

ОБ ИСПОЛЬЗОВАНИИ СИНТАКСИЧЕСКИ-ОРИЕНТИРОВАННЫХ МЕТОДОВ  
ТРАНСЛЯЦИИ В СИСТЕМЕ АВТОМАТИЗАЦИИ ПРОГРАММИРОВАНИЯ

Специализированные программные системы (ПС), позволяющие автоматизировать процесс разработки программного обеспечения, получили название инструментального комплекса программирования и отладки (ИКПО). При построении подобных систем важнейшей задачей является выбор класса языков и связанного с ним метода анализа, обеспечивающего требуемое представление синтаксических структур. В разработанном на ЭВМ М-4030 [1] ИКПО применяются синтаксически-ориентированные методы трансляции (СОМТ), в частности  $LR(K)$  грамматики. Использование  $LR(K)$  грамматик позволяет обеспечить технологичность производства ПС за счет проблемной и синтаксической ориентации структур. Возможность декомпозиции синтаксических структур на подструктуры позволяет произвести структуризацию программы МП автомата (МПА), полученного по исходной  $LR(K)$  грамматике, выделить уровни семантики и произвести поуровневую отладку ПС на интерпретаторе. В этом случае семантические подпрограммы обычно связывают с аксиомами подграмматик. При структуризации облегчается процесс внесения изменений в систему, что значительно ускоряет отладку.  $LR(K)$  грамматики обеспечивают требуемую универсальность ИКПО, так как ныне это наиболее широкий класс грамматик, для которых доказана возможность создания алгоритма построения анализатора (МПА) данного языка [2]. Важными свойствами  $LR(K)$  анализатора является устойчивость к рекурсивным конструкциям и простота функционирования. Недостатком применения СОМТ считается сложность формального описания семантики, что и ограничивает семантическую универсальность ИКПО. В данном ИКПО это решается созданием библиотеки семантических подпрограмм анализа и генерации (БСПАГ) и библиотеки рабочих подпрограмм (БРП).

В настоящей работе рассматриваются вопросы автоматизации процесса преобразования исходной грамматики в соответствующий анализатор. Частично эта задача была решена в указанном ИКПО реализацией модифицированного алгоритма Кнута-Эрли построения программ

$LR(K)$  анализаторов (генератор МПА). Кроме того, в состав ИКПО входит интерпретатор МПА. Недостатком указанного ИКПО являлось наличие неавтоматизированного этапа оптимизации полученной программы автомата [2, 3, 5]. Ниже приводится новая модификация данного метода, позволяющая вести оптимизацию по числу состояний на этапе генерации МПА.

Будем рассматривать грамматики  $LR(I)$ , эквивалентные в отношении соответствующих языков остальным  $LR(K)$  при  $K > 1$  на основании теорем эквивалентности [4]. По определению,  $LR(1)$  грамматика есть четверка  $G = (V_N, V_T, S, P)$ , где  $V_N$  и  $V_T$  - нетерминальный и терминальный алфавиты;  $S$  - аксиома грамматики;  $P$  - множество пар цепочек из  $V_N \cup V_T$  вида:

$$\{ \varphi_i \rightarrow \psi_{ij} \mid i = \overline{1, N}, j = \overline{1, K_i}, \varphi_i \in V_N, |\varphi_i| = 1, \psi_{ij} \in V_N \cup V_T \}.$$

Здесь  $N$  - число подграмматик;

$K_i$  - число правил в подграмматике  $\varphi_i$ .

Основной подграмматике  $\varphi_i$  назовем цепочки вида  $\{ \varphi_i \rightarrow \psi_{ij} - \psi_{ij} \in V_T \}$ . Символы алфавита  $V_N$ , для которых на данном этапе генерации найдена основа, назовем условно-терминальными (включенными в  $V_T'$ ).

Условной основой подграмматике  $\varphi_i$  назовем цепочки вида

$$\{ \varphi_i \rightarrow \psi_{ij} \dots \psi_{ij} \in V_T \cup V_T' \}.$$

Рассмотрим функционирование оптимизатора на примере определения числа состояний МПА. Пусть для подграмматике  $\varphi_i$  найдена условная основа  $\varphi_i \rightarrow \psi_{ij}, j = \overline{1, K_i}$ . Алгоритм определения условной основы состоит в просмотре  $\psi_{ij}$ -ых частей и переходе с возвратом (после определения основы) на подграмматику, определяемую встречным нетерминалом. Тогда справедливо соотношение

$$f(\varphi_i) = 1 + \sum_{j=1}^{K_i} (1 + |\psi_{ij}|) + \sum_{i=1}^{T_i} f(\varphi_i) - (L^{(1)} + L^{(2)} + L^{(3)}), \varphi_i \in \varphi_i,$$

где  $f(\varphi_i)$  определяет число состояний МПА для  $\varphi_i$ ;  $T_i$  - число  $\varphi_i \in V_N$  таких, что  $\varphi_i \in \varphi_i$ ;  $L^{(1)}, L^{(2)}, L^{(3)}$  - управляющие функции оптимизатора, имеют смысл:  $L^{(1)} = \ell - 1$ ;  $\ell$  - количество правил одинаковой длины;  $L^{(2)}$  - число правил вида  $\varphi_i \rightarrow \varphi_i \dots$  (левая рекурсия);  $L^{(3)} = \sum_{m=1}^{M_i} (K_m - 1) + \sum_{m=2}^{M_i} (n_m - 1)$ , где  $m$  - номер позиции в правиле;  $K_m$  - число правил, имеющих одинаковое начало;  $n_m$  - окончание, но при условии, что  $m_2 > m_1$  и правила имеют одинаковую длину.

Рассмотрим теперь отличия процесса генерации МПА от описанного в работе [5]. Пусть строится МПА для  $\varphi_t^j \rightarrow \psi_{ij}, j=1, N_t$ . Каждое правило при этом просматривается сразу до конца и, если возможно, происходит добавление правила в уже имеющиеся состояния. Заранее определенная условная основа гарантирует, что сгенерированы МПА подграмматик от  $\varphi_t^j$  и при встрече  $\varphi_t^j \in V_N \cup V_T$  происходит наращивание текущего состояния  $Z_{tm}^{j-1}$  (под символами позиции  $m$  правила  $j$ ) командами начального состояния  $Z_{i1}^{ki}$  МПА по  $\varphi_t^j$ :

$$Z_{tm}^j = O(Z_{tm}^{j-1} \cup Z_{i1}^{ki} \cup \{ \varphi_t^j \rightarrow x_0 \varphi_t^j y \})$$

для символа  $\alpha \in V_T$ :

$$Z_{tm}^j = O(Z_{tm}^{j-1} \cup \{ \varphi_t^j \rightarrow x_0 \alpha y \}),$$

где подпочка  $x, y \in V_N \cup V_T, |x| = m-1, O$  - оператор оптимизации, компанующий состояния в соответствии с функциями управления  $L^{(n-s)}$  по критерию  $f(S) \rightarrow \min$  и ограничениям по сохранению свойств грамматики. Оптимизатор либо добавляет новые команды в  $Z_{tm}^{j-1}$  (тогда осуществляется переход к генерации следующего состояния, определяемого из правил  $\varphi_t^j \rightarrow \dots \circ y$ ), либо использует готовые команды состояния. Генерация МПА завершается удалением вспомогательных состояний  $Z_{i1}^{ki}$  для  $i=2, N$ . Состояние  $Z_{11}^1$  обозначается как первое и на него передается управление при интерпретации.

Для реализации данного алгоритма выбран технологический комплекс  $RTK$  ОС ЕС [6], применение которого обусловлено особенностями  $R$ -трана (входной язык  $RTK$ ).

Разработанный ИКПО используется для построения диалоговой системы сбора и обработки данных экспериментальных исследований в операционной системе ДОС РВ АСВТ М-6000. С этой целью применялась параллельная ветвь ИКПО, реализованная в ОУС М-6000. По описанию языка диалога, предложенному пользователями системы, была составлена  $LR(1)$  грамматика и получена программа МПА. Затем автомат был проверен на интерпретаторе [7]. Программа МПА была дополнена семантическими признаками из БСПАГ и началась семантическая отладка подпрограмм ДС из БРП на интерпретаторе в ОУС М-6000. Функции интерпретатора при этом заключались в анализе входного текста запроса оператора, стандартном заполнении выходного буфера и вызове необходимых программ из БРП, реализующих тре-

бования оператора. Структуризация программы МПА позволяет вести параллельную отладку основных подпрограмм в комплексе. Модуль трансляции программы МПА, входящий в состав интерпретатора, позволяет получить оттранслированные программы МПА и сформировать рабочий вариант системы, исключая модули, не используемые при интерпретации.

МПА для ДС имеет 80 состояний и занимает 200 16-разрядных слов. Интерпретатор, как часть ИКПО и рабочий вариант ДС, совместно с БРП, занимают каждый около 7 К памяти.

Генерация данных по МПА требует последовательно связанных программ МПА интерпретации и генерации. МПА интерпретации позволяет ввести описания, необходимые для генерации. Задав длину буфера некоторым числом, можно автоматически получать строки, заведомо допустимые данным МПА (и, следовательно, системой), длинн которых будут представлять сходящуюся снизу к данному числу последовательность.

#### Л и т е р а т у р а

1. Г а м и н П.В., К у л и к о в В.В., Ш а м а ш о в М.А. Система автоматизации проектирования синтаксических анализаторов. Всесоюзное совещание по автоматизации проектирования трансляторов. Тезисы докладов. - Таллин: ТПИ, 1980.
2. К н у т Д. О переводе (трансляции) языков слева направо. Сб. переводов: Языки и автоматы. - М.: Мир, 1975.
3. Э р л и Д. Эффективный алгоритм анализа контекстно-свободных языков. Там же.
4. Л е м а н Д. LR(K) грамматики и детерминированные языки. Там же.
5. Г а м и н П.В., К у л и к о в В.В. Генерация программы синтаксического анализатора. Наст. сборник.
6. В е л ь б и ц к и й И.В., Х о д а к о в с к и й В.Н., Ш о л м о в Л.И. Технологический комплекс производства программ на машинах ЕС ЭВМ и БЭСМ-6. - М.: Статистика, 1980.
7. К у л и к о в В.В., Ш а м а ш о в М.А. Автоматизация проектирования синтаксических анализаторов проблемно-ориентированных языков систем автоматизации эксперимента. Наст. сборник.