

Грамматика с памятью для создания трансляторов с параллельных языков вычислительных систем

77-30569/235311

10, октябрь 2011

Руденко Ю. М.

УДК 519.685.3

МГТУ им. Н.Э. Баумана

kirur@bk.ru

Построение транслятора, удобного для применения в вычислительных системах (ВС), особенно в неоднородных, является актуальнейшей задачей. ВС характеризуются наличием большого количества различных языков программирования, а также их диалектов и типов вычислительных модулей. Такой транслятор должен легко настраиваться под используемый язык программирования или тип используемого вычислительно модуля. Идея такого транслятора изложена в работах [1, 2, 3]. Эта задача может быть решена с помощью предлагаемой в этой работе WR – грамматики. при использовании параметрического транслятора. Правила вывода WR - грамматик могут быть представлены в виде обобщенных синтаксических диаграмм (ориентированных графов), у которых на дугах заданы условия их прохождения (предикаты и действия над стеками и регистрами). На дугах WR - грамматик могут встречаться также метaperменные, заданные, в свою очередь, графами, и синтермы - синтаксически эквивалентные классы терминальных символов. Использование предикатов и действий над стеками и регистрами дает возможность представления синтаксиса широкого класса языков. Средствами WR - грамматики - использованием регистровой памяти - могут быть описаны такие элементы семантики, как вычисление типов выражений и согласование типов операндов и операций и т.д. Предложенные WR- грамматики близки W - грамматикам [3], имеющим в правилах вывода операции над стеками. Однако, для построения синтаксически управляемого транслятора более простой и удобной оказалась регистровая память. Операции над памятью введены непосредственно в синтаксис и существенно используются на этапе распознавания в транслирующей системе.

В первом разделе рассмотрен частный случай WR - грамматик: WR_0 - грамматика, в которой в отличие от WR -грамматики, рассмотренной во втором разделе, отсутствуют регистры и функции индикации синтермов.

1. WR_0 - грамматики.

Пусть L - язык с алфавитом термов $A = \{A_1, \dots, A_p\}$, и B - алфавитом синтермов $B = \{B_1, \dots, B_k\}$. Зададим функцию $F = \{F_1, \dots, F_k\}$, осуществляющую отображение $F_p : D_p \rightarrow B_p$, $B_p \in B$, $D_p \subseteq A$, $1 \leq p \leq k$. Определим алфавит нетерминальных символов V и вспомогательный алфавит R . $V = \{V_0, V_1, \dots, V_t\}$, $R = \{R_0, R_1, \dots, R_q\}$, такой что $V \subseteq R$, $R_0 = V_0$. При представлении WR -грамматики на графах R_i имеют смысл имен вершин графов, R_0 - начальной вершины (начальной аксиомы языка), R_γ - заключительные вершины, $\gamma \subseteq Q$.

Определим Π_0 - совокупность правил вывода WR_0 -грамматики вида:

1) $W_{i,j,e}^b$ - единичное правило с предикатом - синтермом $R_i R_j$, где $R_i, R_j \in R$, $B_e \in B$, означающее переход и вершины R_i к вершине R_j при наличии на ленте ввода Λ синтерма B_e , который при построении цепочки дописывается справа. Допускается правило вида $W_{i,j}$ с отсутствием синтерма, означающее безусловный переход из R_i и R_j без изменения ленты ввода Λ .

2) $W_{i,j,x}^v$ - единичное правило с предикатом - нетерминалом

$R_i \xrightarrow{R_j} V_x$, $R_i, R_j \in R$, $V_x \in V$, введем индикатор q стека S следующим образом: $q=0$ - стек не используется, $q=1$ - стек используется. Стек используется в WR_0 - грамматике, если V содержит хотя бы один нетерминал, отличный от V_0 . Через (S) обозначим содержимое S . Если грамматика WR_0 содержит правило вывода $W_{i,j,x}^v$, то из вершины R_i осуществляется переход к вершине V_x с одновременной записью в стек S вершины - приемника R_j . Далее, при попадании в вершину R_γ возможны два случая: $(S) = R_j$ или $(S) = 0$. В первом случае происходит переход в вершину R_j и из стека S удаляется R_j ; во втором случае вывод считается законченным, если содержимое ленты ввода $(\Lambda) = 0$ и предложение не принадлежит языку, если $(\Lambda) \neq 0$.

3) $W_{i,j,m}^B$ матричное правило с предикатами - синтермами, $J = (J_1, \dots, J_q)$, $M = (M_1, \dots, M_q)$, $W_{i,J,M}^B = \bigcup_{m=1}^Q W_{i,J_m,M_m}^B$, где каждое правило $W_{i,J,M}^B$ - это объединение единичных правил предикатов синтермов, для дуг исходящих из i -ой вершины с заданным набором синтермов B_{M1}, B_{M2}, B_{Mm} .

4) $W_{i,j,\Lambda}^V$ - матричное правило с предикатами - нетерминалами,

$$J = (J_1, \dots, J_Q), \quad \Lambda = (\Lambda_1, \dots, \Lambda_Q), \quad W_{i,j,\Lambda}^V = \bigcup_{m=1}^Q W_{i,j_m,\Lambda_m}^V, \text{ объединение правил пре-}$$

дикатов - нетерминалов, аналогично п. 3.

5) $W_{i,j,k,m,\Lambda}^B$ - матричное правило общего вида,

$$W_{i,j,k,m,\Lambda}^B = \left(\bigcup_{m=1}^Q W_{i,j_m,m_m}^B \right) \bigcup_{m=1}^Q \left(\bigcup_{m=1}^Q W_{i,j_m,\Lambda_m}^V \right),$$

которое для каждой дуги, исходящей из i -й вершины, содержит правила с предикатами - синтермами и предикатами - нетерминалами.

Определение 2. WR_0 - грамматикой назовем грамматику вида $WR_0 (A, B, F, V, R, \Pi_0, R_0, q)$.

Требование 1. Будем считать, что грамматика WR_0 удовлетворяет условию: из любой вершины $v_j \in v$, используя правила перехода Π_0 , можно попасть за конечное число переходов в заключительную вершину из γ .

Дадим определение W - грамматик. Пусть S - словарь синтермов. $v = \{o_0, v_i\}$, $i \in \Gamma$ - вспомогательный словарь, $\Gamma = \{1, \dots, k\}$. $\wp$ - множество имен конечных множеств (возможно пустых) подмножеств правил вида $g \xrightarrow{W_i(l_1, \dots, l_n)} R$ или $g \xrightarrow{W_i(l_1, \dots, l_n)} R$, где через $\xrightarrow{W_i(l_1, \dots, l_n)}$ обозначено многоместное отношение, означающее, что синтерм слева контактирует с синтермом левой части любого правила, имя которого указано справа. Точка справа соответствует заключительному выводу, $g \in S$, $R \in \wp$, $\wp = \{\emptyset, R_i\}$, $i \in \Gamma_1$, $\emptyset$ - номер пустого множества правил, $\Gamma_1 = \{1, \dots, k_1\}$, $i_j \in v_1 \bigcup \wp$, $j = 1, \dots, n$. Число n аргументов определяется числом стеков, используемых при интерпретации правил грамматики. При этом $I = 0$ задает многоместное отношение нулевого типа, которое определяет, что синтерм g может конкатенировать с синтермом из левых частей множества $\wp$ с именем R . Многоместное отношение первого типа ($I = 1$) означает, что синтерм g может конкатенировать с синтермом из левых частей R с одно-временной записью в стеки $l_1, \dots, l_n$ непустых слов. Многоместное отношение второго типа ($I=2$) означает, что синтерм g может конкатенировать с синтермом из левых частей R только при условии, что все вершины стеков $l_1, \dots, l_n$ совпадают со словами $v_1, \dots, v_n$.

Пусть $t \in T$ - терм, где T - множество термов, d - параметр, $\delta \in Z$, Z - множество индексов правил. Определим функции $\Phi(t, \delta)$, которая задает соответствие термов и синтермов.

Теорема 1. Пусть $W(T, S, Z, v, \Phi, M, M_0)$ – грамматика с *многоместным отношением нулевого типа*. Тогда $WR_0(A, B, F, V, R, \Pi_0, R_0, o) \sim W(A, B, Z, v_1, F, M, M)$, если $V = \{V_0\}$, $Z = \{1\}$, $v_1 = \{0\}$, с некоторыми M, M_0 .

Доказательство. Условие $q=0$ и $V = \{V_0\}$ означает многоместным отношением нулевого типа. Тогда очевидно, что в Π_0 содержатся только правила вида $W^B_{i,j,m}$. Условие $v_1 = \{0\}$ означает отсутствие стеков в грамматике W . Условие $Z = \{1\}$ означает, что функция $\Phi(t, \delta)$ не зависит от δ и может быть выбрана равной F . Через R_i , $i=0,1,\dots,d$ обозначим все правила $W^B_{i,j,m} \in \Pi_0$. Будем считать, что M_0 определяет множество правил R_0 , а $X = \bigcup_{i=1}^d R_i$. Из определения WR_0 , W и $M = \{0, X\}$, если $R = \{R_0, \dots, R_d\}$, следует, что множества заключительных вершин двух грамматик совпадают, поэтому, $L(WR_0) = L(W)$. Теорема доказана.

Теорема 2. Пусть $W(A, B, Z, v_1, F, M, M_0)$ – грамматика с *многоместными отношениями первого и второго типа*. Тогда $WR_0(A, B, F, V, R, \Pi_0, R_0, 1) \sim W(A, B, Z, v_1, F, M, M_0)$, если $Z = \{1\}$, $v_1 = \{S\}$, S – имя стека в грамматике WR_0 , с некоторыми M, M_0 .

Доказательство. Условие $Z = \{1\}$ означает, что $\Phi(t, \delta)$ не зависит от δ и может быть выбрана равной F . Все правила, не содержащие нетерминала, порождают совпадающие заключительные вершины аналогично теореме 1. Условие $q=1$ в WR_0 – грамматике означает наличие в Π_0 правил вывода вида $W^V_{i,j,x}$ и требует использования стека S . Заменяем правило $W^V_{i,j,x} \in \Pi_0$ для каждого i W_1 – выводом R_i : $B_1 \xrightarrow{W_1(R_i)} V_x$. Для всех заключительных вершин $R_\gamma \in R$ грамматики WR_0 добавим W_2 – выводы R_γ : $B_1 \xrightarrow{W_2(R_k)} R_k$, $k=0,\dots,d$, согласно требованию 1 обеспечивается возврат к вершинам R_γ ; $R_\gamma: B_1 \xrightarrow{W_2(0)} R_\gamma$, где B_1 – синтерм со значением пусто. При таких выборах W_1, W_2 выводов $L(WR_0) = L(W)$. Теорема доказана.

2. WR – грамматики общего вида

WR – грамматики отличаются от WR_0 – грамматик, и в силу теорем 1, 2 от W – грамматики использованием регистровой памяти в предикатах и действиях. При этом, проход по дуге правила вывода, имеющей предикат, осуществляется, если этот предикат принимает значение “истина”. Если дуга содержит действия над регистрами, то при проходе по ней эти действия выполняются. Определим $C = \{C_1, \dots, C_q\}$ – алфавит имен регистров. Введем операции $\langle op \rangle \in \{=, \neq, \leq, \geq, <, >\}$, где $\langle op \rangle$ описывает, например, операции отношения между величинами, а $\langle do \rangle$ – логические операции

дизъюнкции и конъюнкции. Зададим множество целых чисел $v = \{v_1, \dots, v_q\}$. Поставим в соответствие каждому $C_j \in C$ число $v_j \in v$. Это соответствие будем обозначать $(C_j) = v_j$, т.е. применительно к процессору – значение, записанное в счетчике C_j , есть v_j .

Пусть $\iota_{i,m}$, γ_i – целые положительные числа, $v_{i,m} \in v$, $m = 1, \dots, f$. Через h_i обозначим линейную комбинацию значений вида: $h_i = \pm \iota_{i,1} v_{i,1} \pm \dots \pm \iota_{i,f} v_{i,f}$.

Введем предикат: $ph_i = h_i < op > \gamma_i$, где i – номер предиката, определяющий набор чисел $\{\iota_{i,m}\}$, γ_i , набор $\{v_{i,m}\}$, γ_i и вид h_i и более сложный предикат: $px_j = ph_1 < do > \dots < do > ph_q$, $i = 1, \dots, q$, где j – номер предиката, определяющий набор номеров $i = i(j)$, предикатов ph_i и их число $q = q(j)$. Определим синтерм B_j как класс эквивалентных терминальных символов $\alpha_{i,j} \in A$, $\iota = 1, \dots, \mu$, $\mu = \mu(j)$. Введем целочисленную функцию χ_j (индикатор синтерма B_j), равную порядковому номеру терминала в синтерме B_j . Каждому индикатору χ_j поставим в соответствие имя регистра $C_j \in C$, со значением χ_j . Таким образом, функции χ_j могут входить в предикаты ph_i и px_i . Назовем действием над регистрами операцию $d_{i,j}$ засылки линейной комбинации h_i в регистр C_j , т.е. $d_{i,j} : h_i \Rightarrow C_j$, где $\Rightarrow$ обозначает операцию засылки. С помощью операции засылки, таким образом, можно изменить значение регистра, т.е. в результате выполнения этой операции $(C_j) = h_i$. Дополним правила вывода Π_0 правилами вида:

1. $W_{i,j} ph_k$ – единичное правило вида $R_i : \underline{ph_k} \rightarrow R_j$ с предикатом ph_k (переход из вершины R_i в вершину R_j , если ph_i принимает значение “истина”).
2. $W_{i,j} px_k$ – единичное правило вида $R_i : \underline{px_k} \rightarrow R_j$ с предикатом px_k .
3. Матричные правила, представляющие собой всевозможные объединения правил из Π_0 и правил $W_{i,j} ph_k$, $W_{i,j} ph_{k1}$, выходящих из одной вершины R_i .
4. Ко всем перечисленным правилам можно добавить выполнения действия $d_{m,n}$ при переходе по какой-либо дуге данного правила. Рассмотрим, например, правило $W_{i,j,t}^B$. Добавление $d_{m,n}$ позволяет получить правило $W_{i,j,t}^B d_{m,n}$ вида $R_i : \underline{B_t} \rightarrow R_j$, означающее переход из вершины R_i в вершину R_j при наличии на ленте Λ синтерма B_t и одновременном выполнении действия $d_{m,n}$. Полученную совокупность правил обозначим Π .

Определение 3. WR – грамматикой назовем грамматику вида $WR(A, B, F, V, C, R, \Pi, R_0, q)$.

3. Использование WR – грамматик для построения синтаксически управляемого транслятора.

Зададим языки $L_i, i=1, \dots, n$ с алфавитами терминальных символов A_i . Введем параметры: t – номер языка, N – признак атрибута, отвечающий за согласование типов операндов и операций, выражений, выбор типа процессора, и т.п. Пусть задан алфавит синтермов B такой, что $B_j = B_j(t, N) \in B, j=1, \dots, k, A = \bigcup_{i=1}^n A_i$.

Будем считать, что матрица – функция $F\{f_{i,j}\}$ осуществляет отображение $2^{A_1} * \dots * 2^{A_n}$ и $f_{t,j} = f_{i,j}(N)$, где t – номер языка, j – номер синтерма B_j , знак $*$ означает прямое произведение. Обозначим $\beta = B_1 * \dots * B_n$. Пусть $\phi, \aleph \in C$ алфавиту регистров, $t \Rightarrow \phi, N \Rightarrow \aleph$. Значение регистра ϕ считается фиксированным, регистр $\aleph$ может использоваться в вычислениях, как указано в разделе 2.

Определение 4. Параметризованной $\wp\mathcal{R}$ – грамматикой с параметрами t, N , назовем грамматику вида $\wp\mathcal{R}(A, \beta, F, V, C, R, \Pi, R_o, q)$, где $\phi, \aleph \in C$ и $t \Rightarrow \phi, N \Rightarrow \aleph$.

Пусть для языков $L_1, \dots, L_n$ заданы их $WR_{(t)}$ – грамматики. Тогда нетрудно показать, что существует такая грамматика $\wp\mathcal{R}$, что

$$WR_{(t)}(A_t, B_t, F_t, V_t, C_t, R_t, \Pi_t, R_{o,t}, q_t) \subseteq \wp\mathcal{R}(A, \beta, F, V, C, R, \Pi, R_o, q) \subseteq \bigcup_{t=1}^n$$

$WR_{(t)}(A_t, B_t, F_t, V_t, C_t, R_t, \Pi_t, R_{o,t}, q_t)$, где $q_t \in \{0, 1\}$, $q = \max q_t$ по $t, 1 \leq t \leq n$.

В частности можно взять $\beta = \bigcup_{t=1}^n B_t$, $F: 2^{A_1} * \dots * 2^{A_n} \longrightarrow B$ так, что

$$F = \{F_1, \dots, F_n\}, F_t: 2^{A_1} \longrightarrow B, C \subseteq \bigcup_{t=1}^n C_t, V \subseteq \bigcup_{t=1}^n V_t,$$

$$\Pi \subseteq \bigcup_{t=1}^n \Pi_t, R \subseteq \bigcup_{t=1}^n R_t.$$

Однако, реально, за счет параметризации и наложения близких синтаксических структур общее число правил параметризированной грамматики $\wp\mathcal{R}$ удастся существенно сократить для достаточно близких языков. При этом существенно используются дополнительные по сравнению с W – грамматикой типы памяти, предикаты и индикаторы WR – грамматик. В качестве примера использования грамматик $\wp\mathcal{R}$ можно привести построение синтаксиса предложений в стандарте Open_MP для языков ФОРТРАН, СИ, приведенные в работе [4] для процессоров первого и второго типов $N = \{1, 2\}$.

Для иллюстрации использования предлагаемого метода параметризации на рис. 1 представлен ориентированный граф с нагруженными дугами для зависящей от пара-

метра $t=\{1,2,3\}$ подключение системы Open_MP), которое зависит от символьных констант для языков СИ ($t=1$), ФОРТРАН ($t=2$) и НОРМА ($t=3$) и различных типов процессоров $N=\{1,2\}$.

Принятые обозначения:

#pragma omp - символьная константа, определяющая, что далее идут инструкции Open_Mp для языка СИ.

!\$omp - символьная константа, определяющая, что далее идут инструкции Open_Mp для языка ФОРТРАН.

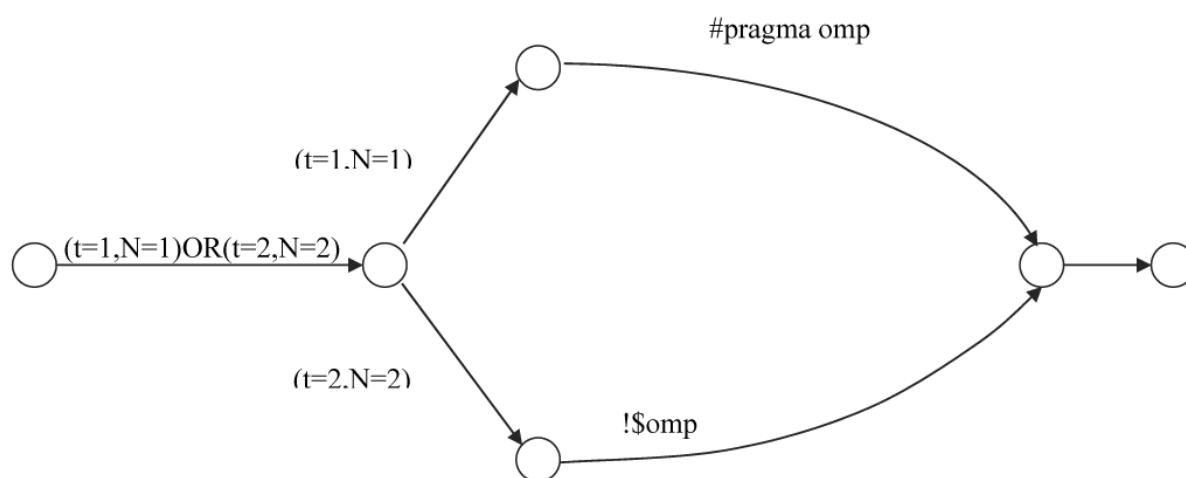


Рис. 1.

При прохождении первой дуги исключается язык НОРМА ($t=3$). Далее, по значению $t=1$ и значению символьной константы '#pragma omp' для первого типа процессора могут быть сформированы инструкции по выполнению параллельных операций. Аналогично, по значению $t=2$ и значению символьной константы '!\$omp' для второго типа процессора могут быть выполнены необходимые действия по подготовке параллельных вычислений

Таким образом, предлагаемый способ параметризации грамматик позволяет создавать объединенные трансляторы, которые затем можно адаптировать под требуемые языки и типы процессоров. Достаточно удобным предлагаемый метод может оказаться для создания транслятора, учитывающего диалекты языка. Особенно широкое применение параметрические трансляторы могут найти в вычислительных системах, объединяющих неоднородные вычислительные сети.

ЛИТЕРАТУРА

1. Руденко Ю.М. Требования к языкам программирования на современном этапе. - УСиМ, N 6, 1991 г., с. 74-79.
2. Параметрический транслятор для неоднородной вычислительной системы. Тезисы докладов на Международной научно-технической конференции, посвященной 30-летию со дня основания Университета, «Гражданская авиация на рубеже веков». Министерство транспорта РФ, государственная служба гражданской авиации, Московский Государственный технический университет гражданской авиации. 30-31 мая 2001 г., с. 247-248
3. Баша В.В., Руденко Ю.М. Использование параметрического подхода при решении проблем мобильности транслирующих систем. – УСиМ, 1988, №5, с. 46-51.
4. Антонов А.С. Введение в параллельные вычисления. Методическое пособие. – М.: Изд-во МГУ, 2002. – 72 с.: ил.

Grammar with memory for creating compilers for parallel languages of computing systems

77-30569/235311

10, October 2011

Rudenko Yu.M.

Bauman Moscow State Technical University

kirur@bk.ru

The article presents WR-grammar which is the basis for creation of a compiler which is adjusted to processing of programs written in various languages or their dialects and for various types of processors. Compilers of this type can be applied successfully in data processing computing systems. Necessity of creating compilers is due to existence of a lot of programming languages and their dialects focused on various types of processors, which causes certain difficulties at task solving on computing systems – even a very skilled programmer has difficulty solving them. Using a parametrical compiler, computer system administrator or operating system only has to do the following: to adjust the compiler to the required language or the language version and type of the processor by means of setting corresponding parameter values from a predefined list.

Publications with keywords: [WR-grammar](#), [processors](#), [parameter](#), [programming language](#)

Publications with words: [WR-grammar](#), [processors](#), [parameter](#), [programming language](#)

Reference

1. Rudenko Iu.M., Upravliaiushchie sistemy i mashiny 6 (1991) 74-79.
2. Basha V.V., Rudenko Iu.M., Upravliaiushchie sistemy i mashiny 5 (1988).46-51.
3. Antonov A.S., Moscow, Moscow State University Press, 2002, 72 p.