

**Комлева Н.О.**

Одеський національний політехнічний університет

**Зіноватна С.Л.**

Одеський національний політехнічний університет

**Марулін С.Ю.**

Одеський національний політехнічний університет

**Онщенко Т.В.**

Одеський національний політехнічний університет

## ПОБУДОВА МОДЕЛІ АТРИБУТІВ ЯКОСТІ ФОРМАЛЬНИХ ГРАМАТИК ТА МОВ ПРОГРАМУВАННЯ

*Роботи в області теорії трансляції можуть носити як комерційний, так і науково-дослідний характер і потребують вирішення ряду проблем. Одна проблема полягає в необхідності розробки транслятора, що дозволяє переводити програму, написану мовою високого рівня, на машинну мову. Це вимагає чіткого розуміння принципів роботи кожного етапу транслятора: лексичний аналіз, синтаксичний аналіз, семантичний аналіз, оптимізація, генерація коду. Інша проблема, пов'язана з мовою програмування, це проблема вирішення нетривіального завдання синтаксису мови програмування. При цьому традиційно для опису синтаксису мови використовують контекстно-вільну граматику, у вигляді якої можна формалізувати більшу частину правил, призначених для опису синтаксичної структури.*

*Метою роботи є підвищення якості процесу розробки формальних граматик та мов шляхом формалізації множини атрибутів якості та їх використання при розробці власних трансляторів або їх окремих блоків. Для досягнення поставленої мети було вирішено наступні завдання: проведено огляд етапів процесу трансляції та визначено вимоги до них; формалізовано склад атрибутів якості для лексичного та синтаксичного етапів розробки трансляторів; виконано аналіз атрибутів якості формальних граматик та мов; побудовано модель атрибутів якості формальних граматик та мов з використанням логіки предикатів з визначенням вимог до словника граматики, лексем, правил та систем правил.*

*Розроблено методичку з побудови граматик та відповідну програмну систему, що дозволяє користувачеві конструювати формальні граматики та аналізувати результати їх побудови. Програмна система оцінює атрибути якості відповідно до формалізованої моделі якості та видає зворотний зв'язок користувачеві. Застосування програми дозволило виявляти та деталізувати приблизно 78% усіх помилок при створенні граматик.*

**Ключові слова:** атрибути якості, побудова моделі, формальна граMATика, транслятор, мова програмування.

**Постановка проблеми.** Визначаючи мову програмування, потрібно, як мінімум, формалізувати наступні поняття: множину символів, які можна використовувати для запису правильних програм; множину правильних програм; «сенса» кожної правильної програми. Визначення множини допустимих символів потрібно для роботи першої частини транслятора – лексичного аналізатора. При завданні мови програмування визначається клас допустимих програм за допомогою граматичних правил. При цьому більшу частину правил, призначених для опису синтаксичної структури, можна формалізувати

у вигляді контекстно-вільної граматики. Більш того, контекстно-вільна граMATика забезпечує опис, достатній для того, щоб його можна було використовувати як частину визначення самого компілятора.

Для можливості формального опису нової вхідної мови необхідна розробка граматичних правил граматики, яка повинна виконуватись з забезпеченням заданих атрибутів якості формальних граматик та мов. Для атрибутів якості необхідно вказати певні значення, дотримання яких забезпечить коректність граматики, а також описати можливі помилки і способи їх усунення.

**Аналіз останніх досліджень і публікацій.** На даний час доступна велика кількість сучасних інструментів для розробки компіляторів, які заслуговують детального вивчення і широкого використання. Серед цих інструментів є такі, які автоматизують розробку лексичних і синтаксичних аналізаторів з використанням сучасних методів, наприклад ANTLR, CoCo/R, SableCC, bison [1, с. 51; 2, с. 96]. Найбільшу популярність має генератор низхідних аналізаторів для формальних мов ANTLR, що перетворює контекстно-вільну граматику у вигляді розширеної форми Бекуса-Наура в програму на C++, Java, C#, JavaScript, Go, Swift, Python [3, с. 1]. Також широко відомий bison – генератор LR-парсерів для мови C, аналогії якого існують для дуже багатьох мов [4, с. 1].

Опис мов програмування у цих інструментах спирається на теорію формальних мов, що є фундаментом для організації синтаксичного аналізу [5, с. 514; 6, с. 261]. Відповідно до теорії формальних мов, мова – це множина ланцюжків кінцевої довжини на заданому алфавіті. Механізм породження дозволяє описати мову за допомогою граматики – системи правил, та ланцюжки мови будуються відповідно до цих правил [7, с. 190; 8, с. 408]. Гідність визначення мови за допомогою граматики в тому, що операції в ході синтаксичного аналізу можна формалізувати і спростити їх виконання.

Для того, щоб стало можливим визначати мову за допомогою граматики, правила цієї граматики повинні відповідати певним вимогам. Недотримання цих вимог призводить до проблем при роботі основних етапів транслятора [9, с. 73]. Коректно побудована синтаксична структура речення допомагає зрозуміти взаємини між різними його частинами. Для побудови точних і недвозначних описів застосовують метод абстракцій, який передбачає виділення найбільш істотних властивостей даного об'єкту і опускання властивостей, менш значущих для певного випадку [10, с. 17; 11, с. 105]. Наприклад, при побудові моделі вхідних мов можна розглядати вихідний текст як послідовність символів, побудовану за певними правилами.

Як показав аналіз, недоліком багатьох інструментів розробки є слабка інформаційна підтримка користувача у ході розробки формальних граматики та мов за певним описом. Для формування системи інформаційної підтримки потрібно розробити методику з побудови граматики та відповідне програмне забезпечення.

**Мета роботи.** Метою роботи є підвищення якості процесу розробки формальних граматики

та мов шляхом формалізації множини атрибутів якості та їх використання при розробці власних трансляторів або їх окремих блоків.

**Виклад основного матеріалу дослідження.** Систематизація атрибутів якості формальних граматики та мов дозволила згрупувати їх за наступними критеріями якості:

- стосовно символів граматики;
- стосовно лексем граматики;
- стосовно початкового символу граматики;
- відповідно до правил граматики [12, с. 110].

Формалізація вимог за цими критеріями дозволила створити відповідну модель атрибутів якості [13, с. 106; 14, с. 147]. Модель атрибутів якості M формальних граматики та мов виглядає наступним чином:

$$M = (D, Lex, R, SR),$$

де D, Lex, R та SB – узагальнені атрибути якості для словника граматики, лексем, правил та систем правил граматики відповідно. Автори розглядають кожний із них більш детально.

*Коректність словника граматики D* визначається трійкою вимог:

$$D = \{D_T, D_N, D_S\},$$

де  $D_T$  – вимоги щодо термінальних символів:

- 1)  $V_T = \{k_1, k_2, k_n\} \neq \emptyset$ ;
- 2) потужність множини  $n = |V_T|$  визначається предметною областю;
- 3)  $\forall (k_i \in V_T, k_j \in V_T) \text{ Name}(k_i) \neq \text{Name}(k_j)$ .

$D_N$  – вимоги щодо нетермінальних символів:

- 1)  $V_N = \{p_1, p_2, p_m\} \neq \emptyset$ ;
- 2) потужність множини  $m = |V_N| \geq 1$ ;
- 3)  $\forall (p_i \in V_N, p_j \in V_N) \text{ Name}(p_i) \neq \text{Name}(p_j)$ ;
- 4)  $V_T \cap V_N = \text{***}$ .

$D_S$  – вимоги щодо початкового символу S граматики:  $S \in V_N$ .

*Коректність лексем граматики Lex* визначається четвіркою вимог:

- 1)  $Lex = \{Lex_1, Lex_2, Lex_r\} \neq \emptyset$ ;
- 2) якщо довжина лексеми обмежена, то кількість їх варіантів також обмежена, тобто  $r = \text{const}$ , інакше  $r \rightarrow \infty$ ;
- 3)  $\forall (Lex_i \in Lex, Lex_j \in Lex) \text{ Name}(Lex_i) \neq \text{Name}(Lex_j)$ ;

4) повинна бути можливість однозначного виділення окремих лексем з речення.

*Коректність правил граматики R* визначається трійкою вимог:

- 1) виведення речень є направленим: запис виду  $\alpha \rightarrow \beta$  означає, що ланцюжок  $\alpha$  породжує ланцюжок  $\beta$ ;
- 2) для випадку контекстно-вільних граматики:  $\forall \alpha \in V_N (\alpha) \neq \emptyset, V_T (\alpha) = \emptyset$ ,

$\forall \beta V_N(\beta) \cup V_T(\beta) \neq \emptyset$ ;

3) кількість нескінченних рекурсій у правилах  $Rec = 0$ .

Коректність системи правил граматики SR визначається двійкою вимог:

1) для мови  $L(G) = \{f_1, f_2, f_h\}$ ,

де  $f_i, i=1, h$  – речення мови,  $h=const$  для кінцевих мов,  $h \rightarrow \infty$  для нескінченних мов (цей випадок є більш перспективним);

2) якщо  $W$  – система правил граматики з  $t \geq 1$

$$W = \begin{cases} W_0 \rightarrow W_1 \\ \dots \\ W_{t-1} \rightarrow W_t \end{cases},$$

то  $W_0 = S$  – початковий символ граматики, при цьому  $W_t \in L(G) = \{f_1, f_2, f_h\}; \forall (W_i, W_j), i \neq j, W_i \neq W_j$ .

Неухильне дотримання всіх перерахованих критеріїв є обов’язковою умовою для побудови граматики, призначеної для трансляції програми або перекладу речень природної мови. Відхилення від вимог знижує якість початкових даних і, як наслідок цього, якість побудованої граматики і відповідної їй мови [15, с. 91].

Відповідно до розробленої моделі створено методику розробки формальних граматик, що є досить об’ємною. В якості її фрагменту автори наводять базові принципи відображення структури ланцюжків у правила граматики:

– ланцюжку, що складається із заданих символів  $abc$ , відповідає правило  $V \rightarrow abc$ ;

– ланцюжку, що починається з заданого символу  $a$ , відповідає правило  $V \rightarrow a B$ ;

– ланцюжку, що закінчується заданим символом  $a$ , відповідає правило  $V \rightarrow B a$ ;

– ланцюжку, що починається і закінчується заданими символами  $a$  та  $b$  відповідає правило  $V \rightarrow a B b$ ;

– ланцюжку, що містить в середині символ  $a$ , відповідає правило  $V \rightarrow A a B$ ;

– ланцюжку заданої довжини  $1 = 2$  відповідають правила:  $A \rightarrow a B$  та  $B \rightarrow a$ ;

– ланцюжку, що складається з повторюваних символів  $a$ , відповідають правила  $A \rightarrow a A$  та  $A \rightarrow a$ ;

– ланцюжку, що складається з символів  $a$  і  $b$ , які чергуються, відповідають правила:  $A \rightarrow a B$  та  $B \rightarrow b A$ .

На базі побудованої моделі атрибутів якості та методики створено відповідну програмну систему, що дозволяє користувачеві конструювати формальні граматики (рис. 1).

Дуже коротко автори описують принципи користування системою. Кожне правило починається з нового рядка, завершувати правило спеціальним символом не потрібно. Якщо нажата кнопка «ABC – перевірка рекурсивності правил» та кнопка «Ошибки», то при наявності помилок вони будуть відображені у нижній частині

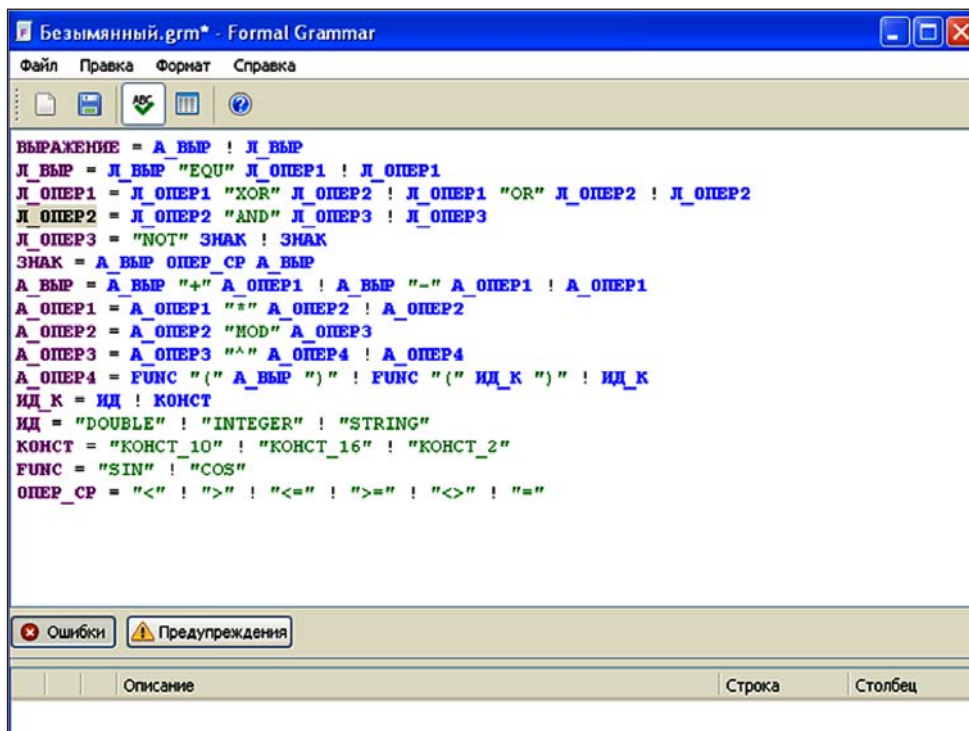


Рис. 1. Вікно з правилами формальної граматики

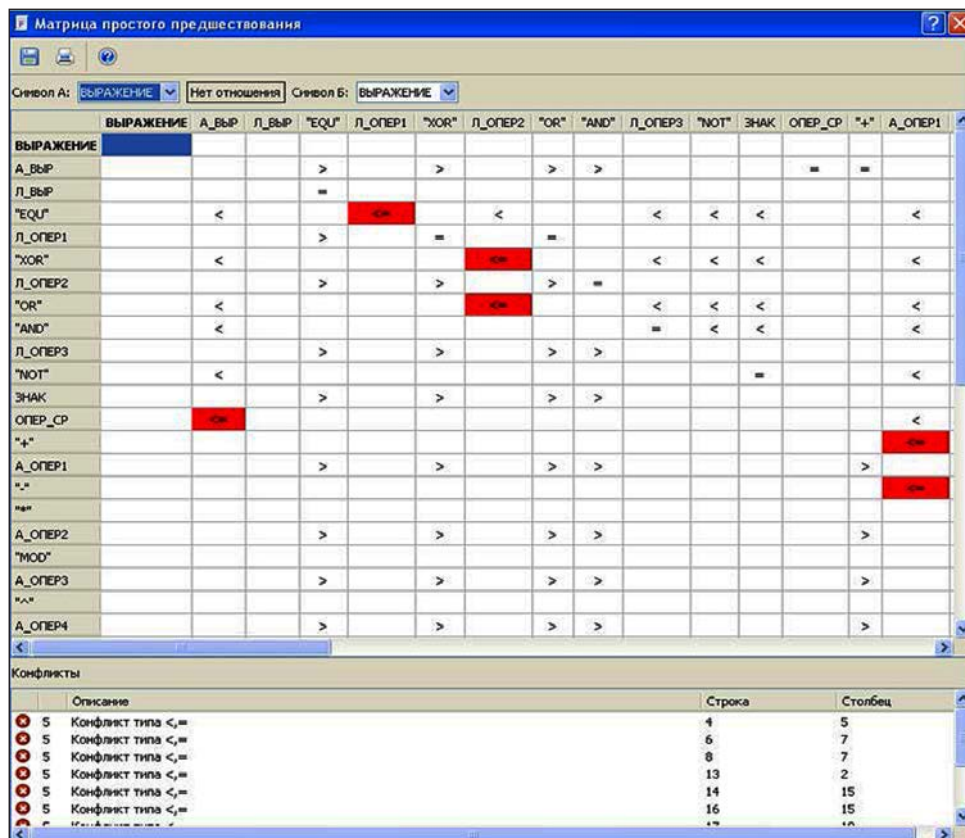


Рис. 2. Вікно з матрицею передування

вікна. Кнопка «Предупреждения» попереджає про дублювання термінальних символів у правих частинах правил. Вибір відповідної команди або натискання піктограми дозволяє відобразити побудовану матрицю передування. На рисунку 2 показана матриця простого передування для вказаної граматики.

Вікно матриці простого передування дозволяє зручно переглядати відношення між парами символів. Якщо граMATика має велику кількість термінальних і нетермінальних символів, то відношення між ними краще переглядати за допомогою двох списків «Символ А» і «Символ Б». Відповідна комірка матриці буде підсвічена, а саме відношення буде показано між списками, що випадають. У вікні також можна переглянути список конфліктів, що є неприпустимими. Вибір елемента зі списку конфліктів підсвітить відповідну комірку в матриці. Матрицю можна зберегти у форматі HTML та роздрукувати.

**Висновки.** У роботі проведено огляд етапів процесу трансляції та визначено вимоги до них; формалізовано склад атрибутів якості для лексичного та синтаксичного етапів розробки трансляторів; виконано аналіз атрибутів якості формальних граматики та мов; побудовано модель атрибутів якості формальних граматики та мов з використанням логіки предикатів з визначенням вимог до словника граматики, лексем, правил та систем правил. Розроблено методику з побудови граматики, що містить рекомендації користувачеві та механізми перевірки коректності побудованої граматики.

Наведений підхід реалізовано при проектуванні програмної системи для аналізу та оцінювання атрибутів якості формальних граматики та мов. Застосування розробленої моделі якості на базі логіки предикатів дозволило виявляти та деталізувати приблизно 78% всіх помилок при створенні граматики.

#### Список літератури:

1. Safonov V.O. Trustworthy Compilers. *Wiley Interscience. John Wiley & Sons.* 2010. 320 p.
2. Левушкин А.В., Турчанинов М.К., Жиганов А.А., Ермолаева В.В. Основные современные языки программирования. *Молодой ученый.* 2018. № 25. С. 96–98.

3. ANTLR [Електронний ресурс] Режим доступу: URL: <https://www.antlr.org/>, режим з екрану. Активне посилання – 12.11.2019.
4. Bison – GNU Project [Електронний ресурс] Режим доступу: URL: <http://www.gnu.org/software/bison/>, режим з екрану. Активне посилання – 12.11.2019.
5. Keith D. Cooper, Linda Torczon. Engineering a compiler. *Rice University, Houston, Texas*. 2012. 787 p.
6. Ахо А., Сети Р., Лам М., Ульман Д. Компиляторы: принципы, технологии и инструментарий. ООО «Вильямс». 2008. 1184 с.
7. Соколов А.П. Системы программирования: теория, методы, алгоритмы. *Финансы и статистика*. 2014. 330 с.
8. Хопкрофт Д., Мотвани Р., Ульман Д. Введение в теорию автоматов, языков и вычислений. ООО «Вильямс». 2012. 528 с.
9. Сельнев П.Б. Разработка интерпретаторов и компиляторов. Использование Bison, Ясс, Zubr. ООО Преміум-СБ. 2011. 314 с.
10. Meduna I. Elements of Compiler Design. *CRC Press*. 2017. 119 p.
11. Lewis P. M. Compiler Design Theory. *Addison-Wesley*. 2006. 423 p.
12. Komleva N.O., Popov S.S. Quality attributes of formal grammars and languages in translator engineering. *XII Міжнародна науково-практична конференція «Інформаційні технології і автоматизація»*. 2019. С. 110–112.
13. Komleva N.O. Some aspects of analyzing and diagnosing complex systems. *International Multidisciplinary Conference «Science and Technology of the Present Time: Priority Development Directions of Ukraine and Poland»*. Wolomin: Izdevnieciba «Baltija Publishing». 2018. V. 3. P. 106–109.
14. Komleva N.O. Comparative characteristics of the Metrics Project Design Tools. *Abstracts of the XII International Conference «Modern information and communication technologies on a transport, in industry and education»*. Dnipro. 2018. P. 147.
15. Крисиллов В.А., Комлевая Н.О. Анализ и оценка компетентности источников информации в задачах интеллектуальной обработки данных. *Международная научно-практическая конференция «Электротехнические и компьютерные системы: теория и практика» ELTECS-2019. Problemele energeticii regionale*. Вып. 1–1 (40). 2019. С. 91–104.

#### **Komleva N.O., Zinovatna S.L., Marulin S.U., Onishchenko T.V. CONSTRUCTION OF MODEL OF QUALITY ATTRIBUTES FOR FORMAL GRAMMARS AND PROGRAMMING LANGUAGES**

*Works in the field of translation theory can be both commercial and research in nature, and require solving a number of problems. One problem is the need to develop a translator that allows you to translate a program written in a high-level language into a machine language. This requires a clear understanding of the principles of operation of each stage of the translator: lexical analysis, parsing, semantic analysis, optimization, code generation. Another problem related to programming language is the problem of solving a non-trivial task of programming language syntax.*

*Traditionally, context-free grammar is used to describe language syntax, in which most of the rules designed to describe syntactic structure can be formalized. The aim of the work is to improve the quality of the process of developing formal grammars and languages by formalizing the set of quality attributes and using them in the development of own translators or their separate blocks. To achieve this aim, the following tasks were solved: a review of the stages of the translation process was carried out and requirements for them were determined; the composition of quality attributes for the lexical and syntactic stages of translator development is formalized; the quality attributes of formal grammars and languages were analyzed; a model of quality attributes of formal grammars and languages which included requirements for grammar, lexemes, rules and rule systems was constructed using predicate logic.*

*The methodology for grammar construction and the appropriate software system, that allow the user to construct formal grammars and analyze the results of their construction, have been developed. The software system evaluates the quality attributes according to the formalized quality model and gives feedback to the user. Using of the program allowed identifying and detailing about 78% of all errors during creating grammars.*

**Key words:** quality attributes, model building, formal grammar, translator, programming language.