УДК 004.423+811.93

В. С. Выхованец

Описание семантики контекстно-свободных языков методом математической индукции

Исследуются контекстно-свободные языки. Определение семантики контекстно-свободного языка осуществляется путем последовательного описания правил вывода формальной грамматики этого языка и индуктивного выражения семантики каждого такого правила через базовые и ранее определенные семантические категории, представленные в виде текстов на определяемом языке. Для определения базовых семантических категорий используется внешний интерпретатор. Рассматривается метод разнесенного грамматического разбора, предназначенный для повышения эффективности обработки текстов, порожденных неоднозначными грамматиками.

ВВЕДЕНИЕ

На данный момент не существует универсального общепринятого подхода для формального описания семантики. Разработано множество различных моделей и методов:

- грамматические модели, основанные на добавлении расширений к грамматике, определяющей этот язык (атрибутные грамматики [1], императивный или операционный метод [2], венский метод [3], W-грамматики [4]);
- аппликативные модели, в которых непосредственно конструируется определение функции, которую вычисляет каждая программа, написанная на этом языке (аксиоматический метод [5], метод денотационной семантики [6], метод функциональной семантики [7]);
- модели спецификаций, в которых описываются отношения между различными функциями языка и, если программа реализует эти отношения, то она корректна по отношению к спецификации (метод спецификаций [8], алгебраический метод [9]).

Во всем перечисленных методах описание семантики осуществляется путем выделения некоторой типологии смыслов – категорий значений (семантических категорий, примитивов), через которые и посредством которых на некотором семантическом языке задается более сложный смысл.

Для описания семантики контекстно-свободных языков упоминаются три наиболее распространенных метода [10]: продукционной (дедуктивной, аксиоматической), денотационной (математической) и операционной семантики, которые являются фактически единственными потенциально пригодными. Последнее обстоятельство объясняется гибкостью этих методов, возможностью статического расширения набора семантических категорий (математических объектов, команд виртуальной машины, и др.), с использованием которых осуществляется описание семантики.

Продукционный метод базируется на исчислении предикатов, где результат вычисления описывается через взаимосвязь переменных до и после применения конструкций языка. Продукционная семантика фактически построена на основе математической логики, ориентирована на человека и предназначена скорее для доказательства правильности программ, чем для формального определения семантики. В качестве примеров продукционных методов описания семантики можно привести аксиоматический метод Хоара [5] и метод индуктивных утверждений Флойда [11].

Денотационный метод основывается на функциональных вычислениях, в которых встроенные операции языка отображаются в однозначные математические объекты, которые затем применяются для описания семантики языковых конструкций [12]. В частности, данный подход иллюстрируется теорией вычислений Скотта, основанной на семантических доменах [13], где сначала перечисляются стандартные и определяемые (конечные) домены, затем, на основе определения конструкторов, задается семантика конструкций языка в виде формулы над доменами и конструкторами. Заметим, что денотационное описание языка используется для порождения компиляторов, однако ни одного работоспособного компилятора на основе этого метода до сих пор не построено [14].

В операционном методе операции языка описываются через команды некоторой абстрактной машины. Смысл выполняемых оператором действий выражается в изменении состояния этой машины. Примерами абстрактных машин является SECD-машина [15] и категориальная машина [16]. Как показала практика, описания, основанные на низкоуровневом моделировании, практически бес-

полезны, поскольку слишком громоздки и неудобны для восприятия человеком [16].

Наибольшее распространение на практике получило операционное описание семантики контекстно-свободных языков в системе генерации синтаксических анализаторов YACC [17, 18]. На вход программы подается контекстно-свободная грамматика, с каждой продукцией которой связывается действие, описанное на языке программирования Си, и выполняемое всякий раз, когда создается узел дерева грамматического разбора, соответствующий этой продукции.

Ни один из перечисленных методов определения семантики не оказался эффективным ни для пользователя, ни для разработчика языка [8]. Операционный метод достаточно удобный для реализации и может быть полезен разработчику, но для пользователя этот метод не имеет большого значения, так как порождаемые им описания содержат слишком много ненужных ему подробностей. Разработчик вряд ли сможет руководствоваться функциональным и денотационным методами, а для пользователя они, как правило, оказываются слишком сложными, чтобы их можно было использовать непосредственно. Пользователю легче понять аксиоматический метод, но при попытке составить полное определение языка он порождает чрезвычайно сложное описание, а для разработчика этот метод и вовсе непригоден.

Несмотря на некоторые успехи в описании семантики формальных языков задача формального определения семантики естественного языка так и не решена по причине его «семантической замкнутости». Семантически замкнутым называется язык, который содержит в себе как выражения, относящиеся к некоторым внеязыковым объектам, так и выражения, относящиеся к характеристике самого языка. Последнее приводит к возникновению в нем противоречий и парадоксов. Например, пытаясь ответить на вопрос, истинно или ложно высказывание «Данное предложение ложно», мы приходим к противоречию.

Однако понятие истинности присуще не объективной действительности, а оно выражает не более чем одно из возможных свойств высказываний [19]. Следовательно, смысл «противоречивых» высказываний может быть различен, в зависимости от контекста их применения. Например, ранее использовалось предложение, которое в одной из своих интерпретаций порождает антиномию. Но это высказывание не разрушило семантическую структуру настоящего текста, так как было применено в особом контексте и для выражения особого смысла.

Таким образом, проблемы описания семантики языков, появляющиеся вследствие возникновения противоречий, вызваны тем, что язык оказывается настолько широким, что позволяет определять внутренне противоречивые ситуации и понятия, например такое, как «множество всех множеств».

Любую формальную теорию, путём введения новых понятий или чрезмерного расширения объёма существующих понятий можно сделать противоречивой. Поэтому расширение понятийного базиса не только в формальной, но и содержательной теории должно сопровождаться тестом на непротиворечивость.

Чтобы избежать возникновения противоречий А. Тарским предложено разделить язык на две части: объективный (предметный) язык, служащий для выражения высказываний, и метаязык, предназначенный для описания семантики этих высказываний [20]. На предметном языке говорят о той или иной предметной области, а на метаязыке обсуждают свойства предметного языка. Благодаря этому разделению в языке не могут появляться предложения, говорящие о самих себе.

В отличие от предметных языков, метаязык является, как правило, некоторым специальным образом созданным языком, не содержащим всякого рода двусмысленностей, препятствующих использованию его в качестве орудия для создания точных высказываний о предметном языке. Однако метаязык, как и любой язык, также нуждается в описании. Для определения метаязыка требуется метаязык более высокого порядка, что приводит к некоторой иерархической структуре метаязыковых средств. По этой причине семантика формальных систем в настоящее время задается путем определения некоторого универсального метаязыка, который замыкает иерархию, строится на основе конечного множества первичных семантических категорий и содержит формальный аппарат, с помощью которого и через первичные категории которого выражается более сложный смысл.

Так как метаязык по своей природе должен быть более выразительным, чем предметный язык, то при его использовании возникли трудности, связанные с необходимостью динамического расширения категориального аппарата в зависимости от того, какие выражения предметного языка встретились при анализе [21].

Суть рассматриваемого подхода состоит в том, что для описания семантики одного из классов формальных языков – контекстно-свободных, применяется метод математической индукции, заключающийся в семантическом замыкании формального языка и в использовании семантических категорий, которые определяются по мере необходимости, в процессе описания формальной грамматики языка и средствами самого языка, где под семантической категорией понимается нетерми-

нальное понятие грамматики определяемого языка.

Базой индукции в этом случае выступают первичные семантические категории, которые объявляются перед использованием, для чего служит аксиома – единственное необходимое и достаточное правило, включаемое в грамматику всех определяемых языков. Если возникает задача описания семантики базовых категорий, то такая задача может быть решена известными методами, например путем непосредственной реализации этих категорий командами целевой вычислительной платформы или определения их семантики на некотором метаязыке.

СИНТАКСИС И СЕМАНТИКА

Определение 1. *Формальным языком L над алфавитом (конечным множеством знаков) T называется произвольное подмножество множества U строк конечной длины в алфавите T,* $L \subseteq U(T)$.

Пример 1. Зададим язык натуральных чисел, представленных в двоичной системе счисления:

$T = \{0,1\}$, $U(T) = \{0, 00, 01, 10, 11, 000, 001, 010, \ldots\}$,

$L = \{0,1,10,11,100,101,110,111,\ldots\}$, $L \subset U(T)$. ♦

Определение 2. *Множество S называется формальной семантикой формального языка L, если задано отображение φ строк языка L в элементы множества S; тогда для строки $\gamma \in L$ элемент $\varphi(\gamma) \in S$ будет ее смыслом.*

Пример 2. Определим семантику языка натуральных чисел L из примера 1 отображением φ его строк в элементы множества натуральных чисел $S = \{0,1,2,\ldots\}$,

$\varphi(0) = 0$, $\varphi(1) = 1$, $\varphi(10) = 2$, $\varphi(11) = 3$, … ♦

В основе описания семантики формальных языков лежит принцип композиционности Г. Фреге [22], заключающийся в определении семантики целого через семантику частей, т.е. между семантическими правилами интерпретации и синтаксическими правилами образования выражений языка устанавливается соответствие.

Синтаксис формальных языков традиционно задается формальными грамматиками, а контекстно-свободным языком называется язык, для которого существует порождающая этот язык контекстно-свободная грамматика.

Определение 3. *Контекстно-свободной грамматикой G называется формальная система $< T, N, P, I >$, задаваемая четырьмя элементами: T – конечным множеством терминальных знаков; N – конечным множеством нетерминальных (вспомогательных) знаков таких, что $T \cap N = \varnothing$; P – конечным множеством правил вывода (продукций) вида $A \to \alpha$, где $A \in N$, α – строка конечной длины в объединенном алфавите грамматики, $\alpha \in U(T \cup N)$; I – начальным нетерминальным знаком (аксиомой) грамматики, $I \in N$.*

Пример 3. Контекстно-свободная грамматика G языка L натуральных чисел в двоичной системе счисления,

$G =< \{0,1\}, \{I, J\}, P, I >$,

где

$P = \{I \to 0, I \to J, J \to 1, J \to J0, J \to J1\}$. ♦

Определение 4. *Выводом в контекстно-свободной грамматике $G =< T, N, P, I >$ называется конечная последовательность строк, которая завершается строкой, не содержащей нетерминальных знаков, и получаемая из аксиомы грамматики I путем последовательной замены вхождений нетерминальных знаков из левых частей продукций P на строки из их правых частей,*

$I \to \alpha_1 \to \alpha_2 \to \ldots \to \alpha_{n-1} \to \gamma$, $\gamma \in U(T)$.

Пример 4. Вывод длины 4 в грамматике G из примера 3,

$I \to J \to J1 \to J01 \to 101$. ♦

Для описания семантики одного формального языка (предметного языка) будем задавать отображения его строк в строки другого формального языка (семантического языка). Принцип композиционности в этом случае позволяет упростить определение семантики, т.к. становится возможным использование конечных (конструктивных) средств для сопоставления строк предметного и семантического языков.

Пример 5. Опишем семантику языка двоичных чисел L аксиоматическим методом. Для этого сопоставим продукциям грамматики G из примера 3 формулы формальной арифметики – аксиоматической теории, задаваемой своими аксиомами и правилами вывода:

$I \to 0 \quad \{S_I \leftarrow 0\}$;

$I \to J \quad \{S_I \leftarrow S_J\}$;

$J \to 1 \quad \{S_J \leftarrow 0'\}$;

$J \to J0 \quad \{S_J \leftarrow (S_J) * (0' + 0')\}$;

$J \to J1 \quad \{S_J \leftarrow (S_J) * (0' + 0') + 0'\}$,

где S_I и S_J – формулы формальной арифметики, сопоставленные нетерминальным знакам I и J соответственно, а знак $\leftarrow$ обозначает присваивание строк. Тогда для вывода из примера 4 имеем:

$S_I \leftarrow S_J$, $S_J \leftarrow (S_J) * (0' + 0') + 0'$,

$S_J \leftarrow (S_J) * (0' + 0')$, $S_J \leftarrow 0'$.

Выполнив подстановки строк в порядке, обратном выводу, получим формулу формальной арифметики,

$S_I \leftarrow ((0') * (0' + 0')) * (0' + 0') + 0'$,

интерпретация которой на множестве натуральных чисел дает число 5. ♦

Пример 6. В денотационном методе описания семантики продукциям грамматики сопоставляют-

ся функциональные денотаты (сложные функции). Для грамматики из примера 5 имеем:

$I \to 0 \quad \{S_I \leftarrow 0\};$

$I \to J \quad \{S_I \leftarrow S_J\};$

$J \to 1 \quad \{S_J \leftarrow 1\};$

$J \to J0 \quad \{S_J \leftarrow (\lambda x.gx)S_J\};$

$J \to J1 \quad \{S_J \leftarrow (\lambda x.fgx)S_J\},$

где в качестве семантического языка использовано λ-исчисление Черча, заданное на множестве из двух функций: $gx = x*2$, $fx = x+1$. Для рассмотренного ранее вывода получаем:

$S_I \leftarrow S_J,\ S_J \leftarrow (\lambda x.fgx)S_J,$

$S_J \leftarrow (\lambda x.gx)S_J,\ S_J \leftarrow 1,$

что в итоге дает $S_I \leftarrow (\lambda x.fgx)(\lambda x.gx)1$ с интерпретацией $(1*2)*2+1$, соответствующей числу 5. ♦

Пример 7. В методе операционной семантики продукциям грамматики сопоставляются последовательности команд (операций) некоторого исполнительного устройства (механизма). В этом случае для ранее рассмотренной грамматики и вывода имеем:

$I \to 0 \quad \{S_I \leftarrow mov\ V_I, 0;\};$

$I \to J \quad \{S_I \leftarrow mov\ V_I, V_J;\};$

$J \to 1 \quad \{S_J \leftarrow mov\ V_J, 1;\};$

$J \to J0 \quad \{S_J \leftarrow S_J\ shl\ V_J, 1;\};$

$J \to J1 \quad \{S_J \leftarrow S_J\ shl\ V_J, 1;\ inc\ V_J;\},$

и

$S_I \leftarrow mov\,V_J, 1;\ shl\,V_J, 1;\ inc\,V_J;\ mov\,V_I, V_J;$

где в качестве семантического языка использованы команды процессора: $mov\,A, B$ – пересылка B в ячейку памяти A; $shl\,A, 1$ – сдвиг влево на один двоичный разряд содержимого ячейки A, $inc\,A$ – увеличение содержимого ячейки A на единицу. Интерпретация процессором строки S_I приведет к записи в ячейку памяти V_I числа 5. ♦

Из рассмотренных примеров видны особенности известных методов описания семантики формальных языков:

- семантика формального языка – это его интерпретация с помощью другого языка;
- описание семантики осуществляется конструктивными (формальными) средствами;
- принцип композиционности реализуется путем использования аппарата формальных грамматик;
- фиксируется конечное множество простых семантических категорий, через которые описывается более сложный смысл;
- определяется только одна семантика (задается только одна интерпретация).

СЕМАНТИЧЕСКОЕ ЗАМЫКАНИЕ

Метаязыковый путь описания семантики формальных языков считается неконструктивным, ибо приводит к бесконечной рекурсии: для описания семантики метаязыка требуется другой метаязык, который, в свою очередь, также нуждается в описании. Для решения этой проблемы выполним замыканием метаязыка на его предметный язык, суть которого заключается в том, что, во-первых, множество первичных семантических категорий оставляется открытым и пополняется на этапе решения конкретной прикладной задачи, во-вторых, используется индуктивная грамматическая форма определения более сложного смысла через первичные (базовые) семантические категории, а также через семантические категории, определенные ранее.

Для этого предметный язык будем рассматривать как двойку, состоящую из предметного синтаксиса, или правил построения строк языка, и предметной семантики, или правил интерпретации этих строк с помощью другого языка (рис. 1).

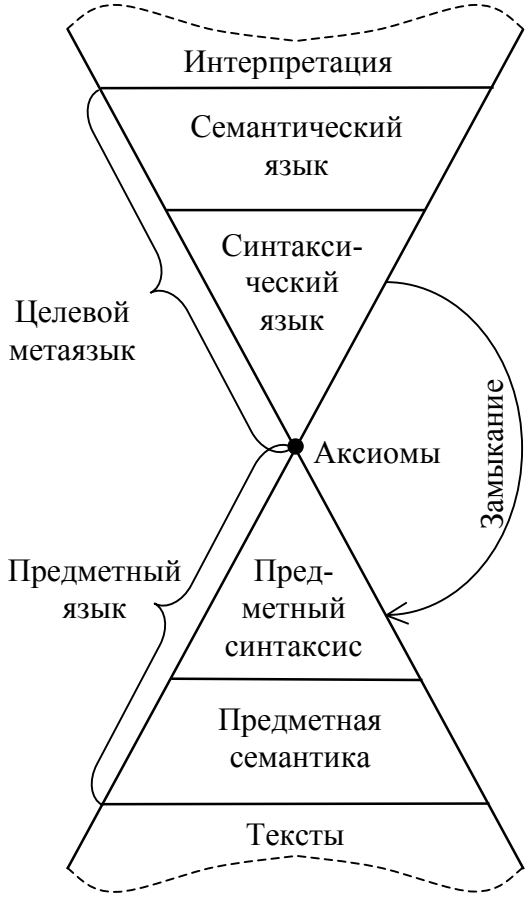

Рис. 1. Семантическое замыкание

Для описания предметного языка воспользуемся метаязыком. В соответствии с определенным выше делением предметного языка, в метаязыке выделим два вида выразительных средств: выразительные средства для описания предметного син-

таксиса (синтаксический язык) и выразительные средства для описания предметной семантики (семантический язык). Связь метаязыка и предметного языка осуществляется через некоторую систему аксиом, например, регламентирующих правила низкоуровнего представления, структурирования и кодирования текстов.

Отождествим составные части метаязыка с соответствующими частями предметного языка, а семантику предметного языка будем определять на самом предметном языке, который в этом случае играет роль своего метаязыка, а то общее, что присуще определению произвольных предметных языков вынесем в протоязык.

В итоге получаем квазиестественный язык, который состоит из аксиомы, универсального протоязыка и некоторого множества проблемных (специализированных) языков (рис. 2).

Рис. 2. квазиестественный zprkое замыкание

В свою очередь один или несколько проблемных языков, определяемых в рамках одного квазиестественного языка, используется не только для определения своей семантики, но служат и для описания некоторой ситуации в предметной области и ее опосредованной интерпретации.

ПРОТОЯЗЫК

Протоязык, или первичный язык, предназначен для описанием синтаксиса проблемных языков и реализует аксиому, которая необходима для объявления базовых семантических категорий при индуктивном описания семантики проблемных языков. Таким образом, протоязык является формальным языком с фиксированными синтаксисом и семантикой.

На рис. 3 приведена формальная грамматика протоязыка, где нетерминальные знаки грамматики обозначены строками над терминальным алфавитом, например, cognition, essences и др., а терминальные знаки заключены в одинарные кавычки, например ′( )′.

cognition → essences [cognition]
essences → ′( )′ notion ′( )′ [intension]
intension → sentence [intension]
sentence → syntax ′{ }′
syntax → item [axiom] [syntax]
item → notion | lexeme
axiom → ′[′ [terminals] ′]′
lexeme → pattern | term
pattern → ′"′ [terminals] ′"′
term → ′'′ [terminals] ′'′
terminals → terminal [terminals]

Рис. 3. Грамматика протоязыка

Грамматика задана с точностью до пробелов и обозначенных курсивом нетерминальных знаков notion, служащих для выражения определяемых нетерминальных знаков проблемных языков. В квадратных скобках указаны части продукций, которые могут быть опущены, а альтернативные правые части правил грамматики разделены вертикальной чертой.

Описание cognition квазиестественного языка состоит из описаний сущностей его предметной области essences. Сущностям присваивается имя notion нетерминального понятия определяемого проблемного языка.

Содержание intension нетерминального понятия notion состоит из предложений sentence, которые служат для выражения синтаксиса syntax предложений, выражающих это понятие в тексте.

Синтаксис предложения syntax выражается последовательностью элементов item: понятий notion и лексем lexeme. Лексема является терминальным понятием определяемого языка и задается ~~в виде~~ последовательностью терминальных знаков terminals. Для выражения лексем используются как терм term, так и множества термов, задаваемых на языке регулярных выражений в виде шаблонов pattern.

После любого элемента item может быть указано применение аксиомы axiom, выражаемое текстом, возможно пустым, заключенным в квадратные скобки. Парные круглые скобки используются для выделения определяемых нетерминальных знаков notion проблемного языка, а фигурные – как разделители предложений sentence.

Пример 8. Опишем на протоязыке синтаксис ранее использованного в примерах языка двоичных чисел (рис. 4). ♦

```
() I ()
        '0' { }
        J { }
() J ()
        '1' { }
        J '0' { }
        J '1' { }
```

Рис. 4. Синтаксис языка двоичных чисел

ПРОБЛЕМНЫЕ ЯЗЫКИ

Проблемный язык – это контекстно-свободный язык, предназначенный для описания своей семантики, предметной области и решаемых в ней задач.

Для описания семантики предложений sentence проблемного языка будем использовать конструкцию semantic, которую определим множеством прагматик pragmatic с именами aspect и текстом text на уже определенном до этого проблемном подъязыке (рис. 5).

```
sentence   → syntax semantic
semantics  → pragmatic [semantic]
pragmatic  → [aspect] '{' [text] '}'
text       → terminals | [aspect] '{' [text] '}'
```

Рис. 5. Правила описания семантики

Таким образом, для каждого предложения имеется возможность описания одной или нескольких именованных семантик. Для указания на ту или иную семантическую интерпретацию некоторого фрагмента текста text используются фигурные скобки, перед которыми указывается имя соответствующей прагматики aspect.

Пример 9. Рассмотрим язык для дифференцирования выражений, включающих целочисленные константы, переменную x и алгебраические операции: изменение знака -, сложение +, вычитание –, умножение *. Свяжем с каждым предложением, выражающим нетерминальный знак Exp, две прагматики: прагматику по умолчанию (без имени) и прагматику с именем dif. Прагматика по умолчанию указывает на то, что выражение не дифференцируется, а именованная прагматика – что выражение необходимо продифференцировать. Таким образом, формальная производная некоторого выражения e – это dif{e}.

Описание семантики языка дифференцирования приведено на рис. 6, где предполагается, что каким-то образом удалось описать семантику предложений (показана многоточием) нетерминального знака Str, который предназначен для представления (первое предложение) и конкатенации (второе предложение) строк.

```
() Str ()
        '".*"' { … }
        Str '&' Str { … }
() Exp ()
        "[0-9]+" [e]
                { e }
                dif { '0' }
        'x' [e]
                { e }
                dif { '1' }
        '(' Exp [e] ')'
                { '(' & e & ')' }
                dif { '(' & dif { e } & ')' }
        '-' Exp [e]
                { '-' & e }
                dif { '-' & dif { e } }
        Exp [e1] '*' Exp [e2]
                { e1 & '*' & e2 }
                dif { '(' & e1 & '*' & dif { e2 } & '+'
                    & dif { e1 } & '*' & e2 & ')' }
        Exp [e1] "+|–" [o] Exp [e2]
                { e1 & o & e2 }
                dif { dif { e1 } & o & dif { e2 } },
```

Рис. 6. Язык дифференцирования

Рассмотрим предложение Exp [e1] '*' Exp [e2]. В этом предложении аксиома используется для задания алиасов (синонимов) нетерминальных знаков, которые необходимы для описания семантики. Неименованная семантика этого предложения тривиальна – если выражение не дифференцируется, то оно остается неизменным, e1 & '*' & e2. Для описания именованной семантики, использовано известное правило дифференцирования произведения функций,

'(' & e1 & '*' & dif {e2} & '+' & dif {e1} & '*' & e2 & ')'.

В итоге, при стандартном описании семантики проблемного подъязыка, соответствующего нетерминальному знаку Str, текст ситуации

dif{-x*x+5*(x–3)}

преобразуется в строку

-(x*1+1*x)+(5*(1-0)+0*(x-3)).

Следует заметить, что порядок перечисления предложений задает их относительный приоритет: предложение, определенное позже (ниже), при грамматическом разборе текста text просматривается раньше. Это позволяет, в частности, задать естественный приоритет используемым в примере алгебраическим операциям и порядок их интерпретации (слева направо или справа налево).

Для тождественных преобразований выражений, получаемых после дифференцирования, возможно определение прагматики с именем equ, в результате применения которой к ситуационному описанию equ{dif{-x*x+5*(x–3)}} получим -2*x+5. ♦

ИНТЕРПРЕТАЦИЯ

Для описания семантики квазиестественного языка было использовано отображение его строк в строки этого же языка. Однако не решенной осталась задача интерпретации квазиестественного языка в объекты реального мира, которая возможна как посредством мыслительного аппарата человека, так и с помощью технического устройства (машины).

В примере 9 использовалась мысленная интерпретация языка дифференцирования, заключающаяся в анализе и преобразовании ситуационного описания в соответствии с заданным определением проблемного языка. Для это потребовалось только знание синтаксиса и семантики протоязыка, а также семантики первых двух предложений. Следует заметить, что их семантика была определена метаязыковыми средствами, для чего использовался естественный язык.

В случае машинной интерпретации языка дифференцирования, в фигурных скобках, где приводится описание семантики первых двух предложений, необходимо записать некоторую последовательность команд (операций), которую машина будет выполнять всякий раз, когда соответствующее предложение будет распознано во входном тексте.

На рис. 7 показан механизм машинной интерпретация строк квазиестественного языка. В этом случае система команд целевой машины реализует множество первичных семантических категорий, а сама интерпретация осуществляется вычислительным механизмом машины, который выполняет последовательности команд, составляющие императивы распознанных в тексте предложений квазиестественного языка.

Для создания таких императивов используется аксиома. Для записи в императив предложения некоторой команды применяются «пустые» квадратные скобки. «Пустые» квадратные скобки, будучи приведенные после некоторого элемента предложения, вызывают запись значения этого элемента в тело императива.

Пример 10. Опишем семантику языка исчисления высказываний для его машинной интерпретации (рис. 8).

Первые два предложения выражают «пустое» понятие и задают выразительные средства для формирования императивов. Первое предложение реализует запись в тело императива байта, задаваемого двумя шестнадцатеричными цифрами.

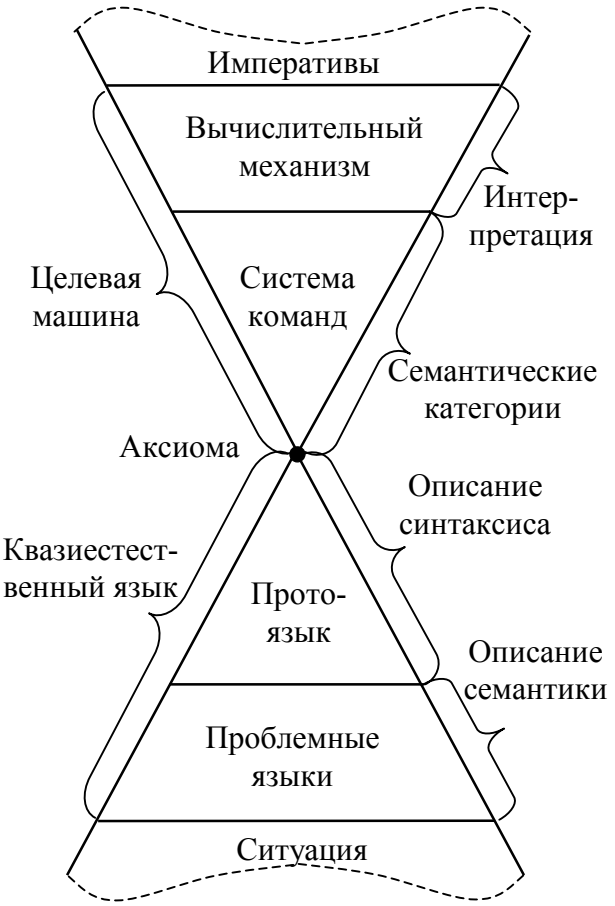

Рис. 7. Машинная интерпретация

```
() ()
        '#' "[0-9A-F][0-9A-F]" [ ] { }
        '<' ".+" [ ] '>' { }
  () Boolean ()
        'false'
                { #B8 #00 #00 #00 #00 #50 }
                asm { <mov eax, 0; push eax;> }
        'true'
                { #B8 #FF #FF #FF #FF #50 }
                asm { <mov eax, -1; push eax;> }
        '(' Boolean ')' { }
        'not' Boolean
                { #58 #F7 #D0 #50 }
                asm { <pop eax; not eax; push eax;> }
        Boolean 'and' Boolean
                { #58 #5A #0B #C2 #50 }
                asm { <pop eax; pop edx;
                    and eax, edx; push eax;> }
        Boolean [a] 'or' Boolean [b]
                { not (not a and not b) }
```

Рис. 8. Язык исчисления высказываний

Второе предложение осуществляет запись в тело императива мнемонического обозначения команды целевой машины или строки целевого языка программирования. В последнем случае подразумевается, что вычислительный механизм перед выполнением императива вызывает интерпретатор

(ассемблер) или целевую систему программирования для получения соответствующей последовательности команд для исполнения.

В примере определено две семантики. Неименованная семантика может исполняться на «голом» процессоре, а семантика с именем asm требует использования вычислительного механизма с предварительным ассемблированием императивов.

Следует обратить внимание на последнее предложение рассматриваемого языка. Для определения семантики этого предложения использованы выразительные средства описанного до этого подъязыка исчисления высказываний. ♦

В итоге получаем, что текст, описывающий семантику предложений квазиестественного языка, преобразуется (компилируется) в императив – некоторую последовательность действий, которую целевая машина выполняет всякий раз, когда при грамматическом разборе входного текста распознается это предложение. По своей сути императивы – единицы вызова целевой машины, в то время как синтаксис предложений – описание структуры таких вызовов.

СЕМАНТИЧЕСКАЯ ИНДУКЦИЯ

Суть предложенного подхода к описании семантики контекстно-свободных языков заключается в использовании метода математической индукции.

Базой индукции в этом случае выступают первичные семантические категории, которые непосредственно реализуются целевой вычислительной платформой и объявляются перед использованием, для чего служит аксиома – единственное необходимое и достаточное правило, включаемое в грамматику всех определяемых языков.

Предположением индукции служат все ранее определенные семантические категории (все ранее определенные нетерминальные понятия языка). Совокупность правил вывода грамматики, выражающая одно и то же нетерминальное понятие, используется для обозначения в тексте семантической категории, связанной с этим понятием.

Индуктивный переход осуществляется в процессе добавления нового правила в грамматику определяемого языка и описания семантики этого правила текстом, выражаемым на уже определенном до этого подъязыке квазиестественного языка.

Заключением индукции является определение новой или уточнение (доопределение) уже существующей семантической категории.

Таким образом, в процессе описания квазиестественного языка создается и семантический язык, необходимый для описания его семантики.

РАЗНЕСЕННЫЙ РАЗБОР

Определение семантики квазиестественного языка осуществлено путем последовательного описания правил вывода формальной грамматики этого языка и индуктивного выражения семантики каждого такого правила через базовые и ранее определенные семантические категории.

Отсюда, в частности, следует, что описание квазиестественного языка и тексты на этом языке должны быть подвергнуты грамматическому разбору таким образом, чтобы обработанные ранее императивы предложений могли быть использованы при грамматическом разборе и компиляции следующих.

Для решения этой задачи применим разработанный метод разнесенного грамматического разбора [23], заключающийся в разделении определения применимости предложения при анализе текста на две части – на контекстное сопоставление предложений, осуществляемое при просмотре текста назад, и структурное распознавание, выполняемое при просмотре вперед. Для этого разделим описание синтаксиса предложений на четыре области: контекст, лексему, область разбора и результат.

Контекстом предложения назовем последовательность понятий, предшествующих первой лексеме. Под лексемой предложения будем понимать его первую лексему. Область разбора определим как часть предложения, которая непосредственно следует за первой лексемой. И, наконец, результат предложения – это нетерминальное понятие языка, выражаемое этим предложением.

Пусть имеется структура данных, которую назовем текущим контекстом и реализуем в виде стека контекста (рис. 9) – запоминающего устройства с дисциплиной доступа «первый вошел, последний вышел». Текущий контекст содержит последовательность понятий, которые уже распознаны, а соответствующий этим понятиям текст извлечен из входного потока. Таким образом, в каждый момент времени контекстный анализатор имеет некоторое текущее состояние, определяемое состоянием стека контекста и текущей позицией во входном потоке.

Поиск предложений, применимых в текущем состоянии анализатора, осуществим путем сравнения контекста предложений с текущим контекстом. Только предложения, имеющие сопоставимый контекст, могут использоваться в текущем состоянии. Здесь под сопоставимостью понимается соответствие нетерминальных понятий на вершине стека контекста понятиям из контекста предложений. Из сопоставимых предложений следует выбрать только применимые, лексема кото-

рых представляется текущим термом из входного потока.

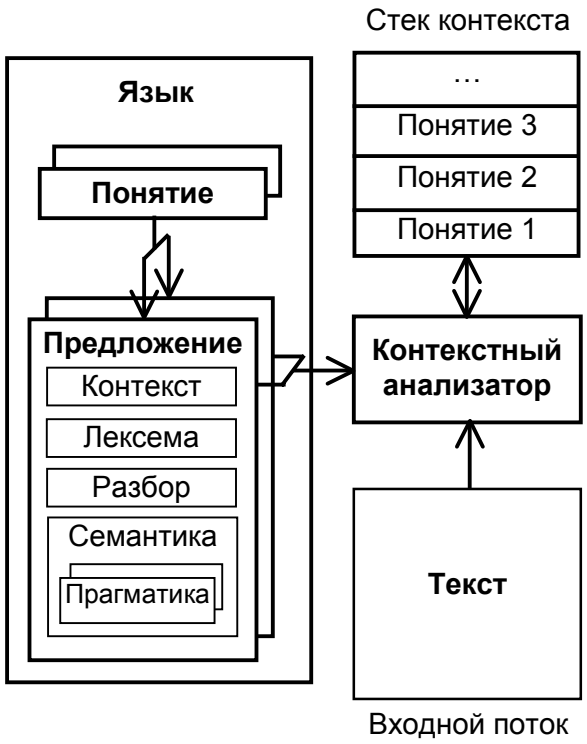

Рис. 9. Механизм разнесенного разбора

После выявления применимого предложения наступает этап разбора оставшейся его части, которая еще не сопоставлена входному потоку. Для этого контекстный анализатор запоминает свое состояние и выполняет просмотр вперед при пустом контексте.

Если элементом разбора является лексема, то выполняется ее сравнение с текущим термом из входного потока. При успешном сравнении терм извлекается из входного потока, а в случае неудачи – состояние анализатора восстанавливается, а анализируемое предложение считается нераспознанным.

Если требуется распознать понятие, то для анализа выбираются только те предложения, которые выражают это понятие. При удачном распознавании анализатор переходит к следующему элементу из области разбора. В противном случае анализируемое предложение считается нераспознанным, состояние анализатора восстанавливается, и он переходит к разбору следующего применимого предложения.

Если все элементы области разбора сопоставлены входному потоку, то предложение считается распознанным, контекст предложения в стеке контекста замещается на его нетерминальное понятие-результат, а полученное состояние входного потока объявляется текущим.

Пример 11. Покажем выполнение разнесенного грамматического разбора на примере языка исчисления высказываний (рис. 8) с анализируемым текстом «true or false and true».

Следует заметить, что при описании языка использовано единственное нетерминальное понятие Boolean, а предложения расположены в порядке естественного приоритета, т.е. константы false и true имеют наибольший приоритет, в то время как логическая связка or имеет наименьший приоритет

При грамматическом разборе текста «true or false and true» с пустым начальным контекстом сопоставимыми являются первые четыре предложения. Однако применимо только второе предложение 'true', т.к. его лексема находится в текущей позиции входного потока. У рассматриваемого предложения область разбора отсутствует, следовательно, это предложение помечается как распознанное, терм true извлекается из входного потока, а в стек контекста заносится понятие Boolean.

Продолжим грамматический разбор оставшейся части входного потока «or false and true» при текущем контексте Boolean. В указанном состоянии сопоставимыми являются последние два предложения, контекст которых сравним с текущим. Однако применимо только последнее предложение Boolean 'or' Boolean, в виду того, что только его лексема может быть выражена термом or из входного потока.

После контекстного сопоставления предложения Boolean 'or' Boolean наступает фаза его структурного распознавания. Теперь из входного потока при пустом контексте необходимо извлечь терм, выражающий нетерминальное понятие Boolean.

Находим, что в текущем состоянии сопоставимым является предложение 'false'. После его распознавания состояние входного потока становится «and true», а в стек контекста заносится искомое понятие Boolean. Однако, в указанном состоянии применимо предложение Boolean 'and' Boolean. Ввиду того, что текущее анализируемое предложение Boolean 'or' Boolean расположено после применимого (т.е. является менее приоритетным), продолжаем грамматический разбор. В процессе распознавания предложения Boolean 'and' Boolean получаем пустой входной поток и понятие Boolean в стеке контекста.

Теперь возвращаемся к анализу предложения Boolean 'or' Boolean. Так как нетерминальное понятие Boolean извлечено из входного потока, завершаем распознавание этого предложения. В итоге имеем пустой входной поток и понятие Boolean в стеке контекста.

Отсюда заключаем, что входной текст «true or false and true» распознан как выражающий понятие Boolean, а при его распознавании были выпол-

нены императивы следующих предложений: 'true', 'false', 'true', Boolean 'and' Boolean и Boolean 'or' Boolean, что соответствует традиционной интерпретации этого текста, при которой связка and имеет больший приоритет, чем or. ♦

Метод разнесенного грамматического разбора позволяет повысить эффективность синтаксического и семантического анализа. Последнее необходимо ввиду возможности обработки текстов, описываемых неоднозначными грамматиками.

Для учета неоднозначности в выборе предложений анализатор перед началом разбора каждого нового предложения сохраняет свое состояние (создает точку отката назад) и начинает грамматический разбор. В каждом новом состоянии анализатор производит поиск применимых предложений. Если таковых предложений не найдено, восстанавливается сохраненное ранее состояние (производится откат назад) и анализируется следующее предложение.

Описанная процедура реализуется рекурсивным вызовом анализатора всякий раз, когда начинается контекстное сопоставление предложений. Учитывая то, что квазиестественный язык создается для описания некоторой предметной области естественным образом, доля неоднозначностей в описании предметной области не должна быть высока. Как бы то ни было, при реализации грамматического разбора может быть предусмотрено некоторое критическое количество точек отката назад. При достижении последнего делается вывод о недостаточной проработке квазиестественного язык или об ошибке в тексте.

ЗАКЛЮЧЕНИЕ

В заключение укажем на работу [24], в которых на основе приведенных в настоящей статье результатов предпринята попытка разработки контекстной технологии программирования, предназначенной для сокращения семантического разрыва между содержательными представлениями относительно предметной области и языками программирования, применяемыми для выражения этих представлений при создании программ.

СПИСОК ЛИТЕРАТУРЫ

1. Кнут Д. Семантика контекстно-свободных языков // В сб.: Семантика языков программирования. М.: Мир, 1980.

2. Specification Case Studies in RAISE / Ed. Van H.D., George C., Janowski T., Moore R. // In Formal Approaches to Computing and Information Technology. Springer, 2002.

3. Wegner P. Vienna definition language // Computer Surveys. 1972. Vol. 4. No 1. PP. 5-63.

4. Wijngaarden V.A., Mailloux B.J., Peck J.E., Koster C.H. Report on the algorithmic language Algol-68. Amsterdam: Mathematisch Centrum, 1969.

5. Hoare C. A. An axiomatic basis for computer programming // Communications ACM. 1969. Vol. 12, No 10. PP. 576-583.

6. Stoy J. E. Denotational semantics: the Scott-Strachey approach to programming language theory. MIT Press. 1977.

7. Linger R., Mills H., Witt B. Structured Programming: Theory and Practice. Reading. MA: Addison-Wesley, 1979.

8. Пратт Т., Зелковиц М. Языки программирования: разработка и реализация. СПб.: Питер, 2002.

9. Замулин А. В. Алгебраическая семантика императивного языка программирования // Программирование. 2003. № 6. С. 51-64.

10. Себеста Р. У. Основные концепции языков программирования. М.: Вильямс, 2001.

11. Floyd R. Assigning meaning to programs // Mathematical Aspects Computer Science. Amer. Math. Soc. 1967. Vol. XIX. PP. 19-32.

12. Вольфенгаген В. Э. Методы и средства вычислений с объектами. Аппликативные вычислительные системы. М.: Центр ЮрИнфоР, 2004.

13. Scott D. S. Lectures on a mathematical theory of computations. Oxford University Computing, 1981.

14. Хантер Р. Основные концепции компиляторов. М.: Вильямс, 2002.

15. Landin P. J. The mechanical evaluation of expressions // Computer Journal. 1964. Vol. 6. PP. 308-320.

16. Cousineau G., Curien P.-L., Mauny M. The categorical abstract machine // Science of Computer Programming. 1987. Vol. 8, No 2. PP. 173-202.

17. Levine R., Mason T., Brown D. LEX & YACC. Sebastopol: O'Reilly & Associates, 1992.

18. Donnelly C., Stallman R. Bison: YACC-compatible parser generator. Boston: Free Software Foundation, 1995.

19. Tarski A. Logic, Semantics, Metamathematics. Oxford, 1956.

20. Tarski A. Das Wahrheitsbegriff in den formalisierten Sprachen // Studia Philosophica. 1935. No 1. PP. 261-405.

21. Смирнова Е. Д. Формализованные языки и проблемы логической семантики. М., 1982.

22. Борщев В. Б. Естественный язык – наивная математика для описания наивной картины мира // Московский лингвистический альманах. 1996. № 1. С. 203-225.

23. Выхованец В. С. Разнесенный грамматический разбор // Проблемы управления. 2006. № 3. С. 32-43.

24. Выхованец В. С., Иосенкин В. Я. Понятийный анализ и контекстная технология программирования // Проблемы управления. 2005. № 4. С. 2-11.

ISBN 0548-0027 НТИ, Сер. 2. Информационные процессы и системы. 2008, № 7. С. 6-14.