УДК 004.414.3

О.С. САВЕНКО, Ю.П. КЛЬОЦ, О.М. ЛАВРЕНЮК
Хмельницький національний університет

ВИБІР ГРАМАТИКИ ДЛЯ ПРОВЕДЕННЯ ЛЕКСИЧНОГО АНАЛІЗУ ТЕХНІЧНОГО ЗАВДАННЯ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

В статті проведено аналіз граматик на предмет можливості їх використання при лексичному аналізі технічних завдань на розробку програмного забезпечення, написаних українською мовою. Доведено можливість використання для такого аналізу правосторонньої граматики.

Ключові слова: технічне завдання на розробку ПЗ, граматика, алфавіт, правостороння граматика.

O.S. SAVENKO, Y.P. KLOTS, O.M. LAVRENUYK
Khmelnytsky National University

GRAMMAR FOR LEXICAL ANALYSIS OF THE SPECIFICATION FOR SOFTWARE DEVELOPMENT

Abstract - The main objective was to study the possibility of using grammars for lexical analysis of the specification for software development.

The authors demonstrated the possibility of lexical analysis specification for software development using the right-hand grammar and structure of the proposed rules, which are necessary for the lexical analysis.

The proposed grammar allows lexical analysis of the specifications for software development only if properly formed base rules, which requiring of further research.

Keywords: specification for software development, grammar, alphabet, right-hand grammar.

Використання комп'ютерної техніки для наукових досліджень, в побутових пристроях, з промисловою метою вимагає розробки відповідного програмного забезпечення. В свою чергу, створення якісного програмного забезпечення неможливе без формування коректного технічного завдання. Таке технічне завдання повинно не суперечити стандартам на іншим нормативним документам, прийнятим у відповідній галузі використання комп'ютерної техніки [1–3].

Для аналізу технічного завдання на повноту і несуперечливість доцільним є використання лексичних аналізаторів. Використання лексичного аналізатора вимагає використання формальної граматики, що описує мову, якою складене технічне завдання [4].

На сьогодні немає обґрунтованого підходу до вибору граматики, яка забезпечить лексичний аналіз технічного завдання на розробку програмного забезпечення (ПЗ).

Постановка задачі: Для розв'язання задачі створення лексичного аналізатора перевірки технічного завдання на розробку програмного забезпечення на повноту, несуперечливість, відповідність вимогам та стандартам предметної галузі необхідно провести аналіз відомих граматик для проведення лексичного аналізу та вибрати граматику, що дозволить проводити лексичний аналіз технічного завдання на розробку ПЗ, написаного українською мовою.

Вибір граматики для опису мови

Граматика – це математична система, яка визначає мову [5]. Граматикою називається множина $H=(N,\Sigma,P,S)$:

N – множина нетермінальних символів, або нетерміналів;

Σ – $\Sigma \cap N = \varnothing$ – множина термінальних символів, або терміналів;

P – підмножина множини $(N \cup \Sigma)^*N(N \cup \Sigma)^*x(N \cup \Sigma)^*$ елемент (α,β) множини P називається правилом, і записується у вигляді $\alpha \to \beta$;

S – виділений символ з множини N, котрий називається початковим символом.

Граматика визначає мову рекурсивним методом. Рекурсія полягає в задані спеціального виду ланцюжків, іменованих ланцюжками виведення граматики $H=(N,\Sigma,P,S)$, S – виведений ланцюжок, якщо α,β,γ – виведений ланцюжок і $\beta \to \delta$, входить в P то α,δ,γ – також виведений ланцюжок;

Виведений ланцюжок граматики H, який не місить нетермінальних символів, називається термінальним ланцюжком. Мова, створювана граматикою H, це множина термінальних ланцюжків, створюваних граматикою H.

Граматики можна класифікувати за їхніми правилами [5]. Нехай $H=(N,\Sigma,P,S)$ – граматика, тоді граматика H називається:

1) Правосторонньою, якщо кожне правило з множини P має вигляд $A \to xB$, або $A \to x$, де $A,B \in N$, $x \in \Sigma^*$, наприклад:

$H = (\{<цифра>\},\{0, \ldots, 9\},\{<цифра>0|, \ldots, |9\},\{<цифра>\})$, $<цифра>$ слід розуміти, як єдиний нетермінальний символ.

Σ^* множина яка включає в себе всі ланцюжки в алфавіті Σ, включаючи e;

2) Контекстно-вільною (безконтекстною) називається граматика, в якій кожне правило з P має вигляд: $A \to \alpha$, де $A \in N$, $\alpha \in (N \cup \Sigma)$, наприклад :

$H=(\{E,T,F\}, \{a,+,*\},P,E)$, де P складається з правил:

$E \to E+T|T$

$T \to T*F|F$

$T \to (E)|a$

Приклад виводу заданої граматики:

$E \Rightarrow E+T \Rightarrow a+T*F$
$\Rightarrow T+T \Rightarrow a+F*F$
$\Rightarrow F+T \Rightarrow a+a*F$
$\Rightarrow a+T \Rightarrow a+a*a$;

3) Контекстно-залежною називається граматика, де кожне правило з P має вигляд: $\alpha \to \beta$, де $|\alpha| \le |\beta|$, в контекстно-залежній граматиці недопустимими є правила типу $A \to e$, оскільки вона має гарантувати рекурсивність створюваної нею мови, наприклад: нехай граматика H задається правилами:

$S \to aSBC|abc$

$CB \to BC$

$bB \to bb$

$bC \to bc$

$cC \to cc$

можливий вивід граматики:

$S \Rightarrow aSBC$
$\Rightarrow aabCBC$
$\Rightarrow aabBCC$
$\Rightarrow aabbCC$
$\Rightarrow aabbcC$
$\Rightarrow aabbcc$

дана граматика породжує мову $\{a^n, b^n, c^n \mid n \ge 1\}$;

4) Граматика, яка не відповідає жодному з вище наведених правил, називається граматикою загального вигляду, наприклад: нехай граматика H задається правилами:

$S \to CD$ $\quad Ab \to aB$
$C \to aCA$ $\quad Ba \to bA$
$C \to bCB$ $\quad Bb \to bB$
$AD \to aD$ $\quad C \to e$
$BD \to bD$ $\quad D \to e$
$Aa \to aA$,

приклад виводу граматики:

$S \Rightarrow CD$
$\Rightarrow aCAD$
$\Rightarrow abCBAD$
$\Rightarrow abBAD$
$\Rightarrow abBaD$
$\Rightarrow ababD$
$\Rightarrow abab$.

В якості мови для складання технічного завдання на розробку ПЗ використовується українська мова. Мова складається з наступних частин: самостійні частини мови, службові частини мови, вигуки і звуконаслідування [6–8].

Самостійні частини мови (їх шість: іменник, прикметник, числівник, займенник, дієслово і прислівник) називають предмети, їх ознаки, дії та кількість. Самостійні частини мови є членами речення, і мають, як лексичне, так і граматичне значення.

Таблиця 1

Самостійні частини мови

Частина мови	**Лексичне значення частини мови**
іменник	означає предмети і явища
прикметник	виражає ознаку предметів
числівник	позначає число, кількість предметів, порядок їх при лічбі
займенник	вказує на особу, предмет
дієслово	виражає дію або стан предметів, включає дієприкметники
прислівник	позначає ознаку дії, ознаку іншої ознаки чи предмета

Таблиця 2

Службові частини мови

Частина мови	Лексичне значення частини мови
прийменник	служить засобом вираження відношень іменника до інших слів у реченні
сполучник	зв'язує між собою однорідні члени речення і частини складного речення
частка	виражає модальні відтінки у реченні

Службові частини мови (їх три: прийменник, сполучник, частка) предметного лексичного значення не мають і служать лише для зв'язку слів у реченні (прийменник, сполучник) або для надання окремим словам і реченням додаткових смислових чи емоційно-експресивних відтінків, а також для творення морфологічних форм і нових слів (частка).

Вигуки і звуконаслідування виражають лише волевиявлення, емоції, етикет, імітують звукові сигнали птахів, тварин, явищ природи, не використовуються в тексті ПЗ [6].

Оскільки українська мова при використанні у технічній документації на розробку ПЗ є більш структурована (формалізована), то опишемо частини мови, як множини [5–8]:

A – множина іменників визначає, яка скінчена кількість іменників a_i входить до множини A, $A=\{a_1...a_n\}$, $n=\overline{1,n}$, n – загальна можлива кількість іменників;

B – множина прикметників визначає, яка скінчена кількість прикметників b_i входить до множини B, $B=\{b_1...b_n\}$, $n=\overline{1,n}$, n – загальна можлива кількість прикметників;

C – множина пунктуаційних знаків визначає, яка скінчена кількість пунктуаційних знаків c_i входить до множини C, $C=\{c_1...c_n\}$, $n=\overline{1,n}$, n – загальна можлива кількість пунктуаційних знаків;

I – множина дієслів визначає, яка скінчена кількість дієслів i входить до множини I, $I=\{i_1...i_n\}$, $n=\overline{1,n}$, n – загальна можлива кількість дієслів;

J – множина прислівників визначає, яка скінчена кількість прислівників j входить до множини J, $J=\{j_1...j_n\}$, $n=\overline{1,n}$, n – загальна можлива кількість прислівників;

O_i – множина прийменників визначає, яка скінчена кількість прийменників o входить до множини O_i, $O=\{o_1...o_n\}$, $n=\overline{1,n}$, n – загальна можлива кількість прийменників;

U_i – множина сполучників визначає, яка скінчена кількість сполучників u входить до множини U_i, $U=\{u_1...u_n\}$, $n=\overline{1,n}$, n – загальна можлива кількість сполучників;

V_i – множина займенників визначає, яка скінчена кількість займенників v входить до множини V_i, $V=\{v_1...v_n\}$, $n=\overline{1,n}$, n – загальна можлива кількість займенників;

Y_i – множина інших частин мови визначає, яка скінчена кількість з інших частин мови y входить до множини Y_i, $Y=\{y_1...y_n\}$, $n=\overline{1,n}$, n – загальна можлива кількість інших частин мови, під іншими частинами мови слід розуміти часку та числівник;

E – множина символів визначає, яка скінчена кількість символів e_i входить до множини E, $E=\{e_1...e_n\}$, $n=\overline{1,n}$, n – загальна можлива кількість символів;

D – множина слів визначає, яка скінчена кількість слів d_i входить до множини D, $D=\{d_1...d_n\}$, $n=\overline{1,n}$, n – загальна можлива кількість слів;

W – множина речень визначає, яка скінчена кількість речень w_i входить до множини W, $W=\{w_1...w_n\}$, $n=\overline{1,n}$, n – загальна можлива кількість речень;

F – множина лексем визначає, яка скінчена кількість лексем f_i входить до множини W, $F=\{f_1...f_n\}$, $n=\overline{1,n}$, n – загальна можлива кількість лексем.

Нехай маємо граматику H, яка описує текст T, де $H=(N,\Sigma,P,S)$, $T=\{W\}$, $W=\{D\}$, $D=\{a,b,c,j,i,o,u,v,y\}$ – класичний художній текст, який складається з усіх частин мови, де $a\in A\subset D, b\in B\subset D, c\in C\subset D, i\in I\subset D, j\in J\subset D, o\in O\subset D, u\in U\subset D, v\in V\subset D, y\in Y\subset D$ [5–8].

Текст з технічного завдання на розробку ПЗ буде складатися з наступних частин мови $T=\{W\}$, $W=\{D'\}$, де W – множина речень складається з множини слів $D'=\{a,b,c,i,j,o,u,v\}$, де $D'\subset D$, D' – підмножина множини слів D', $a\in A\subset D'$, $b\in B\subset D'$, $c\in C\subset D'$, $j\in J\subset D'$, $i\in I\subset D'$, $o\in O\subset D'$, $u\in U\subset D'$, $v\in V\subset D'$, та описується граматикою $H=(N,\Sigma,P,S)$ [5–8].

Множина нетерміналів N, має вигляд $N=\{V,U,C,Y,O,J\}$, де V – множина займенників, U – множина сполучників, C – множина пунктуаційних знаків, Y – множина інших частин мови, O –

множина прийменників, J – множина прислівників;

Множина терміналів Σ, має вигляд $\Sigma=\{F\}$, $F=\{W\}$, $F=\{D\}$, $D=\{A,B,I\}$, де F – множина терміналів (лексем), D' – множина слів утворюваних мовою H, W – множина речень, A – множина іменників, B – множина прикметників, I – множина дієслів.

Множина правил P буде мати вигляд: $D \to xZ$, $D, Z \in N, x \in \Sigma$, де D' – множина слів. Підставивши отримані дані в $H=(N,\Sigma,P,S)$ отримаємо граматику:

$$H=(\{V,U,C,Y,J,O\}, \{A,B,I\}, \{P\}, S) \quad (1)$$

Граматика H утворює мову L(H), якою буде написано текст технічного завдання на розробку ПЗ. Для мови L(H) – характерні наступні властивості:

$H=(\{S\}, \Sigma, \varnothing, S)$ – правостороння граматика, для котрої $L(H)=\varnothing$.

$H=(\{S\}, \Sigma, \{S \to e\}, S)$ – правостороння граматика, для котрої $L(H)=\{e\}$.

$H_a=(\{S\}, \Sigma, \{S \to a\}, S)$ – правостороння граматика, для котрої $L(H_a)=\{a\}$.

Оскільки описані вище властивості граматики співпадають з властивостям правосторонньої граматики доведемо, що вона є правостороння [5].

Доведення:

Якщо L та L_2 правосторонні мови, то мови (I) $L \cup L_2$, (II) LL_2, (III) L^* також правосторонні.

Так як мови L та L_2 правосторонні, то можливо припустити, що існують правосторонні граматики $H=(N, \Sigma, P, S)$ та $H_2=(N_2, \Sigma, P_2, S_2)$, для котрих L(H)=L та $L(H)=L_2$. Припустимою, алфавіти N та N_2 не перетинаються. Так як нетермінали граматики можна перейменувати як завгодно, це припущення не призведе до втрати суті.

(I) Нехай H_3 – правостороння граматика

$$H_3=(N \cup N_2 \cup \{S_3\}, \Sigma, P \cup P_2 \cup \{S_3 \to S|S_2\}, S_3) \quad (2)$$

де S_3 – новий нетермінальний символ, який не належить $S_3 \notin N$ та $S_3 \notin N_2$. Звідси слідує, що $L(H_3)=L(H) \cup L(H_2)$, так як для кожного виводу $S_3 \Rightarrow a_3 w$ існує виведення $S \Rightarrow aw$ або $S_2 \Rightarrow a_2 w$ та навпаки. Отже H_3 – правостороння граматика, то $L(H_3)$ – правостороння мова.

(II) Нехай H_4 – правостороння граматика $(N \cup N_2, \Sigma, P_4, S_1)$, в якій P_4 визначається як:

1) якщо $A \to xB$ належить P, $A \to xB$ належить P_4;

2) якщо $A \to x$ належить P, $A \to xS_2$ належить P_4;

3) всі правила з P_2 належить P_4.

Отже, якщо $S \Rightarrow aw$, то $S_1 \Rightarrow a_4 w S_2$, а якщо $S_2 \Rightarrow a_2 x$, то $S_2 \Rightarrow a_4 x$. Таким чином $L(H)L(H_2) \subseteq L(H_4)$. Припустимо $S \Rightarrow a_4 w$. Так як в P_4 немає правил виду $A \to x$, котрі потрапили б туди з P_1, то цей висновок можна записати у вигляді $S \Rightarrow a_4 x S_2 \Rightarrow a_4 xy$, де $w = xy$ і всі правила, використовуванні у виведенні $S \Rightarrow a_4 x S_2$ потрапили в P_4 за допомогою дій (1), (2). Отже, повинні бути виведення $S \Rightarrow ax$ та $S_2 \Rightarrow a_2 y$. Звідси $L(H_4) \subseteq L(H)L(H_2)$.

(III) Нехай граматика $H_5=(N \cup \{S_5\}, \Sigma, P_5, S_5)$ така, що S_5 не належить N_1, а P_5 задається наступними правилами:

1) якщо $A \to xB$ належить P, $A \to xB$ належить P_5;

2) якщо $A \to x$ належить P, $A \to xS_5$ належить P_5;

3) $S_5 \to S \mid e$ належить P_5;

доведення того, що $S_5 \Rightarrow a_5 x_1 S_5 \Rightarrow a_5 x_1 x_2 S_5 \Rightarrow a_5 \ldots \Rightarrow a_5 x_1 x_2 \ldots x_{n-1} S_5 \Rightarrow \Rightarrow a_5 x_1 x_2 \ldots x_{n-1} x_n$ тоді і тільки тоді, коли $S = ax_1$, $S = a_1 x_2, \ldots$, $S = ax_n$, $L(H_5)=(L(H))^*$.

$L(H_5)$, $L(H_3)$, $L(H_4)$ – мови, які породжуються правосторонніми граматиками в складі яких є граматика H. Отже граматика H є правосторонньою, оскільки виконуються наступні умови: $L(H_4) \subseteq L(H)L(H_2)$, $L(H_5)=(L(H))^*$, $L(H_3)=L(H) \cup L(H_2)$.

Це дозволяє:

1) будувати правила, з слів які входять до технічного завдання на розробку ПЗ;

2) використовувати один і той самий алфавіт для складання тексту технічного завдання на розробку ПЗ T і правил граматики P;

2) чітко розмежувати термінальні та нетермінальні символи. Під символом слід розуміти слово з тексту.

Приклад

Правила:

Речення $\to$ Речення + Слово;

Речення → Слово + Речення;
Речення → Речення + Іменник + Речення;
…
Слово → Слово + Слово;
Слово → Іменник;
Слово → Дієслово;
Слово → Прикметник;
Слово → Частка;
…
Іменник → {модуль, користувач};
Дієслово → {буде, зареєстрований};
Прийменник → {доступний};
Прийменник → {для};
Частка → {тільки, лише};
Речення → Іменник + дієслово + прикметник + частка + дієслово + іменник;
Речення → Модуль + дієслово + доступний + частка + зареєстрований + користувач;

На основі запропонованого набору правил проведемо аналіз трьох речень. Перші два речення близькі за змістом та не значно відрізняються за складом слів. Третє речення відрізняється за змістом, але складається із тих самих іменників та дієслів.

Речення 1: «Модуль буде доступний лише зареєстрованим користувачам» – відповідає граматиці.

Речення 2: «Модуль буде доступний тільки для зареєстрованих користувачів» – відповідає граматиці.

Речення 3: «Модуль буде доступний користувачам» – не відповідає граматиці, оскільки не відповідає жодному із правил.

Лексичний аналіз з використанням правосторонніх граматик передбачає використання множини правил. Він може бути проведений в повному обсязі лише за умови коректної побудови правил.

Висновок

Аналіз відомих граматик показав, що для лексичного аналізу технічного завдання на розробку програмного забезпечення доцільно використовувати правосторонню граматику. Така граматика дозволяє чітко розмежувати термінальні та нетермінальні символи та будувати правила, використовуючи той самий алфавіт, що і технічне завдання. Для аналізу технічного завдання на розробку ПЗ із використанням граматики необхідно мати відповідний набір правил.

Література

1. IEEE Std 830 – 1998 IEEE Recommended Practice for Software Requirements Specifications (Практичні рекомендації по специфікації програмного забезпечення).

2. IEEE Std. 610.12 – 1990 IEEE Standard Glossary of Software Engineering Terminology.

3. Petter L. Quantification and Traceability of Requirements / Petter L. , H. Eide – TDT4735 Software Engineering Depth Study – Fall – 2005.

4. Савенко О.С. Методи лексичного аналізу технічного завдання на розробку програмного забезпечення / О.С. Савенко, Ю.П. Кльоц, В.С. Шевцов // Вісник Хмельницького національного університету. – 2011. – № 5. – С. 167–172.

5. Ю.П. Кльоц. Лексичний аналізатор технічних завдань на розробку критичного програмного забезпечення / Ю.П. Кльоц, В.С. Шевцов // Труди XII МНПК "Сучасні інформаційні та електронні технології" – 2011. – С. 109.

6. А. Ахо Теория синтаксическоо анализа, перевода, компиляции Синтаксический анализ / А. Ахо, Дж. Ульман. – М., 1978. – Т. 1 – 613 с.

7. Daniel D. Sleator Parsing English with a Link Grammar – CS143 – Handout03 – 2008.

8. Sleator, D. D., D. Temperlay, "Parsing English with a Link Grammar", Technical report CMU-91-196, Carnegie Mellon University, School of Computer Science, October 1991.

References

1. IEEE Std 830 – 1998 IEEE Recommended Practice for Software Requirements Specifications

2. IEEE Std. 610.12 – 1990 IEEE Standard Glossary of Software Engineering Terminology.

3. Petter L. Quantification and Traceability of Requirements / Petter L. , H. Eide – TDT4735 Software Engineering Depth Study – Fall – 2005.

4. Savenko O.S. Metody leksychnoho analizu tekhnichnoho zavdannia na rozrobku prohramnoho zabezpechennia /O.S. Savenko, Yu.P. Klots, V.S. Shevtsov // Visnyk Khmelnytskoho natsionalnoho universytetu. 2011, #5, s. 167-172. [in Ukrainian]

5. Y.P. Klots, V.S. Shevtsov Leksychnyi analizator tekhnichnykh zavdan na rozrobku krytychnoho prohramnoho zabezpechennia / Trudy XII MNPK "Suchasni informatsiini ta elektronni tekhnolohii" – 2011 - s.109. [in Ukrainian]

6. Akho Teoryia syntaksycheskoo analyza, perevoda, kompyliatsyy Tom 1 Syntaksycheskyi analyz / A. Akho, Dzh. Ulman – Moskva – 1978 – 613s. [in Russian]

7. Daniel D. Sleator Parsing English with a Link Grammar – CS143 – Handout03 – 2008.

8. Sleator, D. D., D. Temperlay, "Parsing English with a Link Grammar", Technical report CMU-91-196, Carnegie Mellon University, School of Computer Science, October 1991.

Рецензія/Peer review : 10.4.2013 р. Надрукована/Printed :18.6.2013 р.
Рецензент: д.т.н., проф. зав. каф. СПр, ХНУ Поморова О.В.