

МЕТАЯЗЫК НОРМАЛЬНЫХ ФОРМ ЗНАНИЙ

Аннотация. Представлен метаязык нормальных форм знаний, даны его текстовое и графическое описания. В текстовой и графической формах даны формальные описания множества лингвистических примеров, в том числе синтаксиса подмножества простых предложений английского языка и метаязыка Extended Backus–Naur Form. Сделано заключение об адекватности предложенного метаязыка возможностям человека представлять и использовать знания.

Ключевые слова: синтаксис, метаязык нормальных форм знаний, графические средства метаязыка, формальное описание метаязыка, метаязык Extended Backus–Naur Form, синтаксис английского предложения.

ВВЕДЕНИЕ

Обычно человеком воспринимается лишь то, что имеет форму; объекты без формы просто не существуют. Любая форма нами воспринимается, когда представляем ее структуру на основании знаний о синтаксисе языка для описания множества форм объектов некоторого класса.

Для строгого и точного описания синтаксиса языков используют специальные метаязыки (языки для описания языков) [1]. Наиболее распространены металингвистичные формы BNF (Backus–Naur Form) и Extended BNF (EBNF) — расширенный метаязык [2].

Однако в качестве языка представления знаний метаязык EBNF (и другие известные метаязыки) имеет определенные недостатки. В частности, язык EBNF, созданный первично для узкоспециальных задач и хорошо их обеспечивающий, не является функционально полным и поэтому непригоден для представления произвольных знаний.

Цель настоящей статьи — устранение недостатков известных метаязыков в рамках нового способа представления и использования знаний, предложенного в [1, 3], и его применение для описания множества лингвистических примеров.

НОВЫЙ МЕТАЯЗЫК ОПИСАНИЯ ЗНАНИЙ

Метаязыком BNF, его расширением или произвольным другим метаязыком представляют спецификацию синтаксических структур произвольного объектного языка в виде системы правил — формул, определяющих отношение между терминалами и нетерминалами. Терминалы — элементы структуры, не имеющие собственной структуры, это определенные вне BNF-описания идентификаторы или цепочки — последовательности символов в одинарных или двойных кавычках. Нетерминалы — элементы структуры, имеющие и собственные имена, и структуру. Полным описанием структуры является набор правил, определяющих все нетерминалы таким образом, чтобы каждый из них мог быть сведен к комбинации терминалов с последовательным (рекурсивным) применением правил.

Согласно спецификации объектного языка разрабатывают в языке реализации программу распознавателя (parser) — анализа и построения дерева вывода (разбора) утверждений объектного языка. Факт построения дерева разбора подтверждает принадлежность входной цепочки символов данному языку.

В [3] предложено решение задачи синтеза нового способа представления и использования знаний, адекватного процессу решения структурно сложных задач. Согласно этому конечный программный продукт состоит из двух частей.

Первая часть имеет информационную структуру множества определений понятий (нетерминалов), связанных базовыми отношениями нормальных форм знаний (НФЗ), развивающих выразительные возможности метаязыка EBNF. Вторая часть включает два подмножества — множество элементарных алгоритмов и множество элементарных структур данных. Эти множества реализуют на одном из традиционных языков программирования в виде единой библиотечной системы. Каждый терминал имеет собственное имя и в описании структуры может использоваться наравне с нетерминальными понятиями.

Для достижения функциональной полноты в метаязык EBNF вводятся такие отношения, как отрицание, применимое к любому понятию в описании структуры (обозначается символом ^, предшествующим имени отрицаемого понятия), а также три элементарных операции: распознавание, распознавание со следом и порождение над информационными структурами, моделями прикладных областей, обозначаемые соответственно знаками ?, # и !, присоединяемыми к имени понятия, которое вводит некоторую информационную структуру.

Самоопределение метаязыка НФЗ:

описание = определение (определение);
определение = инверсия имя_понятия тело_определения тзпт;
инверсия = отрицание / истина;
имя_понятия = идентификатор / целое / цепь_знаков;
идентификатор = буква (буква / цифра);
целое = цифра (цифра);
цепь_знаков = знак (^метазнак знак);
тело_определения = структура / терминал;
структура = есть_структура выражение;
терминал = есть_структура;
выражение = элемент (ИЛИ элемент / И элемент);
элемент = инверсия имя режим;
имя = итерация / строка / имя_понятия;
режим = зн_режим / истина;
зн_режим = ? / # / ! ;
итерация = скобка_откр выражение скобка_закр;
строка = кавычка (знак) кавычка;

где терминальные понятия имеют значения:

есть_структура — разделитель, изображается символом = ;
ИЛИ — отношение альтернативного выбора изображается символом / ;
И — отношение конкатенации, изображается символом пробел;
скобка_откр — итерационная скобка, изображается символом (;
скобка_закр — итерационная скобка, изображается символом) ;
отрицание — отношение отрицания, изображается символом ^ ;
тзпт — конец определения, изображается символом " ;
кавычка — текстовая кавычка, изображается символом ' ;
истина — отношение "тождественная истина", пустой объем.

В этом описании понятия "буква", "цифра", "знак", "метазнак" — терминальные:

буква = A / Б / В / Г / ... / а / б / в / г / ... / я / А / В / С / ... / а / б / с / ... / z ;
цифра = 0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 ;
знак = - / / (/ * / & / ^ / % / \$ / # / @ / ! / ~ / " / ' / ; / : / < / > / , / . / = / _ / ? / // ;
метазнак = (/) / пробел / / / = / ? / # / ! / ; / ' ;

ГРАФИЧЕСКОЕ ОПИСАНИЕ МЕТАЯЗЫКА НФЗ

Наряду с текстовыми используются и графические метаязыки, среди которых наиболее популярен язык диаграмм Н. Вирта (рис. 1).

Графические средства метаязыка НФЗ развивают графические средства диаграмм Н. Вирта так же, как НФЗ развивает метаязык EBNF (рис. 2).

В табл. 1 приведено соответствие между текстовой и графической формами представления знаний средствами метаязыка НФЗ для множества разных понятий: Numeral (десятичная цифра), Regular (последовательность единиц за последовательностью нулей), Natural (натуральное десятичное число), Integer (десятичное целое число), RealValue (действительное число), Letter (буква), Identifier (идентификатор) и character (знак).

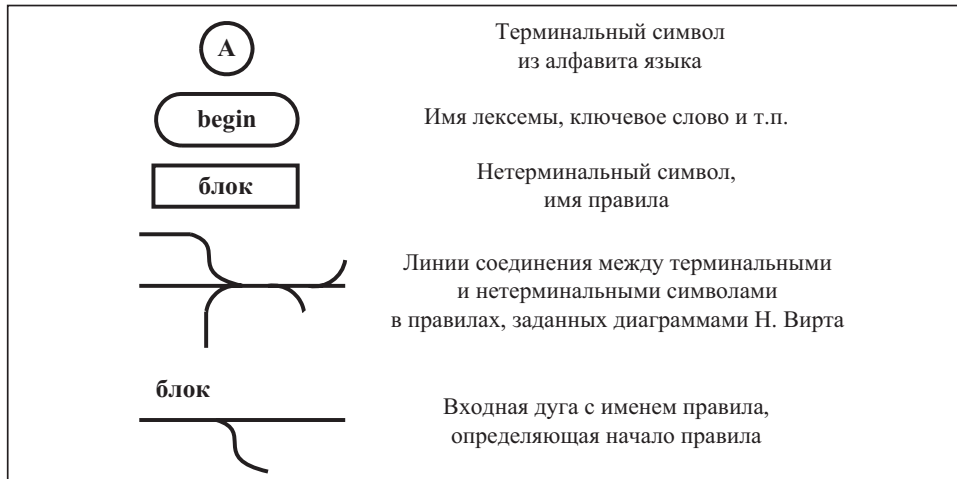


Рис. 1. Графические обозначения метасимволов диаграмм Н. Вирта

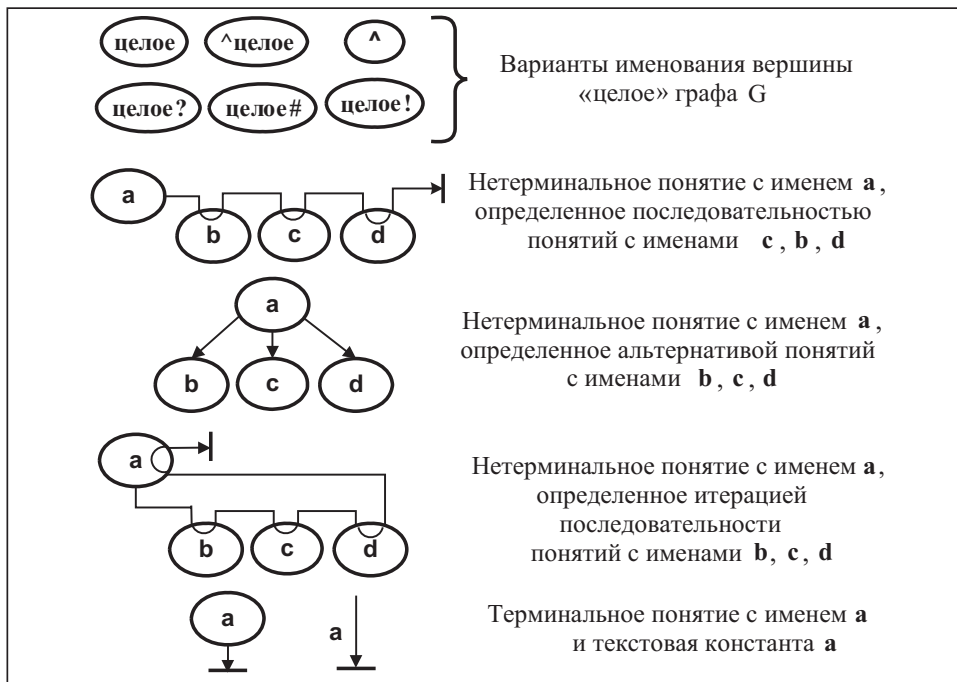
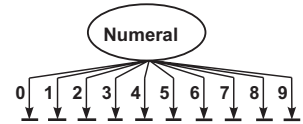
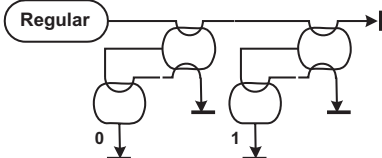
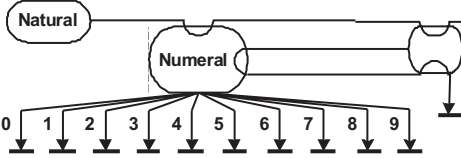
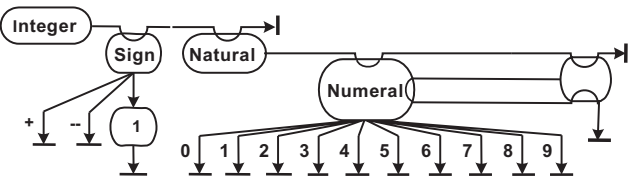
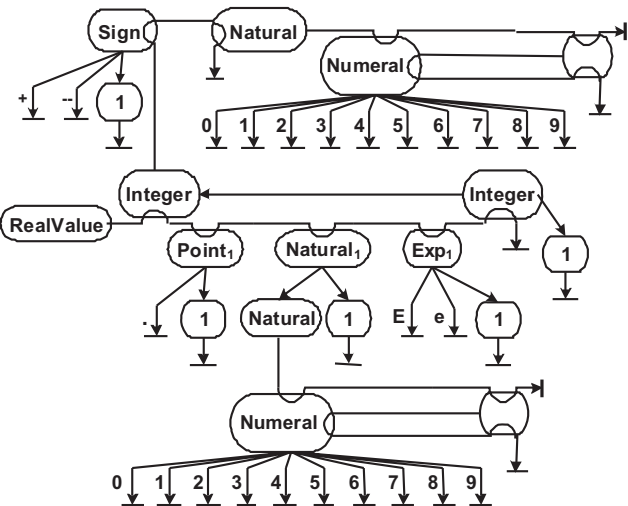
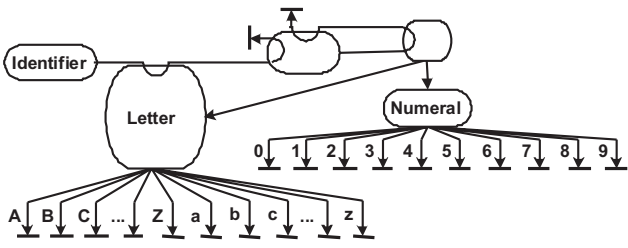


Рис. 2. Графические средства метаязыка НФЗ

Таблица 1. Простые структуры в представлении НФЗ

Текстовое представление знаний	Графическое представление знаний
Numeral = 0/1/2/3/4/5/6/7/8/9;	
Regular = (0)(1);	
Numeral = Numeral (Numeral);	
Integer = Sign Numeral; Sign = + / - / 1; Numeral = Numeral (Numeral);	
RealValue = Integer Point ₁ Natural ₁ Exp ₁ Integer ₁ ; Point ₁ = . / 1; Natural ₁ = Natural / 1; Exp ₁ = E/e/1; Integer ₁ =Integer / 1;	
Identifier = Letter (Letter / Numeral);	

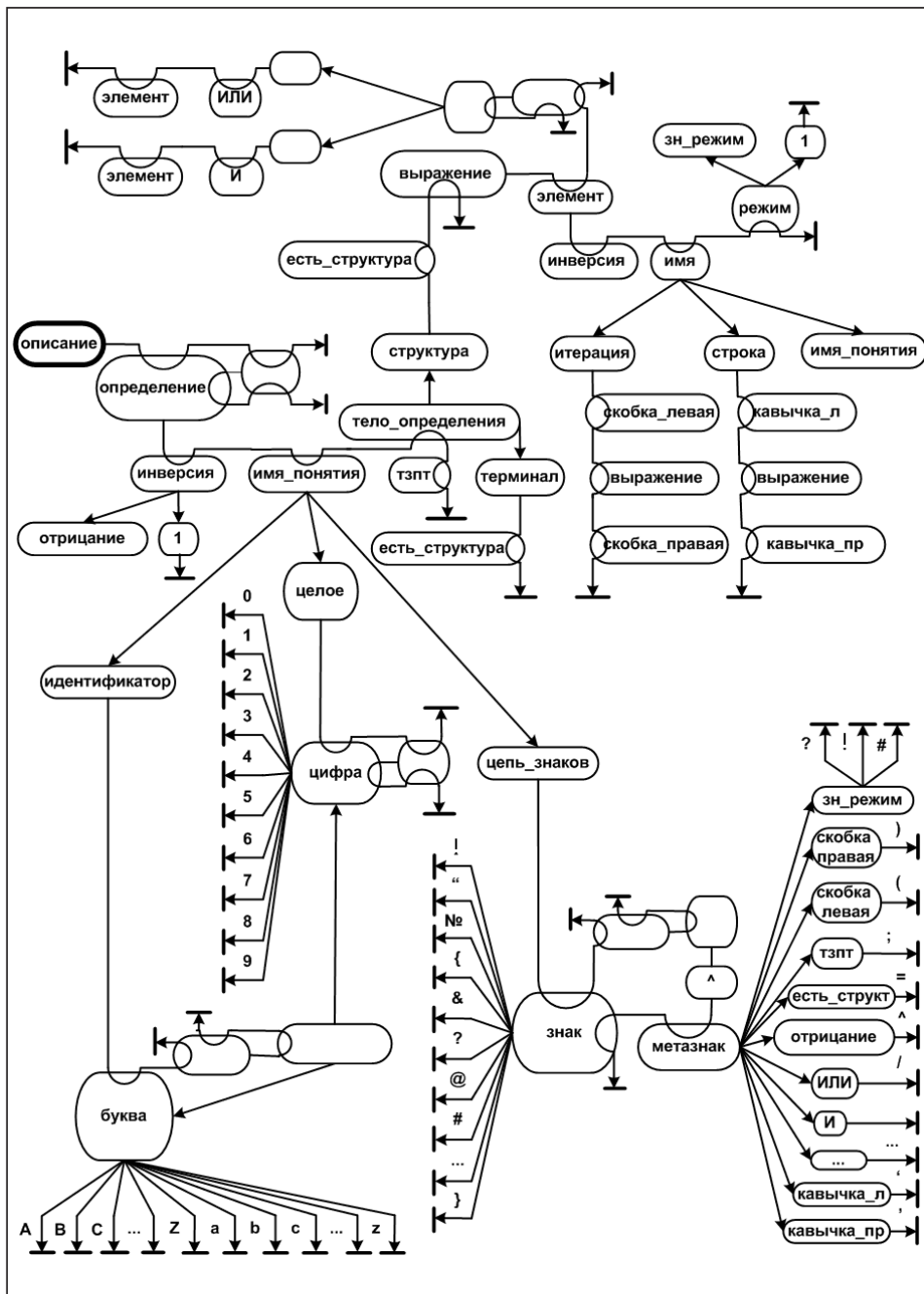


Рис. 3. Графическое описание метаязыка НФЗ

На рис. 3 с использованием графических средств метаязыка НФЗ представлена графическая форма самоописания метаязыка НФЗ, эквивалентная приведенному выше текстовому описанию.

ПРИМЕР МЕТАЯЗЫКОВОГО ОПИСАНИЯ АНГЛИЙСКОГО ЕСТЕСТВЕННОЯЗЫКОВОГО ПРЕДЛОЖЕНИЯ

Используя результаты работы [4], представим синтаксис подмножества простых предложений английского языка в нотации метаязыка НФЗ следующим образом:

Sentence = NP Sep₁ Aux₁ Sep₁ VP;
 NP = Art₁ Sep₁ AP₁ Sep₁ N Sep₁ PP₁;
 Sep₁ = (Sep);
 Aux₁ = Aux/1;

Aux = 'will'/'can'/'might';
 VP = Verb Sep₁ NP₁ Sep₁ PP₁ Sep₁ Adv₁;
 Art₁ = Art/1;
 Art = 'the'/'an'/'a';
 AP₁ = AP/1;
 AP = Adj Sep₁ PP₁;
 Adj = 'old'/'red'/'slimy'/'white'/'new'/'hungry';
 PP₁ = PP/1;
 PP = Prep Sep₁ NP;
 Prep = 'at'/'in'/'to'/'with'/'out_of';
 N = 'tree'/'wind'/'children'/'toys'/'toy'/'box'/'boy'/'ball'/'house'/'shorts'/'letter'
 /'jon'/'it'/'man'/'hat'/'fish'/'toby'/'he'/'book'/'we'/'dogs'/'swimmer';
 Verb = 'swayed'/'put'/'found'/'kicked'/'was_reading'/'gave'/'read'/'fed'/'was'
 /'was'/'pulled'/'have_been_reading';
 NP₁ = NP/1;
 Adv₁ = Adv/1;
 Adv = 'slowly'/'now'/'quite_quick';

В этом описании использованы следующие понятия: Sentence — предложение английского языка, NP — группа существительного, VP — группа глагола, PP — группа предлога, AP — группа прилагательного, N — существительное, Verb — глагол, Adj — прилагательное, Prep — предлог, Adv — наречие, Art — артикль, Aux — вспомогательный глагол, Sep — пробел и др.

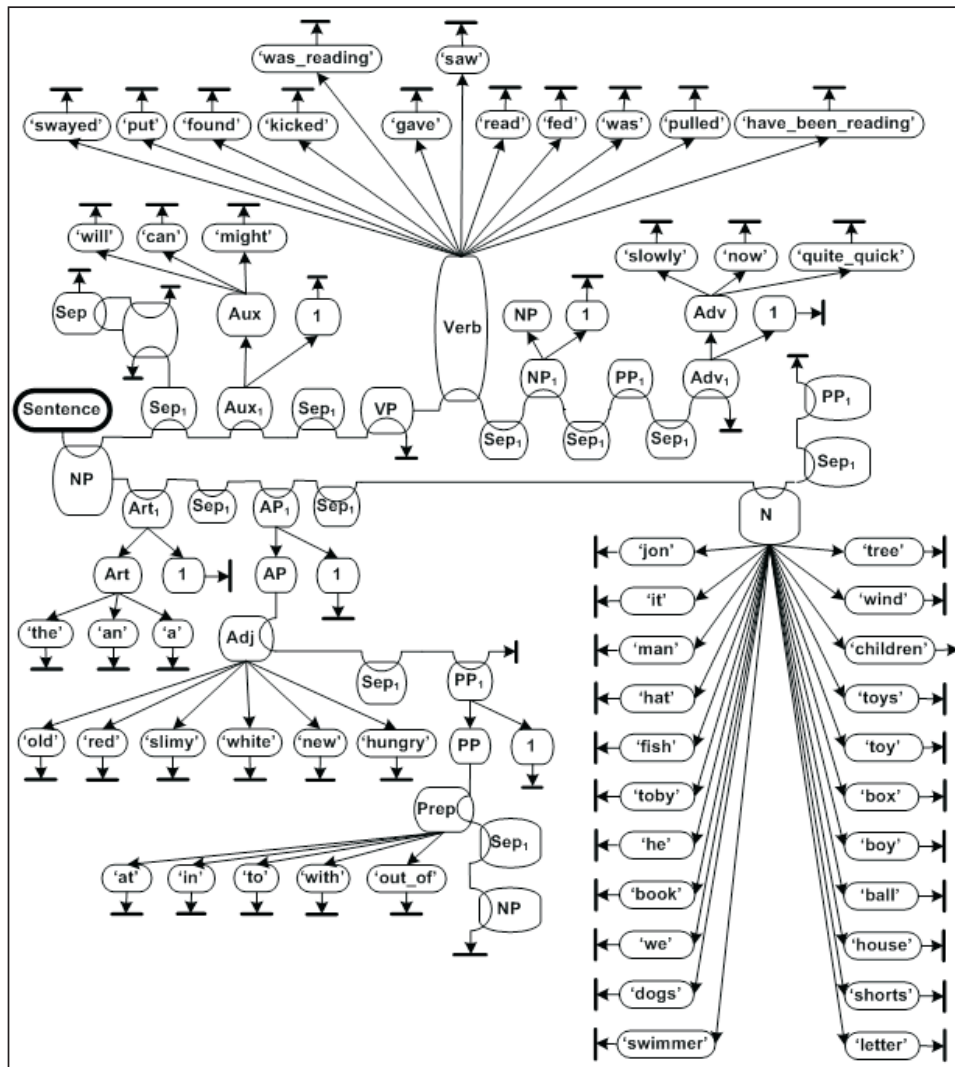


Рис. 4. Графическое описание примера синтаксиса английского предложения

Описание приведенного примера синтаксиса предложения английского языка, выполненное графическими средствами метаязыка НФЗ, изображено на рис. 4. Очевидно, что это описание эквивалентно приведенному выше текстовому описанию.

МЕТАЯЗЫК EBNF

Семантика правила в EBNF: нетерминал, заданный идентификатором слева от знака =, определяется некоторым отношением нетерминалов и терминалов, указанным справа от этого знака. Набор основных отношений EBNF: конкатенация (обозначается запятой), альтернативный выбор (обозначается вертикальной чертой, разделяющей альтернативы), итерация (обозначается фигурными скобками, выделяющими итерируемую структуру) и отрицание (обозначается тире с последующим исключаемым термином); набор дополнительных, стилистических отношений — необязательность (необязательный элемент выделяют квадратными скобками) и структурные круглые скобки. Каждое правило имеет явный заключительный символ для исключения неоднозначности его окончания. Пара символов (* и *) выделяет комментарий, размещаемый обычно или перед описанием синтаксиса, или в конце определения нетерминала. Метаязык EBNF пригоден для описания синтаксиса произвольных языков, имеющих практический интерес.

Средствами EBNF может быть определен и сам метаязык EBNF. Так, основная его часть представляется в следующей форме [2]:

```
syntax = [comment], syntax rule, { syntax rule};
syntax rule = meta identifier, '=', definitions list, [comment], ';' ;
definitions list = single definition, {'|', single definition} (* | разделяет альтернативы <single definitions> *);
single definition = term, {';', term} (*, разделяет последовательные термины <term>*);
term = factor, ['-', exception] (*<term> определен через <factor> без <exception> *);
exception = factor (*если <factor> не содержит <meta identifier>, определяемого термина*);
factor = [integer, '*'], primary (* <integer> указывает число повторений <primary>*);
primary = optional sequence | repeated sequence | special sequence | grouped sequence | meta identifier | terminal string | empty;
empty = ;
optional sequence = '[' , definitions list , ']' (* Скобки [ и ] содержат необязательные символы *);
repeated sequence = '{', definitions list , '}' (* Скобки { и } содержат повторяемые символы *);
grouped sequence = '(', definitions list , ')' (* Скобки ( и ) группируют любой <definitions list>*);
terminal string = '"', character - '"', {character - '"'}, '"' | "'", character - "'", {character - "'", '"'} (*<terminal string> представляет <characters> между кавычками ' _ ' или " _ " *);
meta identifier = letter, {letter | decimal digit} (*<meta identifier> — имя определяемого термина *);
integer = decimal digit, {decimal digit};
special sequence = '??', {character - '??'}, '??' (* <special sequence> не определено в метаязыке *);
comment = '(*', {comment symbol}, '*)' (* <comment> может быть, например, в <syntax rule> и/или <syntax> *);
comment symbol = comment | terminal string | special sequence | character;
```

Здесь и далее (в отличие от нотации правил) в комментариях и тексте имена терминов даны в угловых скобках для их различия относительно обычного текста.

Метаязык, эквивалентный основной части метаязыка EBNF, определим в терминах метаязыка НФЗ следующей системой правил:

```
syntax = comment rule);
rule = identifier '=' def_list comment ',' ;
def_list = single_def (separator single_def);
single_def = term (space term);
term = if_minus factor;
if_minus = '^minus / ^exception;
minus = '^';
exception = factor;
factor = if_int (int_repet primary);
if_int = integer sum:=integer counter:=0/sum:=0;
int_repet = sum_0/counter ≤ sum increment;
primary = opt_seq/rep_seq/identifier/term_str/empty;
empty = ;
opt_seq = def_list/empty;
rep_seq = (space def_list);
term_str = ""^"" character ( "" character "" "" "" character ( "" character "" );
identifier = letter (letter/numeral);
integer = numeral (numeral);
comment = '( * (comm_symb) * )' / empty;
comm_symb = comment / term_str/character;
```

Для этого описания характерно следующее:

1) согласно нотации метаязыка НФЗ все имена терминов представлены в однословном варианте с возможным использованием подчеркивания для соединения в одно слово двух и более слов многословных имен метаязыка EBNF;

2) использованы следующие сокращения имен из самоопределения метаязыка EBNF:

```
<rule> — вместо <syntax rule>;
<identifier> — вместо <meta identifier>;
<def_list> — вместо <defininitions list>;
<single_def> — вместо <single definition>;
<opt_seq> — вместо <optional sequence>;
<rep_seq> — вместо <repeated sequence>;
<term_str> — вместо <terminal_string>;
<numeral> — вместо <decimal digit>;
<comm_symb> — вместо <comment symbol>;
<st_comm_symb> — вместо <start-comment-symbol>;
<end_comm_symb> — вместо <end-comment-symbol>;
<first-q> — вместо <first-quote-symbol>;
<second-q> — вместо <second-quote-symbol>;
<defining> — вместо <defining-symbol>;
<terminator> — вместо <terminator-symbol>;
```

3) термин <special sequence> исключен, поскольку представляется мало продуктивным, тем более что он не определен в метаязыке EBNF:

4) термин <grouped sequence> исключен, поскольку представляется непродуктивным. Его содержимое легко заменить дополнительным термином;

5) определение в метаязыке EBNF термина term:

a) term=factor, ['-', exception];

b) замена семантически эквивалентным его определением в метаязыке НФЗ:

```
term = if_minus factor;
if_minus = '^minus / ^exception;
minus = '^';
```

c) имеет следующий смысл: если нет знака <minus> или <exсption>, то выполняется <factor>, иначе — ложь;

б) определение в метаязык EBNF термина factor:

```
factor = [integer, '*'], primary;
```

а) замена семантически эквивалентным его определением в метаязыке НФЗ:

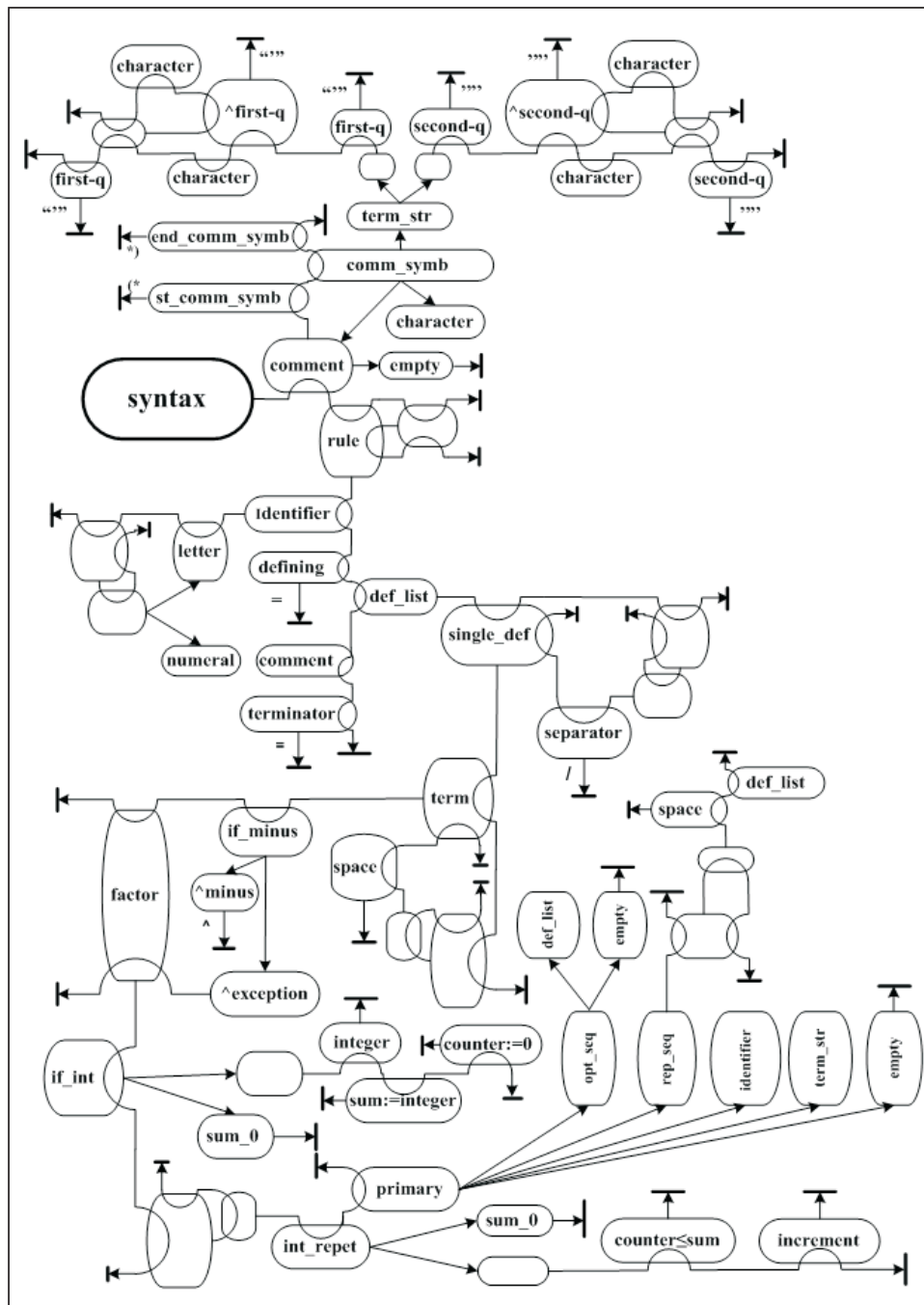


Рис. 5. Графическое описание метаязыка EBNF

factor = if_int (int_repet primary);
 if_int = integer sum := integer counter := 0/sum := 0;
 int_repet = sum_0/counter ≤ sum increment;

б) имеет следующий смысл: если $\langle integer \rangle$ есть, то выполняется $\langle sum := integer \rangle$ и $\langle counter := 0 \rangle$, иначе выполняется $\langle sum := 0 \rangle$. Далее, если значение $\langle sum \rangle$ есть 0, то $\langle primary \rangle$ повторяется неопределенное число раз, иначе число повторений $\langle primary \rangle$ определяется заданным значением $\langle integer \rangle$.

Описание метаязыка EBNF с использованием графических средств метаязыка НФЗ представлено на рис. 5. Очевидно, что оно эквивалентно приведенному выше текстовому описанию.

Факт описания метаязыка EBNF в метаязыке НФЗ обосновывает утверждение.

Утверждение. Всякий язык-объект, описанный в метаязыке EBNF, может быть описан в метаязыке НФЗ.

ЗАКЛЮЧЕНИЕ

В предложенном способе представления и использования знаний предельно отделены знания от данных. В описании и представлении знания и данные отделены физически: знание сконцентрировано в информационной структуре, представление и обработка данных локализованы в терминалах. Эти две компоненты представлены двумя разными подпространствами единого пространства решения задач. Разделение пространств и механизмов обработки знаний и данных позволяет применить для каждого из них оптимальные аппаратные средства интерпретации при сохранении целостности всего двуединого процесса решения задач.

Знание — описание объективно существующей информационной структуры предметной области. С использованием этих знаний можно решать все задачи, существующие для этой области (если достаточна точность описания предметной области). Иными словами, знания — это компонента информации, инвариантная относительно задач, решаемых в этой области. Итак, знания, когда-то осознанные, формализованные и введенные в компьютер, можно использовать многократно для решения различных задач.

Этот способ представления и использования знаний в отличие от известных характеризуется большей технологической адекватностью возможностям человека.

СПИСОК ЛИТЕРАТУРЫ

1. Кургаев А.Ф., Григорьев С.Н. Нормальные формы знаний // Доклады НАН Украины. — 2015. — № 11. — С. 37–44.
2. International Standard ISO / IEC 14977 : 1996(E). — <http://www.cl.cam.ac.uk/~mgk25/iso-14977.pdf>.
3. Спосіб представлення і використання знань / О.П. Кургаєв, С.М. Григор'єв / Патент на корисну модель UA 92484 U, 2014, Бюл. № 16.
4. Fabb N. Sentence structure. Second edition. — New York: Nigel Fabb Copyright, 2005. — 122 p.

Надійшла до редакції 23.02.2016

О.П. Кургаєв, С.М. Григор'єв

МЕТАМОВА НОРМАЛЬНИХ ФОРМ ЗНАТЬ

Анотація. Представлена метамова нормальних форм знань, дано її текстовий й графічний описи. У текстовій і графічній формах дано формальні описи множини лінгвістичних прикладів, у тому числі синтаксису підмножини простих речень англійської мови й метамови Extended Backus–Naur Form. Зроблено висновок про адекватність представленої метамови можливостям людини щодо подання й використання знань.

Ключові слова: синтаксис, метамова нормальних форм знань, графічні засоби метамови, формальний опис метамови, метамова Extended Backus–Naur Form, синтаксис англійського речення.

A.F. Kurgaev, S.N. Grigoriev

META-LANGUAGE OF NORMAL FORMS OF KNOWLEDGE

Abstract. The meta-language of normal forms of knowledge is presented and its graphic and text descriptions are provided. The formal descriptions of numerous linguistic examples (including syntax of a subset of simple English sentences and the meta-language of extended Backus–Naur forms) are set out in text and graphic forms. It has been concluded that the meta-language presented can be favorably compared to human capabilities of knowledge representation and usage.

Keywords: syntax, meta-language of Normal Forms of Knowledge, graphic tools of the meta-language, meta-language formal description, Extended Backus–Naur Form, syntax of English sentence.

Кургаєв Александр Филиппович,

доктор техн. наук, профессор, ведущий научный сотрудник Института кибернетики им. В.М. Глушкова НАН Украины, Киев, e-mail: afkurgaev@ukr.net.

Григорьев Сергей Николаевич,

соискатель, Институт кибернетики им. В.М. Глушкова НАН Украины, Киев, e-mail: Sergey@Grigoriev.kiev.ua.