

ФОРМАЛИЗАЦИЯ СЕМАНТИКИ ИСКУССТВЕННЫХ ЯЗЫКОВ

В.Я. Иосенкин, В.С. Выхованец

Приднестровский государственный университет

E-mail: tdsu@tirastel.md

Решение любой задачи всегда связано с проникновением в соответствующую предметную область. Предметы и события, лежащие в основе понимания информации, необходимой для решения задачи, составляют предметную область задачи. Предметная область представляется состоящей из абстрактных объектов (сущностей). Сущности предметной области находятся в определенных отношениях друг к другу (ассоциациях), которые также можно рассматривать как сущности и включать в предметную область. Совокупность сущностей, между которыми наблюдаются различные отношения подобия, составляет группу сущностей, являющуюся новой сущностью этой предметной области.

Отношения между сущностями выражаются с помощью суждений (мысленно возможных ситуаций, которые могут иметь или не иметь места для рассматриваемых сущностей). В языке (формальном или естественном) суждениям соответствуют лексические конструкции (предложения). Предложения также можно рассматривать как сущности и включать в предметную область. Совокупность лексических конструкций, характеризующих предметную область, будем называть знаниями об этой предметной области.

Универсальным языком представления знаний является естественный язык. Использование естественного языка в компьютерных системах наталкивается на трудности из-за присущих ему нерегулярностей, двусмысленностей, неоднозначностей, неопределенностей и т.п. Главное препятствие заключается в отсутствии формальной семантики естественного языка, которая имела бы достаточно эффективную операционную поддержку. Как известно, современные технологии программирования реализуются в рамках языков программирования, порождаемых узким классом контекстно-свободных формальных грамматик. Ограничением выразительных качеств этих языков программирования является бесконтекстность правил вывода, что не согласуется с явлением контекстной интерпретации слов в естественных языках и, как следствие этого, уменьшает надежность программирования, затрудняет понимание текста на искусственном языке.

Узким местом современных технологий программирования остается объективная трудоемкость, интеллектуальная и технологическая сложность процесса программирования. Известные технологии объектно-ориентированного программирования реализуются в рамках языков программирования, порождаемых узким классом контекстно-свободных формальных грамматик. Ограничением выразительных качеств известных языков программирования является бесконтекстность правил вывода, что не согласуется с явлением контекстной интерпретации слов в естественных языках и, как следствие, уменьшает надежность программирования.

Контекстный язык близок к естественным языкам и представлен в виде полуконтекстной формальной грамматики $G = \langle T, N, P, I \rangle$, где множество T – терминальный алфавит, N – множество базовых элементов различной природы, (например слов из некоторого ограниченного словаря, деталей конструктора, входящих в состав некоторого набора и т.п.), P – множество синтаксических правил (с их помощью из элементов T образуют синтаксически правильные совокупности) вида $\alpha A \rightarrow \beta$, $\alpha, \beta \in (T \cup N)$, $A \in N$, I – аксиомы [1].

Приведем грамматику языка контекстного программирования Esse [2]. Язык Esse позволяет приблизить процесс проектирования программных средств к парадигмам постановки и решения многих прикладных задач.

```
program ::= '{' knowledge '}' text
knowledge ::= essence | essence '.' knowledge
essence ::= name 'is' visible nation 'that' description
visible ::= 'public' | 'protected' | 'private'
description ::= sentence | sentence ';' description
sentence ::= context words 'makes' nations 'so' definition
context ::= '[' nations ']'
words ::= term | term words | nation local | nation local words
nations ::= nation local | nation local nations
definition ::= token | token definition
local ::= 'as' name
term ::= '" token "' | '" token"' term
```

Программа состоит из декларации знаний о предметной области **knowledge** и описания некоторой ситуации **text**, которая является исполняемой частью программы. Определяемые сущности **essence** образуют понятия **nation** с именами **name** находящиеся в отношении наследования. Каждое понятие определяется лексическими конструкциями **sentence**. В последних задается контекст их применения **context**, последовательность **words** терминальных знаков **term** и понятий **nation**, производящих новый контекст **result** путем выполнения действий **definition**. В **definition** используются ранее описанные лексические конструкции и локальные значения понятий **local** из определяемой конструкции, посредством чего формируются значения нового контекста. Лексемы **token** представляют собой последовательности знаков, разделенных пробельными знаками или знаками пунктуации, которые, в свою очередь, также являются лексемами.

Основными синтаксическими единицами языка является определение словаря <имя> и словарной статьи <слово>. Словари образуют иерархическую структуру, соответствующую иерархии классов при объектно-ориентированном программировании. Поиск определения слова осуществляется по первому имени в текущем контексте, которое задает точку входа в дерево словарей (определяет предметную область). Выбор слова осуществляется по левому контексту, определяемому предыдущими словами-лексемами. После компиляции слова, левый контекст из стека контекста заменяется контекстом, извлекаемым из словаря, что определяет семантику интерпретации следующего слова. Структура словарной статьи: контекст, имя, результат, связь, высокоуровневый код. Контекст фактически определяет последовательность типов-объектов, которые находятся в стеке и при наличии которых может быть применено слово «имя». В результате действия слова вместо объектов контекста в стеке появляются объекты, описываемые полем «результат».

Литература

1. **Выхованец В.С.** Контекстная технология программирования // Труды IV Международной научно-технической конференции по телекоммуникациям (Телеком-99). – Одесса, 1999. – С. 16-119.

2. **Выхованец В.С.** Технология безопасного программирования // Тезисы докладов 8-й Международной конференции «Проблемы управления безопасностью сложных систем». – М., 2000. – Т. 2. – С. 89-91.