

## МОДЕЛИРОВАНИЕ ДИНАМИЧЕСКОЙ СИСТЕМЫ СРЕДСТВАМИ ФУНКЦИОНАЛЬНЫХ ГРАММАТИК

© Николаева Дарима Доржиевна

научный сотрудник,

Бурятский государственный университет имени Доржи Банзарова

Россия, 670000, г. Улан-Удэ, ул. Смолина, 24а

E-mail: darisha89@yandex.ru

© Ширапов Дашадондок Шагдарович

доктор физико-математических наук, профессор,

Бурятский государственный университет имени Доржи Банзарова

Россия, 670000, г. Улан-Удэ, ул. Смолина, 24а

E-mail: shir48@mail.ru

© Анахин Владимир Дмитриевич

доктор технических наук, профессор,

Бурятский государственный университет имени Доржи Банзарова

Россия, 670000, г. Улан-Удэ, ул. Смолина, 24а

E-mail: anakhin@mail.ru

В работе рассматривается вопрос о моделировании класса компьютерных программ, которые порождают вычислительные процессы и адекватно описывают работу динамической системы. Для решения данной задачи используются средства функциональных грамматик. Основной базисной операцией функциональных грамматик является операция отождествления. В работе рассмотрен пример использования операции отождествления на фрагменте компьютерной программы, которая моделирует динамическую систему. Конечным результатом является получение суперпозиции базисных функций с привлечением операции отождествления. Полученная суперпозиция является математической моделью динамической системы, представленной в виде компьютерной программы. Построено дерево синтаксического разбора компьютерной программы, в узлах которой находятся базисные функции (нетерминалы). Если в узлах требуется представление в виде **функ** (объект есть функция), то данный узел интерпретируется как суперпозиция базисных функций. Если же в узлах требуется представление в виде **текст**, то объектом является фрагмент текста программы, соответствующий нетерминалу (нетерминал в данном узле представляет собой фрагмент текста программы). Также в узлах может потребоваться представление в виде **знач** (объект есть результат выполнения вышесказанной суперпозиции функций, соответствующей нетерминалу).

*Ключевые слова:* динамические системы; моделирование; математическая модель языка; универсальная алгебра; контекстно-свободная грамматика; рекурсия; интерпретатор; семантика.

**Для цитирования:**

*Николаева Д. Д., Ширапов Д. Ш., Анахин В. Д.* Моделирование динамической системы средствами функциональных грамматик // Вестник Бурятского государственного университета. Математика, информатика. 2019. № 3. С. 69–76.

**Введение**

Конструктивное построение отображений между множествами слов, которые порождаются контекстно-свободными грамматиками, имеет большое практическое значение, поскольку контекстно-свободные грамматики являются гибкими инструментами для описания иерархических классификаций. Для разработки компьютерных программ, в которых присутствуют сложные объекты, необходима математическая модель вычислительных процессов, порождаемых программой.

Статья посвящена построению отображений между множествами, которые порождаются контекстно-свободными грамматиками. Такой подход дает возможность математически описывать компьютерную программу, моделирующую динамическую систему.

**1 Постановка задачи**

В работе рассматривается задача об уточнении понятия класса компьютерных программ, моделирующих динамическую систему. Решение этой задачи позволит исследовать структуру класса алгоритмов, моделирующих динамическую систему, и в той или иной степени сформулировать утверждения о поведении динамической системы, исследуя вычислительный процесс, изоморфный к функционированию динамической системы. Для такой постановки вопроса требуется построить формальную математическую модель класса компьютерных программ, моделирующих динамическую систему. Для построения математической модели компьютерной программы исследуется аппарат функциональных грамматик. Основной базисной операцией функциональных грамматик является операция отождествления, которая эффективна для построения алфавитных отображений над множествами слов, порождаемых контекстно-свободными грамматиками. Используя построенную математическую модель языка программирования в виде универсальной алгебры  $(M, \Omega)$ , строится конкретная формальная математическая модель программы, которая является компьютерной моделью функционирования динамической системы.

**2 О построении отображений над множествами слов, порожденных контекстно-свободными грамматиками**

Пусть задан конкретный и известный язык программирования  $L$  (Pascal, Си ++ и т. д.). Математическая модель языка программирования представляется в виде универсальной алгебры при помощи аппарата функциональных грамматик.

Основание функциональных грамматик состоит из следующих примитивов:

1. Контекстно-свободная грамматика: алгоритм синтаксического анализа, выводы, сентенциальные фразы и выводы из них и т. д.
2. Операция левого отождествления.
3. Операция выбора «либо»  $|$ .
4. Рекурсия.
5. Многоуровневость  $G_1, G_2$  и т. д.
6. Суперпозиция (есть дерево, в узлах которого стоят функции). Если функция, находящаяся в узле, дает результат, имеющий метатип **знач**, то данный результат передается по выходной ветви узла дерева как аргумент к следующей функции.
7. Операция Eval (первый аргумент есть функция, второй аргумент — исходные данные этой функции). Eval применяет функцию к исходным данным и получает результат. Данный результат имеет метатип **знач**.
8. Метатип **знач** является типом результата при выполнении функции.
9. Метатип **функ** — тип композиций функций.
10. Метатип **текст** — текст программы в заданном языке.
11. Арифметические операции. В языке считаются заданными множество целых чисел; операции сложения, вычитания, умножения и целочисленного деления в десятичной арифметике и операции сравнения  $<, \leq, =, \geq, >$ .

С точки зрения универсальной алгебры язык  $L$  будет представлять собой конечную совокупность  $\{\Phi_i\}_{i \in \mathbb{N}}$ , которая будет являться сигнатурой операций алгебры. Тогда любая программа  $x \in L$  будет представляться с помощью синтаксической грамматики  $G_0$  в виде суперпозиции базисных функций из сигнатуры алгебры  $\{\Phi_i\}_{i \in \mathbb{N}}$ .

Если программа  $x$  не содержит синтаксических ошибок и синтаксический разбор пройдет успешно, то получим алгебраический терм, который является семантикой программы  $x$ . Программа  $x$  является отображением их входных данных в выходные данные. Суперпозиция базисных функций программы  $x$  (алгебраический терм) является математической моделью данной программы  $x$ , а семантика языка программирования

представляет собой универсальную алгебру  $A$  с сигнатурой операций  $\Omega = \{\Phi_i\}_{i \in \mathbb{N}}$ . Таким образом, если  $L$  — это множество синтаксически правильных программ типа (текст),  $T(\Omega)$  — множество суперпозиций базисных функций из  $\Omega$ ,  $\sigma$  — алгоритм синтаксического разбора, то  $\sigma : L \rightarrow T(\Omega)$  — взаимно однозначное отображение,  $\forall x \in L \sigma(x) \in T(\Omega)$ .

Из примитивов наиболее интересным и существенным является операция отождествления. Операцию отождествления для формальных грамматик впервые ввел Виталий Алексеевич Тузов [1; 2].

Пусть  $\varphi, \psi$  — слова в алфавите  $V_N \cup V_T$  (сентенциальные формы грамматики  $G$ ), где нетерминальные символы помечены индексами, которые являются формальными параметрами операции отождествления. Тогда операция отождествления синтаксически представляется в виде  $(\varphi)\psi$  и содержательно определяется как отображение совокупности входных формальных параметров, принадлежащих к  $\varphi$ , на совокупность выходных (т. е. результирующих) формальных параметров, входящих в  $\psi$ . Семантически операция отождествления  $(\varphi)\psi$  трактуется так: если  $x$  — фактический аргумент этой операции, то строится вывод  $\varphi \Rightarrow^* x$  этого слова в грамматике  $G_M$ . Тогда нетерминальные символы слова  $\varphi$  получают в качестве значений некоторые подслова слова  $x$  (одинаковые нетерминалы, помеченные одинаковыми символами, должны принимать одни и те же значения), слово  $\psi$  определяет результат этой операции [1].

Все операции начинаются с исходных данных (что дано). Любая операция имеет исходные данные и результат. Сказанное продемонстрируем на примере.

Пусть задана грамматика  $G = (V_N, V_T, P, A)$ , где  $A \in V_N$  — начальный нетерминальный символ.

Обозначим за:  $\varphi$  — исходные данные,  $\psi$  — схема получения результата.

Синтаксически операция отождествления выглядит следующим образом:

$$(\varphi)\psi, \quad \text{где } \varphi, \psi \in (V_N \cup V_T)^*.$$

Рассмотрим подробнее исходные данные операции отождествления на примере:

$$\varphi = N_1\tau_1 N_2\tau_2 N_3\tau_3, \quad \text{где } N_i \in V_N, \tau_i \in V_T^*.$$

(в более общем случае  $\varphi = N_1\tau_1 N_2\tau_2 \dots N_m\tau_m$ ). Отметим, что  $\varphi$  называется сентенциальной строкой грамматики  $G$ . Рассмотрим строку  $x \in V_T^*$  в данной грамматике  $G$ . Запись

$$\varphi \Rightarrow_*^G x$$

будет обозначать вывод из сентенциальной строки  $\varphi \in (V_N \cup V_T)^*$  в строку  $x$ , состоящую из терминальных символов  $V_T$ , т. е. ( $x \in V_T^*$ ). На рис. 1 вывод  $\varphi \Rightarrow_*^G x$  будет иметь вид:

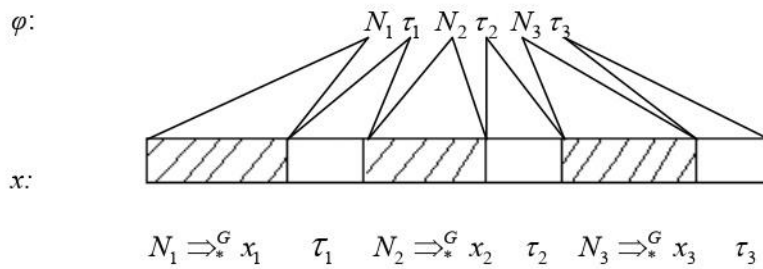


Рис. 1

На этой диаграмме  $x_i$  ( $i=1,3$ ) являются подстрокой строки  $x$ . Процесс вывода начинается слева направо по очереди:  $N_1 \Rightarrow_*^G x_1$ , затем тождественное отображение  $\tau_1 \in V_T^*$  переходит на остаток строки  $x$  справа от  $x_1$ , затем  $N_2 \Rightarrow_*^G x_2$  и т. д. (рис. 1). Этот грамматический разбор является левым, т. к. разбор начинается слева направо и задает однозначность операции отождествления. В результате этого разбора (на данном этапе) нетерминалы получают следующие значения:  $N_1 = x_1$ ,  $N_2 = x_2$ ,  $N_3 = x_3$ . Так как множество нетерминалов из  $\varphi$  содержит множество нетерминалов  $\psi$ , то  $\psi$  получают однозначное значение:  $N(\varphi) \supseteq N(\psi)$ .

Вышеизложенное легко можно обобщить для  $m$ -го порядка:

$$\varphi = N_1\tau_1 N_2\tau_2 \dots N_{m-1}\tau_{m-1} N_m\tau_m, \text{ где } \tau_i \in V_T^* \text{ (м.б. пустой строкой)}$$

$$\psi = t_1 N_{i_1} \dots t_{i_{m-1}} N_{i_m} t_{i_m}, \text{ где } t_{i_j} \text{ могут быть пустыми строками.}$$

$$\underbrace{\{N_1, N_2, \dots, N_m\}}_{\varphi} \supseteq \underbrace{\{N_{i_1}, N_{i_2}, \dots, N_{i_m}\}}_{\psi}.$$

Областью определения операции отождествления является множество  $D_\varphi = \{x \mid x \in V_T^*, \varphi \Rightarrow_*^G x\}$ .

Продemonстрируем на содержательном примере применение операции отождествления. Для этого рассмотрим фрагмент программы:

**начало**

имя имя цел S, SQ; \ \ <описания>  $\Phi_2$

S := 2; SQ := 1; \ \ <операторы>

SQ := SQ + S \ \

<программа>  $\Phi_1$

**конец**.

Интуитивно понятно, что это отображение исходных данных на множество результативных данных  $\Phi_1 : D_\varphi \rightarrow D_\psi$  ( $D_\varphi$  — область определения,  $D_\psi$  — область результативных значений).

Естественным было бы рассмотреть отображение всей программы  $\Phi_1$  как композицию функций. Поскольку синтаксис программы подчиняется синтаксическому правилу

<программа> := **начало** <описания> ; <операторы> **конец**  $\Phi_1$ ,

которое разбивает программу на два раздела <описания> и <операторы>, то естественным было бы построить отдельную функцию для <описания> —  $\Phi_2$  и <операторы> —  $\Phi_4$ . Тогда функция  $\Phi_1$  является композицией функций  $\Phi_2$  и  $\Phi_4$  :  $\Phi_1 = \Phi_2 \circ \Phi_4$  (прим. композиция выполняется слева направо, обычно в математике  $\Phi_1 = \Phi_4(\Phi_2$  (исходные данные))).

Эту идею разбивки отображений и предложений в виде композиций более «мелких» функций можно рекурсивно продолжить по правилам синтаксической грамматики. Рассмотрим дерево грамматики (рис. 2).

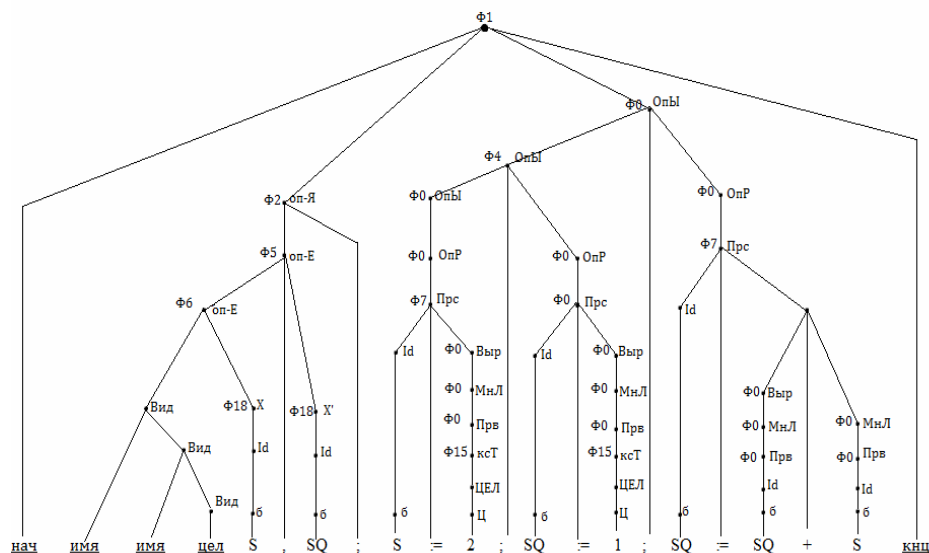


Рис. 2

### Заключение

Разработан подход к компьютерному моделированию динамических систем с помощью функциональных грамматик. Компьютерную программу удалось представить как суперпозицию базисных функций. В работе рассмотрена операция отождествления, которая является существенной при разработке моделирующих программ. Представлен фрагмент компьютерной программы, на которой выполнена операция отождествления и получен терм — суперпозиция базисных функций. Также построено дерево компьютерной программы, в узлах которой находятся базисные функции.

### Литература

1. Тузов В. А. Математическая модель языка. Л.: Изд-во ЛГУ, 1984. 176 с.
2. Тузов В. А. Подход к построению универсальной схемы языка. Синтаксис // Программирование. 1980. № 5. С. 17–25.

## MODELING OF A DYNAMIC SYSTEM BY FUNCTIONAL GRAMMARS

*Darima D. Nikolaeva*

Research Assistant,  
Dorzhi Banzarov Buryat State University  
24a Smolina St., Ulan-Ude 670000, Russia  
E-mail: darisha89@yandex.ru

*Dashadondok Sh. Shirapov*

Dr. Sci. (Phys. and Math.), Prof.,  
Dorzhi Banzarov Buryat State University  
24a Smolina St., Ulan-Ude 670000, Russia  
E-mail: shir48@mail.ru

*Vladimir D. Anakhin*

Dr. Sci. (Engineering), Prof.,  
Dorzhi Banzarov Buryat State University,  
24a Smolina St., Ulan-Ude 670000, Russia  
E-mail: anakhin@mail.ru

The article considers the problem of modeling a class of computer programs that generate computational processes and adequately describe the operation of a dynamic system. To solve this problem, we use the means of functional grammars. Identification is the basic operation of functional grammars. The article gives an example of using the identification operation on a fragment of the computer program that simulates a dynamic system. The end result is the obtaining of basic functions superposition with the involvement of the identification operation. The superposition obtained is an algebraic model of dynamic system, presented in the form of a computer program. We have constructed a parse tree for the computer program, which nodes contain the basic functions (non-terminals). If nodes require a representation in the form of **functions** (an object is the function), then this node is interpreted as a superposition of basic functions. If in nodes a representation in the form of **text** is required, then an object is the fragment of the program text corresponding to non-terminal (non-terminal in this node is a fragment of the program text). Also, nodes may require a representation in the form of **values** (an object is the result of the above superposition of functions corresponding to non-terminal).

*Keywords:* dynamic systems; modeling; algebraic model of language; universal algebra; context-free grammar; recursion; interpreter; semantics.

## References

1. Tuzov V. A. *Matematicheskaya model yazika* [A Mathematical Model of Language]. Leningrad: Leningrad State University Publ., 1984. 176 p.
2. Tuzov V. A. Podkhod k postroeniyu universalnoi skhemy yazika. Sintaksis [An Approach to the Construction of Universal Language Scheme. Syntax]. *Programming and Computer Software*. 1980. No. 5. Pp. 17–25.