РЕАЛИЗАЦИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ, ОРИЕНТИРОВАННЫХ НА ОБРАБОТКУ СЕМАНТИЧЕСКИХ СЕТЕЙ, НА ТРАДИЦИОННЫХ КОМПЬЮТЕРАХ

Д.Н. Корончик, Д.А. Лазуркин

Белорусский государственный университет информатики и радиоэлектроники, г. Минск, dilaz03@gmail.com , denis.koronchik@gmail.com

Abstract. This paper describes main principles of programming language realization, that used to process semantic networks – SCP (Semantic Code Programming). Also that paper contains some description of associative sc-memory realization. Described language is a part of OSTIS (Open Semantic Technology for Intelligent Systems) project.

В данной статье описан один из возможных подходов к реализации интерпретатора языка процедурного программирования SCP (Semantic Code Programming) [1, 2, 3] на традиционных компьютерах. Язык SCP ориентирован на обработку семантических сетей, кодируемых при помощи SC-кода (Semantic Code) [1].

Программы языка SCP являются семантическими сетями и закодированы в SC-коде [2] (Рисунок 1). Они ориентированы на хранение и интерпретацию в специальной памяти, которая поддерживает ассоциативный доступ – sc-памяти. В то же время на традиционных компьютерах используется линейный, а не ассоциативный доступ к содержимому памяти. Это указывает на необходимость реализации модели sc-памяти на традиционных компьютерах.

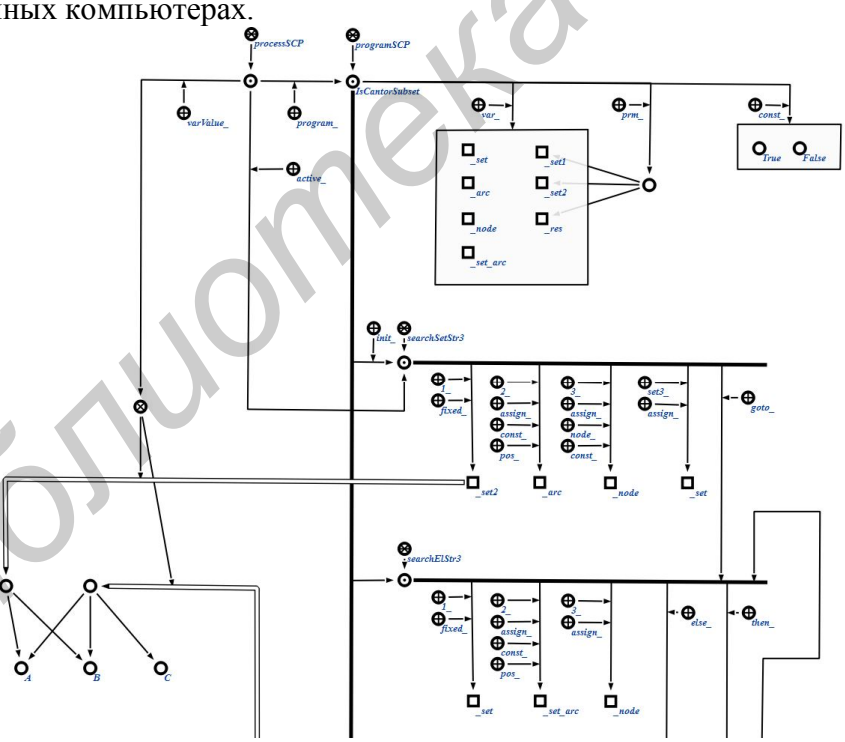

Рисунок 1 - Пример представления фрагмента scp программы в sc-памяти

Важным моментом такой реализации является выбор правильной формы представления sc-конструкций в линейной памяти. Так как sc-конструкция является графом определенного вида, то для ее представления в линейной памяти подходит любая из форм кодирования графов. Однако, на деле, из классических форм кодирования графов подходят только списки смежности и инцидентности, так как

структура sc-памяти часто меняется, поэтому форма кодирования графов должна поддерживать быстрое добавление/удаление элементов и их компактное хранение.

После создания модели sc-памяти следующей задачей, которую нужно решить, является создание формата хранения и транспортировки sc-конструкций, который удобен для программной обработки. Этой цели служит двоичный формат TGF (Transfer Graph Format)[1]. Общая структура формата TGF состоит из заголовка и последовательности команд, каждая из которых формирует часть графа. Основными командами TGF являются:

GENEL – генерировать элемент;

SETBEG – установить начало дуги;

SETEND – установить конец дуги.

Последовательность инструкций в таком формате важна, так как команды получают свои аргументы, не только за счет явного их указания, но и за счет указания номера команды, результат которой необходимо использовать в качестве аргумента.

Бинарный формат хранения sc-конструкций удобен для программной обработки, но не удобен для обработки человеком. Поэтому следующей задачей является разработка транслятора с одной из удобных для человека форм представления scp-программ в бинарный формат. Такими формами могут быть SCs или M4SCP [1], которые в данный момент используются для представления scp-программ. Реализация этого функционала позволяет изменять, хранить и загружать в sc-память scp-программы и обрабатываемые ими sc-конструкции. Следующим шагом является создание интерпретатора scp-программ.

Для реализации интерпретатора scp-программ необходима среда выполнения и планирования процессов и sc-операций обработки sc-памяти. Основную сложность здесь составляет обеспечение эффективности инициирования sc-операций, так как они обладает условием активации, т. е. активируется только при появлении в sc-памяти определенной задачной sc-конструкции. Постоянный поиск таких конструкций является неэффективным, поэтому реализация sc-памяти должна поддерживать подпись обработчиков на различные происходящие в ней события. Такие события могут быть как простыми (генерация sc-элемента, удаление sc-элемента, включение sc-элемента в множество), так и более сложными (генерация в sc-памяти произвольной sc-конструкции). Когда обеспечена среда исполнения, можно реализовывать scp-интерпретатор либо как набор sc-операций либо как итеративный процесс перебора и интерпретации последовательности операторов.

Реализация SCP обеспечивает базовый уровень интерпретации программ на других языках программирования, основанных на SC-коде. Их интерпретаторы можно писать на языке SCP или компилировать их программы непосредственно в SCP. Это позволяет абстрагироваться от того реализована sc-память на традиционных компьютерах, или она реализована на аппаратном уровне.

Литература

1. OSTIS // Open Semantic Technology for Intelligent Systems [Электронный ресурс] – 2010. - Режим доступа: http://ostis.net/ – Дата доступа: 01.04.2010

2. Представление и обработка знаний в графодинамических ассоциативных машинах / В.В. Голенков [и др.]. – Минск, БГУИР, 2001. – 412 с.

3. Программирование в ассоциативных машинах / В.В. Голенков [и др.]. – Минск, БГУИР, 2001 – 276 с.