, т. е. если в результате чтения входной строки автомат перейдет из исходной конфигурации в заключительную. Языком $L(M)$, *определяемым* (*допускаемым*) МП-автоматом M, называется множество строк, допускаемых этим автоматом.

МП-автомат называется *детерминированным*, если выполняются следующие условия:

а) функция вида $\delta(k, a, A)$ имеет не более одного элемента;

б) функция вида $\delta(k, \varepsilon, A)$ имеет не более одного элемента;

в) если $\delta(k, \varepsilon, A) \neq \varnothing$, то $\delta(k, a, A) = \varnothing$ для любого $a \in T$, т. е. если из некоторой конфигурации можно осуществить хотя бы один ε-переход, то он является единственным переходом, который можно осуществить из этой конфигурации.

Другими словами, если из любой конфигурации возможен единственный переход, МП-автомат является детерминированным, в противном случае – *недетерминированным*.

3.2. Автоматы с магазинной памятью и контекстно-свободные грамматики

Справедливо следующее утверждение: для любого КС-языка существует недетерминированный МП-автомат, который принимает его, и наоборот – если некоторый МП-автомат принимает некоторый язык, то этот язык является КС-языком.

Рассмотрим вопросы построения МП-автомата, распознающего КС-язык, заданный КС-грамматикой $G = (V_T, V_N, P, S)$. МП-автомат $M = (K, T, \Gamma, \delta, k_0, Z_0, F)$, принимающий данный язык, определяется следующим образом:

$K = \{k\}$, т. е. МП-автомат имеет единственное состояние k;

$T = V_T$, т. е. входной алфавит МП-автомата совпадает с множеством терминалов КС-грамматики;

$\Gamma = V_T \cup V_N$, т. е. стековый алфавит образуется объединением множеств терминалов и нетерминалов;

$k_0 = k$;

$Z_0 = S$, т. е. начальным символом стека является начальный нетерминал грамматики;

$F = \{k\}$;

т. е. $M = (\{k\}, V_T, V_T \cup V_N, \delta, k, S, \{k\})$, где функция переходов δ определяется следующим образом:

$\delta(k, \varepsilon, A) = \{(k, \alpha) \mid A \to \alpha \in P\}$ для всех $A \in V_N$;

$\delta(k, a, a) = \{(k, \varepsilon)\}$ для всех $a \in V_T$.

Такой МП-автомат эмулирует левосторонний вывод. При каждом такте, выполняемом автоматом, из стека извлекается один символ. Если извлеченный символ оказывается нетерминалом, то ему в соответствие ставится продукция, правая часть которой в таком случае заносится в стек. Если же извлеченный из стека символ оказывается терминалом, то он используется в качестве входного символа и, следовательно, определяет следующий такт автомата.

Определим МП-автомат для КС-грамматики, порождающей КС-язык $L = \{a^n b^n c^m \mid n, m \geq 1\}$,

$S \to TC$

$T \to aTb \mid ab$

$C \to cC \mid c$

МП-автомат $M = (\{k\}, \{a, b, c\}, \{a, b, c, S, T, C\}, \delta, k, S, \{k\})$, распознающий данный язык, имеет следующие функции переходов:

$\delta(k, \varepsilon, S) = \{(k, TC)\}$;

$\delta(k, \varepsilon, T) = \{(k, aTb), (k, ab)\}$;

$\delta(k, \varepsilon, C) = \{(k, cC), (k, c)\}$;

$\delta(k, a, a) = \{(k, \varepsilon)\}$ для всех $a \in \{a, b, c\}$.

При анализе входной строки *aabbc* автомат может выполнить последовательность тактов, приводящую к заключительной конфигурации:

$(k, aabbc, S) \vdash (k, aabbc, TC)$
$\vdash (k, aabbc, aTbC)$
$\vdash (k, abbc, TbC)$
$\vdash (k, abbc, abbC)$
$\vdash (k, bbc, bbC)$
$\vdash (k, bc, bC)$
$\vdash (k, c, C)$
$\vdash (k, c, c)$
$\vdash (k, \varepsilon, \varepsilon)$.

Приведенная последовательность тактов соответствует левосторонней схеме вывода $S \Rightarrow TC \Rightarrow aTbC \Rightarrow aabbC \Rightarrow aabbc$.

Табличное задание функции переходов

$\delta(k, \varepsilon, S) = \{(k, TC)\}$;

$\delta(k, \varepsilon, T) = \{(k, aTb), (k, ab)\}$;

$\delta(k, \varepsilon, C) = \{(k, cC), (k, c)\}$;

$\delta(k, a, a) = \{(k, \varepsilon)\}$ для всех $a \in \{a, b, c\}$.

	a	b	c	ε
S				k, TC
T				k, aTb k, ab
C				k, cC k, c
a	k, ε			
b		k, ε		
c			k, ε	
ε				stop

Из недетерминированности автомата следует, что построенный на его основе синтаксический анализатор КС-языка в процессе функционирования может осуществлять возврат к предыдущей конфигурации. Это означает, что, обнаружив ошибку выбора, необходимо вернуться к моменту осуществления выбора, вновь осуществить выбор и выполнить другой переход. Однако если на порождающую язык грамматику наложить определенные ограничения, то можно построить эффективный детерминированный синтаксический анализатор для такого языка.

Язык, принимаемый детерминированным МП-автоматом, называется *детерминированным*. Не всякий КС-язык детерминированный, и такие языки не могут анализироваться детерминированным образом. Тем не менее детерминированные языки составляют очень важный класс языков, поскольку для них значительно упрощается решение задачи анализа. Большинство языков программирования являются детерминированными или почти таковыми. Некоторые языки можно разбирать детерминированно с помощью только одного из методов грамматического разбора.

Важно отметить, что в отличие от конечных автоматов в общем случае нельзя преобразовать недетерминированный МП-автомат в эквивалентный детерминированный МП-автомат. Это объясняется тем, что детерминированные КС-языки составляют только подкласс КС-языков. Для недетерминированного КС-языка невозможно построить детерминированный МП-автомат, принимающий этот язык.