УДК 004.4'413

О структурировании синтаксических диаграмм

С. З. Свердлов, А. А. Хивина

Доказана теорема структурирования для синтаксических диаграмм, утверждающая, что произвольную синтаксическую диаграмму можно заменить эквивалентным (порождающим тот же язык) набором диаграмм, использующих только параллельное и последовательное соединения ветвей.

При исследовании, разработке, спецификации, реализации и изучении формальных языков, в частности языков программирования, широко используются синтаксические диаграммы [2, 5]. Ни рис. 1 показан пример синтаксической диаграммы для нетерминала «слагаемое», используемого в грамматике языка программирования Паскаль [2].

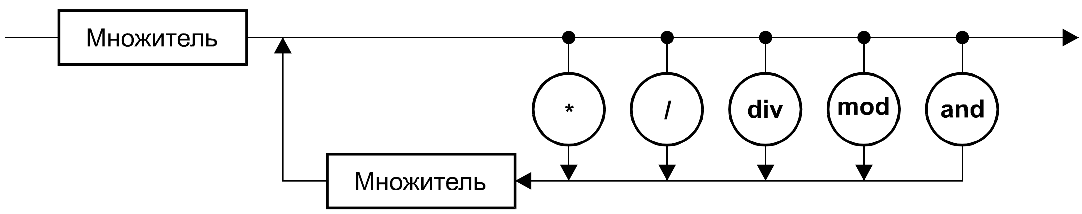

Рис. 1. Синтаксическая диаграмма слагаемого (язык Паскаль)

В дальнейшем нас будут интересовать синтаксические диаграммы контекстно-свободных (КС) грамматик и языков. Грамматики именно этого типа широко используются в практике спецификации языков программирования. Правила КС-грамматики имеют вид

$$A \to \alpha,$$

где A — нетерминальный символ; α — цепочка (возможно пустая) терминальных и нетерминальных символов.

Синтаксические диаграммы КС-языка могут быть построены по соответствующей формальной грамматике по следующему алгоритму.

Алгоритм 1

1. Для каждого нетерминала грамматики строится отдельная диаграмма, обозначенная названием этого нетерминала.
2. Нетерминалы из правых частей правил изображаются на диаграммах прямоугольниками, внутри которых записывается название нетерминала. Терминальные символы изображаются в кружках или овалах.
3. Для каждой правой части правила строится ветвь, представляющая собой последовательно соединенные прямоугольники и круги (овалы), следующие в том же порядке слева направо, что и соответствующие нетерминалы и терминалы правой части правила.
4. Ветви, соответствующие альтернативным правым частям правил для одного нетерминала, соединяются параллельно и образуют диаграмму для данного нетерминала.

В действительности синтаксические диаграммы строятся часто не по имеющейся грамматике, а служат самостоятельным средством спецификации языков, в том числе и языков программирования. При этом язык определяется совокупностью диаграмм.

При построении программ-распознавателей по синтаксическим диаграммам могут быть использованы шаблоны, соответствующие типовым фрагментам диаграмм. Один из возможных наборов таких типовых фрагментов показан на рис.2.

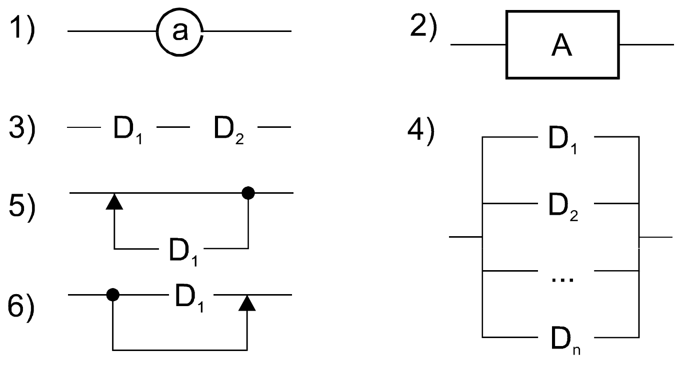

Рис. 2. Типовые фрагменты синтаксических диаграмм.
1) терминальный символ; 2) нетерминальный символ;
3) последовательное соединение; 4) параллельное соединение;
5) повторение ноль или более раз; 6) повторение один или более раз

При рассмотрении набора типовых фрагментов возникают несколько вопросов:

1. Достаточен ли такой набор для построения диаграмм произвольного КС-языка?
2. Можно ли произвольную синтаксическую диаграмму заменить эквивалентной (порождающей тот же язык) диаграммой или эквивалентным набором диаграмм, составленных только из типовых фрагментов?
3. Если ответ на вопрос 2 утвердительный, то нельзя ли обойтись меньшим набором?

Ответ на первый вопрос следует из приведенного выше алгоритма построения синтаксической диаграммы по КС-грамматике. Достаточно параллельного и последовательного соединения.

Ответ на вопросы 2 и 3 не очевиден. Постановка этих вопросов применительно к синтаксическим диаграммам подобна известной проблеме структурирования программ. Теорема структурирования для программ [1, 3, 5], доказанная в [1] утверждает:

Для любой программы, выраженной блок-схемой с одним входом и одним выходом, можно построить эквивалентную программу, то есть выполняющую те же преобразования исходных данных в результаты с помощью тех же вычислений, единственными управляющими структурами в которой являются цепочка и цикл с предусловием.

Исследуем возможность структурирования синтаксических диаграмм.

Теорема 1. *Для произвольной синтаксической диаграммы может быть построена эквивалентная, то есть порождающая тот же язык, контекстно-свободная грамматика.*

Из теоремы 1 и алгоритма 1 следует

Теорема 2. *Для произвольной синтаксической диаграммы может быть построен эквивалентный, то есть порождающий тот же язык, набор диаграмм, содержащих только параллельное и последовательное соединения.*

Действительно, построив КС-грамматику, эквивалентную исходной синтаксической диаграмме (теорема 1), можно по алгоритму 1 получить синтаксические диаграммы, использующие только последовательное и параллельное соединения.

Доказательство теоремы 1. Назовем узлами синтаксической диаграммы точки разветвления и точки соединения линий диаграммы, а также вход и выход диаграммы, если они не совпадают с точками ветвления или соединения.

Участок диаграммы, соединяющий два узла, содержащий ноль или более последовательно соединенных овалов (терминалы) и прямоугольников (нетерминалы) и не содержащий других узлов назовем ветвью. Пронумеруем ветви диаграммы, поставив в соответствие номеру каждой ветви цепочку (возможно, пустую) терминальных и нетерминальных символов, расположенных последовательно вдоль этой ветви.

Узел, соответствующий входу диаграммы (начальный узел) для нетерминала N обозначим N. Узел, соответствующий выходу диаграммы пометим символом ε. Остальные узлы диаграммы обозначим произвольными символами, не совпадающими с терминалами и нетерминалами исходной диаграммы.

После принятых соглашений произвольной синтаксической диаграмме можно сопоставить направленный граф, вершины которого соответствуют узлам диаграммы, а пронумерованные однонаправленные ребра — ветвям. При этом одна из вершин графа (начальная) помечена символом N, конечная вершина — символом ε.

Любому пути от входа к выходу синтаксической диаграммы соответствует путь от начальной до конечной вершины на полученном графе. При этом любой путь на графе однозначно определен последовательностью номеров проходимых ребер.

Запишем «правила прохождения» рассматриваемого графа следующим образом. Количество правил равно количеству ребер. Для ребра с номером k, идущего из вершины A в вершину B, правило будет выглядеть так:

$$A \to kB$$

Значения k во всех правилах различны и могут выступать в роли номеров правил. Полученный набор правил прохождения графа может рассматриваться как множество продукций [4] автоматной грамматики G, в которой набор терминальных символов есть множество номеров ребер, множество нетерминалов включает обозначения всех вершин кроме конечной, начальным нетерминалом является N, а ε обозначает пустую цепочку.

Каждому пути от начальной до конечной вершины графа можно поставить в соответствие последовательность номеров правил грамматики G, определяющая вывод из начального нетерминала N грамматики G терминальной цепочки, состоящей из номеров правил и совпадающей с последовательностью номеров ребер, проходимых на данном пути в графе. Заметим, что последним применённым правилом всегда будет правило вида $A \to k\varepsilon$, или, что то же самое, $A \to k$.

Каждому пути от начальной до конечной вершины в графе соответствует предложение языка, порождаемого грамматикой G и представляющее собой последовательность номеров ребер, определяющую рассматриваемый путь. И наоборот, любая последовательность номеров, порождаемая грамматикой G, однозначно соответствует пути из начальной в конечную вершину графа.

Заменяя в правилах грамматики G номера ребер на соответствующие этим номерам цепочки (возможно пустые) терминалов и нетерминалов, расположенных в соответствующих ветвях исходной синтаксической диаграммы, получим КС-грамматику G^*, удовлетворяющую теореме 1.

Теорема 1 доказана.

Литература

1. *C. Böhm, G. Jacopini*, "Flow diagrams, Turing Machines and Languages with only Two Formation Rules", Comm. of the ACM, 9(5): 366-371,1966.

2. *Йенсен К., Вирт Н.* Паскаль. Руководство для пользователя и описание языка/Пер. с англ., предисл. и послесл. Д. Б. Подшивалова. — М.: Финансы и статистика, 1982. — 151 с., ил.
3. *Лингер Р. Миллс., Уитт Б.* Теория и практика структурного программирования: Пер. с англ. М.: Мир, 1982. — 406 с., ил.
4. *Рейуорд-Смит В. Дж.* Теория формальных языков. Вводный курс: Пер. с англ. — М.: радио и связь, 1988. — 128 с.: ил.
5. *Свердлов С. З.* Языки программирования и методы трансляции: Учебное пособие. — СПб.: Питер, 2007. — 638 с.: ил.

Сведения об авторах

1. Свердлов Сергей Залманович, 1954 г.р., к.т.н., доцент. Область научных интересов: программирование, компьютерные технологии, языки программирования и их реализация, цифровая фотография и обработка изображений. Тел. +7 921-122-74-43
2. Хивина Анна Александровна, 1984 г.р., математик, системный программист. Область научных интересов: компьютерные технологии.