

# ПРИНЦИПЫ ПРОЕКТИРОВАНИЯ И ИНТЕРПРЕТАЦИИ ПРОГРАММ, ОРИЕНТИРОВАННЫХ НА ОБРАБОТКУ СЕМАНТИЧЕСКИХ СЕТЕЙ

Шункевич Д. В.

Кафедра интеллектуальных информационных технологий, Белорусский государственный университет информатики и радиоэлектроники  
Минск, Республика Беларусь  
E-mail: shu.dv@tut.by

Данная статья рассматривает основные особенности и преимущества языка программирования, ориентированного на обработку семантических сетей. В статье описана структура программ такого языка, а также кратко рассматриваются некоторые принципы построения интерпретатора такого языка.

## ВВЕДЕНИЕ

В данной статье рассматривается язык программирования, выбранный для реализации модели обработки знаний, описываемой в [1]. Указанный язык SCP (Semantic Code Programming), представляет собой язык процедурного программирования, предназначенный для эффективной интерпретации графодинамических моделей различного вида. Язык SCP относится к классу графовых языков программирования. Особенностью этого языка является то, что не только данные, но и сами программы, написанные на языке SCP, представляются в виде sc-конструкций. Язык SCP является языком параллельного асинхронного программирования. Это необходимо для того, чтобы обеспечить адекватную интерпретацию не только последовательных, но и параллельных, не только синхронных, но и асинхронных формальных моделей.

Язык SCP относится к классу процедурных языков параллельного программирования, ориентированных на переработку нечисловой информации.

Особенностями языка SCP по сравнению с другими языками указанного класса являются:

- приспособленность к переработке нечисловых структур, имеющих мощные выразительные возможности, т.е. обеспечивающих описание любых сложноструктурированных предметных областей;
- использование структурно перестраиваемой графодинамической ассоциативной памяти. Как тексты языка SCP (sc-процедуры), так и перерабатываемые данные являются графовыми конструкциями, хранимыми в графовой структурно перестраиваемой ассоциативной памяти (это означает, что язык SCP относится к классу графовых языков);
- ориентация на переработку непосредственно семантических сетей, а не структур (например, списковых), с помощью которых семантические сети кодируются (это означает, что язык SCP описывает переработку

информации непосредственно на уровне семантического кодирования);

Из всех вышеперечисленных отличительных особенностей языка SCP самой существенной является ориентация на принципиально иную модель памяти - ориентация на использование графодинамической ассоциативной памяти [2]. Более подробно язык SCP рассматривается в работе [3].

## I. СТРУКТУРА ПРОГРАММ, ОРИЕНТИРОВАННЫХ НА ОБРАБОТКУ СЕМАНТИЧЕСКИХ СЕТЕЙ

С теоретико-множественной точки зрения SCP-программа может быть рассмотрена как четверка вида  $\langle V, C, P, O \rangle$ , где

- $V$  – множество переменных sc-программы;
- $C$  – множество констант sc-программы;
- $P$  – множество параметров sc-программы;
- $O$  – множество операторов sc-программы.

Данную структуру можно изобразить в виде следующего фрагмента семантической сети:

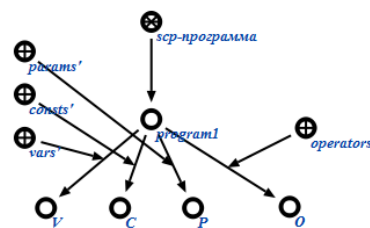


Рис. 1 – Структура sc-программы

**Константы** sc-программы явно участвуют в sc-операторах в качестве элементов и напрямую обрабатываются при интерпретации sc-программы. Константами в рамках sc-программы могут быть узлы любого типа, как sc-константы, так и sc-переменные [2].

**Переменные** sc-программы не обрабатываются явно при интерпретации. Каждая переменная sc-программы может иметь одно значение в каждый момент времени, что указывается связкой соответствующего отношения:

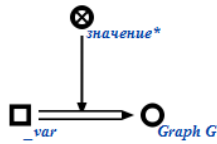


Рис. 2 – Значение переменной в процессе интерпретации scr-программы

Переменной scr-программы может быть только sc-переменная. Значение каждой переменной может меняться в ходе интерпретации scr-программы. При этом интерпретатор при обработке scr-оператора работает непосредственно со значениями переменных, а не самими переменными (которые также являются узлами той же семантической сети). Константа в рамках scr-программы остается неизменной в течение всего срока интерпретации, в связи с чем нет необходимости хранить связку отношения «значение\*» для констант. Константа scr-программы может быть рассмотрена как переменная, значение которой совпадает с самой переменной в каждый момент времени, и изменено быть не может.

Следует отметить, что в рассматриваемой реализации интерпретатора нет необходимости указывать scr-программу, в рамках которой конкретная переменная имеет некоторое значение. Это обусловлено тем, что для интерпретации scr-программы создается ее уникальная копия, в рамках которой задаются значения переменных.

## II. ТИПОЛОГИЯ ПАРАМЕТРОВ SCR-ПРОГРАММ

Среди параметров scr-программ могут быть выделены три типа:

### – *in-параметры*.

Параметры данного типа хоть соответствуют переменным scr-программы, однако не могут менять значение в процессе ее интерпретации. Фиксированное значение переменной устанавливается при создании уникальной копии scr-программы для ее интерпретации, и, таким образом, переменная scr-программы на момент начала ее интерпретации становится константой, обладающей всеми свойствами констант scr-программы. Использование in-параметра можно рассматривать по аналогии с использованием варианта механизма передачи по ссылке в языке C, в котором значение передаваемой переменной не может быть изменено.

### – *out-параметры*.

Параметры данного типа соответствуют переменным scr-программы и обладают теми же свойствами, что и все остальные переменные scr-программы. Чаще всего предполагается, что значение данного параметра необходимо родительской scr-программе, инициировавшей интерпретацию текущей scr-программы. При этом на момент начала интерпретации указанная переменная никогда не имеет значения, которое может

впоследствии появиться, а может быть по прежнему не задано. Использование out-параметра можно рассматривать по аналогии с использованием механизма возврата значений функциями в языке C, но с учетом факта, что количество out-параметров в одной scr-программе может быть произвольным. Случай, когда out-параметры отсутствуют, может быть рассмотрен по аналогии с указанием типа void в качестве типа возвращаемого значения в языке C.

### – *in-out-параметры*.

Параметры данного типа аналогичны out-параметрам за исключением того факта, что соответствующие переменные могут иметь значение переданное из родительской программы уже на момент начала интерпретации текущей программы. Использование out-параметра можно рассматривать по аналогии с использованием варианта механизма передачи по ссылке в языке C, в котором значение передаваемой в функцию переменной может быть изменено, а затем использовано в вызывающей программе. Использование операторов данного типа не приветствуется, поскольку четкое разделение in- и out- параметров позволяет делать in-параметры константами scr-программы, что может существенно ускорить процесс интерпретации scr-программы.

Запись всех типов параметров scr-программы в виде семантической сети можно изобразить следующим образом:

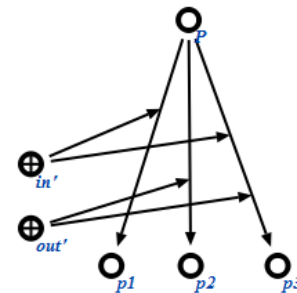


Рис. 3 – Параметры scr-программы

Как видно, тип параметра scr-программы указывается при помощи соответствующих ролевых отношений.

## III. СПИСОК ЛИТЕРАТУРЫ

1. Шункевич, Д.В. Модели и средства компонентного проектирования машин обработки знаний на основе семантических сетей / Д.В. Шункевич; В сб.: Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2013): материалы Междунар. науч.-техн. конф. Минск, 21-23 февраля 2013 г. – Минск: БГУИР, 2013.
2. Голенков, В. В. Представление и обработка знаний в графодинамических ассоциативных машинах / Голенков В. В. [и др.]; под ред. В. В. Голенкова – Минск, БГУИР, 2001. – 412с.
3. Голенков, В. В. Программирование в ассоциативных машинах / Голенков В. В. [и др.]; под ред. В. В. Голенкова – Минск, БГУИР, 2001 – 276 с.