4. Ядро открытого семейства графовых языков представления знаний, построенных на теоретико-множественной основе

Разделы 2 и 3 были посвящены рассмотрению базового графового теоретико-множественного языка SCB, который является основой для разработки различных языков представления фактографических знаний различного вида. Средства указанного языка являются достаточными для представления любых предметных областей. Расширение семантической мощности языка SCB прежде всего должно быть направлено на обеспечение представления не только фактографических знаний, но и знаний, являющихся описанием логических свойств и закономерностей описываемых предметных областей. Для этой цели вводится язык более высокого уровня SC (Semantic Code), который является ядром открытого семейства графовых языков представления знаний, построенных на теоретико-множественной основе. Рассмотрению указанного языка и посвящен данный раздел.

Данный раздел может быть использован в качестве учебного пособия по дисциплинам «Математические основы искусственного интеллекта» и «Модели представления знаний, базы данных и СУБД» для студентов специальности «Искусственный интеллект».

4.1. Основные понятия, лежащие в основе логических языков

Ключевые понятия: знак множества (= константа), переменная, простая переменная, метапеременная, значение константы, значение переменной, значение метапеременной, область возможных значений переменной, область возможных значений метапеременной, переменная 1-го уровня, переменная 2-го уровня, логическая формула, связанная переменная логической формулы, свободная переменная логической формулы, пропозициональная переменная, высказывание (замкнутая логическая формула, логическая формула без свободных переменных)**, высказывательная форма** (незамкнутая логическая формула, логическая формула со свободными переменными)**, истинное высказывание, ложное высказывание, формальная теория, формальная теория и субъект** (отношение, связывающее формальные теории с соответствующими субъектами)**, предметная область формальной теории, атомарная логическая формула, атомарное высказывание** (фактографический текст)**, фактографический язык, позитивная логическая формула, негативная логическая формула** (отрицательная логическая формула)**, нечеткая логическая формула, неатомарная логическая формула, конъюнктивная логическая формула, дизъюнктивная логическая формула, строгая дизъюнктивная логическая формула** (логическая формула исключающего ИЛИ)**, импликативная логическая формула, логическая формула эквивалентности, кванторная логическая формула, логическая формула о существовании, логическая формула о существовании и единственности, логическая формула о всеобщности, вхождение логической формулы** (в состав другой логической формулы, но не обязательно непосредственно)**, высказывательная форма и релевантное высказывание.**

Переменная – это принципиально отличающийся от **знака** семантически элементарный фрагмент текста. Если знак при теоретико-множественной трактовке семантики текстов обладает свойством обозначать некоторое множество (см. подраздел 2.1), то переменная обладает свойством принимать значения. Таким образом если каждому знаку ставится в соответствие множество, обозначаемое этим знаком, то каждой переменной ставится в соответствие область (множество) ее возможных значений.

Значением переменной является какой-либо знак или какая-либо другая переменная. Образно говоря, значение переменной – это то, что можно вместо нее "подставить" (т.е. то, на что эту переменную можно заменить).

В дальнейшем нам будет удобно распространить свойство принимать значения и для знаков. При этом будем считать, что **область возможных значений** каждого знака представляет собой множество, состоящее из одного элемента, каковым является сам этот знак.

Каждой конкретной **переменной** ставится в соответствие:

1) множество всевозможных изображений этой переменной в различных текстах. Все эти изображения считаются синонимичными и семантически элементарными. Чаще всего синонимичные изображения переменных (т.е. различные изображения одной и той же переменной) в текстах представляются синтаксически подобными (одинаковыми) фрагментами текстов;
2) множество всевозможных значений этой переменной.

В зависимости от типа значений переменные разбиваются на два класса:
- **простые переменные**, значениями которых являются знаки множеств;
- **метапеременные**, значениями которых являются переменные.

Введение метапеременных в язык SC позволяет создавать на его основе логические метаязыки, позволяющие описывать свойства и закономерности самих логических формул и формальных теорий. Для этого, в частности, необходимо логические формулы и формальные теории трактовать как реляционные структуры (см. об этом в подразделе 5.4). Очевидно, что первичными элементами реляционных структур такого рода могут быть не только константные, но и переменные sc-элементы.

Следующим понятием, лежащим в основе логических языков, является понятие **логической формулы**.

Логической формулой будем называть соответствующим образом оформленный текст логического языка либо фрагмент этого текста.

Классификация **логических формул** по признаку наличия свободных переменных имеет следующий вид:
- **высказывание** (замкнутая логическая формула, логическая формула без свободных переменных, повествовательное предложение);
- **высказывательная форма** (незамкнутая логическая формула, логическая формула со свободными переменными).

Классификация **логических формул** по их внутренней структуре имеет следующий вид:
- **атомарная логическая формула** (логическая формула, не содержащая других логических формул);
- **неатомарная логическая формула** (логическая формула, в состав которой входят другие логические формулы);
  - **некванторная логическая формула;**
    - **конъюнктивная логическая формула;**
    - **дизъюнктивная логическая формула;**
    - **строгая дизъюнктивная логическая формула;**
    - **импликативная логическая формула;**
    - **логическая формула эквивалентности;**
    - **отрицательная логическая формула;**
    - **нечеткая логическая формула;**
  - **кванторная логическая формула;**
    - **логическая формула о существовании;**
    - **логическая формула о существовании и единственности;**
    - **логическая формула о всеобщности.**

На основании понятия логической формулы вводится специальный вид переменных – **пропозициональные переменные**, значениями которых являются знаки логических формул.

Логические формулы, относящиеся к классу **высказываний**, обладают истинностным свойством, т.е. являются либо **истинными высказываниями**, либо **ложными высказываниями**. Кроме того, высказывания могут классифицироваться по тому, известно ли в текущий момент значение истинностного свойства. По этому признаку высказывания делятся на **четкие высказывания** (т.е. высказывания, истинность или ложность которых установлена) и на **нечеткие высказывания**.

Специальным видом высказываний можно считать **формальные теории**, каждую из которых можно трактовать как априори истинное (с чьей-то точки зрения) конъюнктивное высказывание, т.е. как множество истинных высказываний, описывающих некоторую предметную область с точки зрения некоторого субъекта.

Особо отметим относительность истинностного свойства высказываний (зависимость от субъекта). Об истинности и ложности, четкости и нечеткости высказываний можно говорить только по отношению к конкретным формальным теориям, отражающим точки зрения различных субъектов. Так, например, одно и то же высказывание в одной формальной теории может быть истинным, а в другой формальной теории – ложным.

Логическую формулу, относящуюся к классу **высказывательных форм**, можно преобразовать в высказывание в результате замены входящих в формулу свободных простых переменных на некоторые константы и входящих в формулу свободных метапеременных на некоторые простые переменные.

4.2. Язык SC (Semantic Code), являющийся основой построения различных логических языков и языков представления знаний

Ключевые понятия: SC, sc-текст, sc-элемент, sc-узел, sc-дуга, sc-константа, sc-переменная, простая sc-переменная, sc-метапеременная, константный sc-узел, константная sc-дуга, переменный sc-узел, переменная sc-дуга.

Язык **SC** (Semantic Code) – это расширение языка SCB (Semantic Code Basic) путем включения в число текстовых элементов не только знаков множеств, но и переменных. Таким образом, элементы, входящие в состав **sc–текстов** (т.е. **sc-элементы**), делятся на следующие классы:

- **sc-константы** (константные sc-элементы; sc-элементы, являющиеся знаками множеств; sc-элементы, каждый из которых имеет одно значение, каковым является сам этот элемент; sc-элементы нулевого уровня; scb-элементы);
- **простые sc-переменные** (sc-элементы, значениями которых являются sc-константы; sc-элементы 1-го уровня; sc-переменные 1-го уровня);
- **sc-метапеременные** (sc-элементы, значениями которых являются sc-переменные; sc-элементы 2-го уровня).

В свою очередь, sc-переменные разбиваются на следующие подклассы:

- переменные, значениями которых являются знаки множеств неуточняемого типа;
- переменные, значениями которых являются знаки пар принадлежности;
- переменные, значениями которых являются знаки узловых множеств;
- метапеременные, значениями которых являются переменные, значениями которых являются знаки множеств неуточняемого типа;
- метапеременные, значениями которых являются переменные, значениями которых являются знаки пар принадлежности;
- метапеременные, значениями которых являются переменные, значениями которых являются знаки узловых множеств.

Язык SC, как и язык SCB, имеет две модификации – язык SCs (Semantic Code string) и язык SCg (Semantic Code graphical).

В языке SC scb-дугу будем называть **константной sc-дугой.**

В языке SC scb -узел будем называть **константным sc-узлом.**

Вводятся также ребра, указывающие на совпадения либо возможные совпадения значения некоторой переменной со значением другой переменной или константы.

Пояснение 4.2.1. Уточним семантику **sc-переменной**. Каждый scb-элемент (каждая sc-константа) является знаком какого-то конкретного множества (правда, об этом множестве может быть не все известно, как в случае scb-узлов неопределенного типа и scb-элементов неопределенного типа). В отличие от этого каждая sc-переменная является знаком не конкретного, а произвольного множества, принадлежащего какому-то дополнительно уточняемому семейству множеств. Знаки множеств, принадлежащих указанному семейству множеств, будем называть **значениями** соответствующей sc-переменной.

Пояснение 4.2.2. Понятие **sc-переменной** (переменного sc-элемента) следует четко отличать от понятия **неопределенного scb-элемента**, который относится к числу sc-констант и является знаком конкретного, но неизвестного в данный момент множества (в частности, унарного предметного множества). Примерами неопределенного sc-элемента являются: знак неизвестного числа, которое требуется вычислить на основании имеющейся о нем информации; знак неизвестного человека (например, преступника), личность которого требуется установить на основании некоторой имеющейся о нем информации. Каждому неопределенному sc-элементу ставится в соответствие область его возможных **синонимов**. Для этого вводится бинарное ориентированное отношение с именем ***"область возможных синонимов"*** и с атрибутами, имеющими идентификаторы

“*неопределенный знак_*” и **“*область возможных синонимов_*”**. Уточнение (вычисление) неопределенного scb-элемента часто осуществляется методом исключения и сводится к сужению области возможных его синонимов. Формальным результатом такого уточнения является склеивание неопределенного scb-элемента с одной из sc-констант, входящей в область возможных синонимов этого неопределенного sc-элемента. После чего неопределенный scb-элемент перестает быть неопределенным.

Особо подчеркнем то, что язык SC обладает такой семантической мощностью, которая позволяет создавать на его основе (в качестве подъязыков) языки представления информации (знаний) самого различного вида. Таким образом, для представления самых различный знаний вполне достаточно средств языка SC. Создание на базе языка SC каждого конкретного языка, обеспечивающего представление знаний того или иного вида, сводится к формированию некоторой системы понятий и соответствующих этим понятиям константных sc-узлов. Такие sc-узлы будем называть **ключевыми узлами** соответствующего подъязыка языка SC.

К числу таких подъязыков языка SC, в частности, относится рассматриваемый в разделе 5 язык SCL (Semantic Code Logic), специально предназначенный для записи логических формул и формальных теорий.

4.3. Язык SCg (Semantic Code graphical) – графическая модификация языка SC

Ключевые понятия: SCg , scg-текст, графический примитив, изображение sc-элемента .

Так как набор элементов языка SC по сравнению с языком SCB значительным образом расширен, то его графическая (нелинейная) модификация требует введения дополнительных графических примитивов для изображения различных типов sc-элементов. В связи с этим в языке SC приняты следующие соглашения:

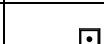
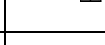
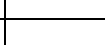
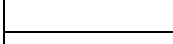
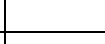
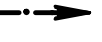
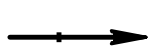
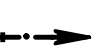
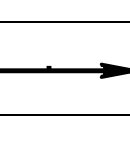
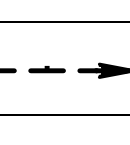
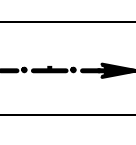
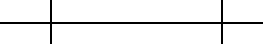
- константные sc-элементы неуточняемого типа и константные sc-узлы изображаются кружочками (маленького и более крупного размера);
- переменные sc-элементы неуточняемого типа и переменные sc-узлы изображаются квадратами, ориентированными по вертикали и горизонтали;
- изображение sc-метапеременных отличается тем, что изображающий их квадрат повернут на 45 градусов;
- изображение sc-элементов неуточняемого типа от изображения sc-узлов отличается уменьшенным размером;
- изображения дуг и ребер, являющихся константами, простыми переменными и метапеременными, отличаются друг от друга тем, что константные дуги и ребра представлены сплошными линиями (того или иного вида), простые переменные – пунктирными линиями, а метапеременные – штрих-пунктирными линиями.

В соответствии с введенными соглашениями в табл. 4.3.1 приведены изображения sc-элементов для графической модификации языка SC. В число графических примитивов языка SC полностью входят графические примитивы языка SCBg, поскольку язык SCBg является подъязыком языка SCg. Алфавит графических примитивов, используемых для изображения основных типов scb-элементов в графическом языке SCBg приведен в табл. 2.3.1. Алфавит дополнительных графических примитивов языка SCBg приведен в табл. 2.4.1.

Таблица 4.3.1. Основные графические примитивы языка SC

Константы	Переменные	Мета-переменные	Пояснения
•	▪	◆	изображение sc-элемента неуточняемого типа
○	□	◇	изображение sc-узла неуточняемого типа
●	■	◆	обозначение предметного множества

Окончание табл. 4.3.1.

Константы	Переменные	Мета-переменные	Пояснения
			обозначение узлового непредметного множества
			обозначение множества знаков пар принадлежности
			обозначение отношения
			обозначение ориентированной связки
			обозначение неориентированной связки
			обозначение атомарной логической формулы
			обозначение простой ориентированной пары с дополнительно уточняемой семантикой
			обозначение пары принадлежности
			обозначение пары непринадлежности
			обозначение пары нечеткой принадлежности
			обозначение пары равенства значений
			обозначение неориентированной (неупорядоченной) пары с дополнительно уточняемой семантикой
			замкнутая линия, являющаяся обозначением множества sc-элементов, изображенных внутри этой линии
			шинная линия, являющаяся способом увеличения контактной зоны sc-узла
			линия, ограничивающая содержимое узла

4.4. Язык SCs (Semantic Code string) – линейная модификация языка SC

Ключевые понятия: SCs, scs-текст, scs-разделитель, scs-ограничитель.

Теперь перейдем к рассмотрению линейной (символьной) модификации языка SC – языка SCs (Semantic Code string). В табл. **4.4.1** перечислены разделители и ограничители этого языка.

В число разделителей и ограничителей языка SCs входят все разделители и ограничители языка SCBs. Кроме этого, к указанному перечню разделителей и ограничителей добавляются новые:

Таблица 4.4.1

Константные разделители		Переменные разделители		Метапеременные разделители		Пояснение
;		разделитель scs-предложений				
,		разделитель идентификаторов sc-элементов в сложных scs-предложениях				
,		разделитель слов в простых идентификаторах sc-элементов				
Begin		признак начала scs-текста, в котором гарантировано отсутствие неявной омонимии				
End;		признак конца scs-текста, в котором гарантировано отсутствие неявной омонимии				
/*	***/**	левый и правый scs-ограничители комментария (в scs-тексте)				
/"	**"/**	левый и правый scs-ограничители содержимого sc-узла				
⟩—	—⟨	scbs-связки инцидентности				
—>	<—	—>>	<<—	—>>>	<<<—	scs-связки принадлежности
-/->	<-/-	-/->>	<<-/-	-/->>>	<<<-/-	scs-связки непринадлежности
~>	<~	~>>	<<~	~>>>	<<<~	scs-связки нечеткой принадлежности
{	**}**	**{·**	**·}**	**{··**	**··}**	scs-ограничители сложного идентификатора неориентированного множества
❮	**❯**	**❮·**	**·❯**	**❮··**	**··❯**	scs-ограничители сложного идентификатора кортежа
[	**]**	**[·**	**·]**	**[··**	**··]**	scs-ограничители сложного идентификатора системы множеств
:		**::**		**:::**		разделители атрибута и компонента кортежа
/:	**:/**	**/::**	**::/**	**/:::**	**:::/**	scs-ограничители, неявно задающие дугу, выходящую из элемента, указанного внутри данных ограничителей, и входящего в sc-элемент, записанный непосредственно справа от правого ограничителя
(	**)**	ограничители обозначения неидентифицируемой связки, ограничители аргументов функций				
=>	<=	=>>	<<=	=>>>	<<<=	связка, соответствующая ориентированной паре неуточняемого вида
<=>		<<=>>		<<<=>>>		связка, соответствующая неориентированной паре неуточняемого вида
=		==		===		связка равенства значений

≠	≠≠	≠≠≠	связка неравенства значений

Продолжение табл. 4.4.1

Константные разделители		Переменные разделители		Метапеременные разделители		Пояснение
≈		≈≈		≈≈≈		связка нечеткого равенства значений
scs-связки бинарных отношений над множествами:						
⩭		⩭		⩭		обозначение пары равенства множеств
⩮		⩮		⩮		обозначение пары неравенства множеств
⧦		⧦		⧦		обозначение пары эквивалентности множеств по набору элементов
⧧		⧧		⧧		обозначение пары неэквивалентности множеств по набору элементов
⧈		⧈		⧈		обозначение пары пересекающихся множеств
⧅		⧅		⧅		обозначение пары непересекающихся множеств
⊂	⊃	⊂	⊃	⊂	⊃	связка строгого включения множеств
⊄	⊅	⊄	⊅	⊄	⊅	связка строгого невключения множеств
⊆	⊇	⊆	⊇	⊆	⊇	связка включения множеств
⊈	⊉	⊈	⊉	⊈	⊉	связка невключения множеств
scs-связки бинарных отношений над числами:						
<	>	<	>	<	>	связка строгого сравнения чисел
≤	≥	≤	≥	≤	≥	связка нестрогого сравнения чисел
имена унарных функций над числами:						

—	обозначение операции арифметического отрицания
abs	обозначение функции взятия абсолютного значения
!	обозначение функции факториала
exp	обозначение показательной функции
ln	обозначение функции взятия натурального логарифма
sin ***cos*** ***tg*** ***ctg***	обозначения тригонометрических функций

Окончание табл. 4.4.1

Константные разделители	Переменные разделители	Метапеременные разделители	Пояснение
scs-связки бинарных функций над множествами:			
∪	∪	∪	связка объединения множеств
∀	∀	∀	связка соединения множеств
∩	∩	∩	связка пересечения множеств
\	\	\	связка разности множеств
△	△	△	связка симметрической разности множеств
×	×	×	связка декартова произведения
scs-связки бинарных функций над числами:			
+	+	+	связка арифметического сложения
•	•	•	связка арифметического умножения
–	–	–	связка арифметического вычитания
/	/	/	связка арифметического деления
↑	↑	↑	связка возведения в степень
√	√	√	связка взятия корня
имена бинарных функций над числами:			
log	обозначение функции взятия логарифма		

4.5. Ключевые узлы графового языка SC

Ключевые понятия: ключевой узел графового языка, ключевой узел языка SC, ключевой узел sc-подъязыка.

Каждому линейному языку ставится в соответствие некоторое множество ключевых слов, знание смысла которых обеспечивает расшифровку произвольных текстов этого языка. К числу ключевых слов, в частности, относятся слова-разделители, слова-ограничители, слова-кванторы, слова-связки.

Аналогом ключевых слов в графовых языках являются **ключевые узлы**. В отличие от ключевого слова каждый ключевой узел имеет не более чем однократное вхождение в каждую языковую конструкцию.

Главным достоинством языка SC является то, что он представляет собой удобную основу для целого семейства языков, имеющих самое различное назначение и легко интегрируемых друг с другом. При этом построение каждого конкретного графового языка, являющегося подъязыком базового графового языка SC, в конечном счете сводится к формированию набора ключевых узлов, обозначающих основные понятия (в том числе и метапонятия), используемые в создаваемом языке. Некоторые

ключевые узлы, используемые во многих таких графовых языках, условно отнесем к ключевым узлам языка SC. Список этих ключевых узлов может быть легко расширен.

Во множестве ключевых узлов языка SC можно выделить, в частности, следующие группы узлов:

1) ключевые узлы, используемые для описания теоретико-множественных соотношений;
2) ключевые узлы, используемые для задания метаотношений, типологии отношений;
3) ключевые узлы, используемые для задания реляционной структуры, т.е. описания соотношения между исходной описываемой реляционной структурой и той однородной информационной конструкцией (т.е. константной sc-конструкцией с позитивными дугами), которая представляет указанную исходную реляционную структуру;
4) ключевые узлы, используемые для описания соотношений между реляционными структурами и для описания топологических свойств реляционных структур;
5) ключевые узлы, используемые для описания числовых соотношений;
6) ключевые узлы, используемые для описания всевозможных свойств (в том числе измеряемых свойств, таких как мощность множеств, точность неточных чисел, чёткость нечётких sc-дуг);
7) ключевые узлы, используемые для описания пространственных соотношений;
8) ключевые узлы, используемые для описания соотношений во времени (темпоральных соотношений);
9) ключевые узлы, используемые для описания отношений, заданных на множестве линейных информационных конструкций;
10) ключевые узлы, используемые для описания содержимого sc-узлов и идентификаторов sc-элементов;
11) ключевые узлы, используемые для записи команд просмотра и редактирования sc-конструкций, хранимых в графодинамической памяти.

Рассмотрим каждую группу ключевых узлов языка SC.

1-я группа ключевых узлов языка SC связана с понятием множества. Это понятие, как было отмечено в подразделе 2.1, является фундаментом денотационной семантики языка SC. Каждое множество задается:

- знаком (оболочкой) этого множества;
- набором элементов этого множества;
- подмножеством отношения принадлежности, связывающим знак этого множества со всеми (!) его элементами.

При этом не все элементы представляемого множества могут присутствовать в текущем состоянии перерабатываемой sc-конструкции и не все элементы представляемого множества, присутствующие в текущем состоянии sc-конструкции, могут быть явно указаны как элементы представляемого множества.

Типология множеств и соответствующие ключевые узлы рассмотрены в подразделе 3.1. Например, ключевой узел с scb-идентификатором ***"множество"*** обозначает универсальное нормализованное множество, т.е. множество, по отношению к которому все остальные нормализованные множества являются подмножествами.

К рассматриваемой группе ключевых узлов языка SC также относятся константные sc-узлы со следующими идентификаторами: ***пара принадлежности, узловое непредметное множество, кортеж, включение множества, строгое включение множества, равенство множеств, эквивалентность множеств по совпадению элементов, пересекающиеся множества, Отношение принадлежности, Отношение непринадлежности, невключение множества, не быть строгим включением множества, неравенство множеств, не быть множествами с одинаковыми элементами, непересекающиеся множества, объединение множеств, соединение множеств, разбиение множества, пересечение множеств, разность множеств, симметрическая разность множеств.***

Семантика этих ключевых узлов рассмотрена в подразделах 2.7 и 3.3.

2-я группа ключевых узлов языка SC связана с понятием отношения, т.е. содержит метаотношения и типологию отношений.

Ко 2-й группе ключевых узлов языка SC относятся константные sc-узлы со следующими идентификаторами: ***отношение , схема отношения , отношение_ , схема отношения_ , область определения , домен , проекция , функция , минимальное декартово произведение, дополнение до декартова произведения, соединение отношений , произведение бинарных отношений, бинарное отношение, бинарное ориентированное отношение, классическое бинарное отношение, бинарное неориентированное отношение, бинарное отношение без функций, бинарное отношение с одной функцией, взаимно однозначное бинарное отношение, бинарное отношение с двумя функциями, рефлексивное бинарное отношение, иррефлексивное бинарное отношение, частично рефлексивное бинарное отношение, симметричное бинарное отношение, антисимметричное бинарное отношение, частично симметричное бинарное отношение, транзитивное бинарное отношение, антитранзитивное бинарное отношение, частично транзитивное бинарное отношение, отношение эквивалентности , отношение предпорядка, отношение частичного порядка, отношение линейного порядка.***

Семантика этих ключевых узлов рассмотрена в подразделе 3.3.

К 3-й группе относятся константные sc-узлы со следующими идентификаторами: ***реляционная структура , первичный элемент_ , сигнатурный атрибут_ , сигнатурное отношение_ , сигнатурное множество_ , функция реляционной структуры, алгебраическая операция реляционной структуры .***

Семантика этих ключевых узлов рассмотрена в пункте 3.4.1.

Каждая конкретная реляционная структура, как уже отмечалось, в языке SC задается кортежем метаотношения ***реляционная структура*** . Типология реляционных структур определяется теоретико-графовыми ("топологическими") их свойствами (в частности, можно говорить о связных и несвязных реляционных структурах), а также алгебраическими свойствами (так, например, можно говорить о реляционных структурах, являющихся алгебраическими группами). Можно говорить о различных фрагментах реляционной структуры, удовлетворяющих тем или иным требованиям. Так, например, можно говорить о минимальном (в том или ином смысле) пути между заданными первичными элементами реляционной структуры. Можно говорить о различных соотношениях между реляционными конструкциями, например, о различных морфизмах (см. пункт 3.4.4.).

К 4-й группе ключевых узлов языка SC относятся константные узлы со следующими идентификаторами: ***гомоморфизм , изоморфизм , автоморфизм.***

Семантика этих ключевых узлов была описана в пункте3.4.

К 5-й группе ключевых узлов языка SC относятся константные sc-узлы со следующими идентификаторами: ***порядок чисел , меньше или равно_ , больше или равно_ , строгий порядок чисел , меньше_ , больше_ , арифметическое отрицание , числовой модуль , модуль_ , целая часть числа , целое_ , сложение , сумма_ , слагаемое_ , умножение , произведение_ , сомножитель_ , sin , sin_ , cos , cos_ , tg , tg_ , ctg , ctg_ , arc_ , степень(=корень-логарифм-степень), корень_ , логарифм_ , степень_ , факториал, факториал_ , основание_.***

Также ключевыми узлами для десятичной системы счисления являются: знаки натуральных чисел в интервале от *0* до *9* включительно (десятичные цифры) и знак числа *10* (основание десятичной системы счисления). Десятичное представление числа в языке SCB можно также трактовать как кортеж специального отношения ***"десятичное представление чисел"***, заданного на множестве десятичных цифр (т.е. натуральных чисел от *0* до *9*) и использующего атрибуты, задающие номер соответствующего разряда в десятичном представлении. Учитывая то, что десятичное представление чисел легко и однозначно изображается в виде цепочки символов, это представление можно изображать в виде содержимого scb-узла. При этом scb-узел, обозначающий число, и scb-узел, содержимое которого является строковым изображением десятичного представления этого числа, связаны отношением ***"строковая запись десятичного представления".***

Семантика этих ключевых узлов рассмотрена в пункте 3.3.14.

6-я группа ключевых узлов языка SC связана с понятием измеряемого параметра (свойства, признака, характеристики), в основе которого лежит идея описания всевозможных объектов как точек в n-мерном пространстве возможных значений различных параметров.

Каждому измеряемому параметру ставится в соответствие некоторое (чаще всего упорядоченное) множество, которое называется шкалой значений этого параметра. Значение некоторого параметра для некоторого конкретного объекта в языке SC задается проведением позитивной константной sc-дуги из sc-узла, представляющего элемент шкалы, в sc-узел, обозначающий класс всех объектов, имеющих соответствующее значение измеряемого параметра.

Существуют самые различные шкалы, соответствующие различным параметрам: числовые, нечисловые, непрерывные и дискретные. Простейшим примером такого многообразия шкал являются различные единицы измерения. Кроме этого можно противопоставлять абсолютные (реальные) шкалы и условные числовые шкалы, являющиеся способом формализации таких понятий, как "мало", "очень мало", "очень-очень мало", "много", "очень много" и т.д.

К 6-й группе ключевых узлов языка SC, в частности, относятся константные sc-узлы со следующими идентификаторами: ***шкала измерения***, ***номер***, ***мощность множества*** (=***pwSet***), ***количество элементов*** (=***pwEl***), ***вес пары принадлежности*** (=***pwArc***), ***арность отношения***, ***absScale***, ***convScale***, ***семейство множеств одинаковой мощности***, ***килограмм***, ***грамм***, ***километр***, ***метр***, ***секунда***.

Семантика этих ключевых узлов рассмотрена в подразделе 6.1.

Ключевые узлы языка SC, относящиеся к 7-й группе ключевых узлов, обеспечивают описание всевозможных пространственных соотношений между объектами, таких как целое–часть, пространственная смежность, пространственное объединение, пространственное разбиение, пространственное пересечение.

Ключевые узлы языка SC, относящиеся к 8-й группе ключевых узлов, обеспечивают описание всевозможных темпоральных (временных) соотношений между процессами, таких как "быть этапом данного процесса", "начаться одновременно", "кончиться одновременно", "происходить во время выполнения данного процесса" (т.е. начаться не раньше и кончиться не позже), "начаться раньше", "кончиться позже", "происходить раньше" (в целом), "пересекаться во времени" (иметь одновременно выполняемые этапы), "непосредственно следовать за", "быть разбиением процесса на непересекающиеся во времени смежные этапы" и т.д.

Семантика темпоральных соотношений и соответствующих им ключевых узлов рассмотрена в пункте 6.2.

К 9-й группе ключевых узлов языка SC относятся константные узлы со следующими идентификаторами: ***string***, ***eqString***, ***comprString***, ***min_***, ***max_***, ***addString***, ***symbol***.

Ключевой узел ***string*** является знаком множества знаков всевозможных строк символов (линейных текстов). Каждая такая строка в языке SC обозначается соответствующим константным sc-узлом.

Ключевой узел ***eqString*** является знаком симметричного отношения нефиксированной арности, каждая связка которого обозначает множество всех одинаковых символьных информационных конструкций. Например, запись вида: ***eqString*** —≻ {***s***, ***t***, ***u***}; означает, что sc-узлы ***s***, ***t***, ***u*** обозначают одинаковые строковые конструкции.

Ключевой узел ***comprString*** является знаком асимметричного отношения нефиксированной арности, выделяющего из множества строк минимальную и максимальную. Атрибутами отношения являются ключевые узлы ***min_*** и ***max_***.

Ключевой узел ***min_*** является знаком атрибута, используемого в некоторых отношениях, указывающего на минимальный элемент. В частности, в рамках отношения ***comprString*** он указывает на минимальную строку.

Ключевой узел ***max_*** является знаком атрибута, используемого в некоторых отношениях, указывающего на максимальный элемент. В частности, в рамках отношения ***comprString*** он указывает на максимальную строку.

Ключевой узел ***addString*** является знаком асимметричного отношения нефиксированной арности соединения (конкатенации) строк, использующего атрибут ***sum_*** для указания результата конкатенации и числовые атрибуты ***1_, 2_, ...*** для указания порядка, в котором располагаются соединяемые строки. Например, запись вида ***addString*** —≻ **{ *1_* : *s* , *2_* : *t* , *sum_* : *u* };** означает, что строка, обозначенная узлом ***u***, является результатом соединения строк ***s*** и ***t***.

Ключевой узел ***symbol*** является знаком унарного отношения ”быть символьной информационной конструкцией, состоящей из одного символа”.

10-я группа ключевых узлов языка SC связана с понятием содержимого и с понятием идентификатора sc-элемента. Идентификатор sc-элемента и содержимое sc-элемента – это информационные конструкции, которые ставятся в соответствие указанному sc-элементу. Идентификаторы sc-элементов и содержимое sc-элементов вместе с текущим состоянием перерабатываемой sc-конструкции хранятся в памяти абстрактной sc-машины и используются при реализации ряда операций. Идентификаторы могут иметь sc-элементы любого вида (как узлы и дуги, так и элементы неопределенного типа, как константы, так и переменные). Разные sc-элементы обязаны иметь разные идентификаторы. Каждый идентификатор представляет собой некоторую строку символов.

Далеко не все хранимые в памяти sc-элементы должны иметь идентификаторы. Это требование предъявляется к ключевым узлам и ко всем хранимым sc-элементам, которые могут упоминаться при вводе новых sc-конструкций в память абстрактной sc-машины.

Содержимым могут обладать только константные предметные sc-узлы. При этом разные константные sc-узлы в принципе могут иметь одинаковое содержимое. Содержимое представляет собой информационную конструкцию произвольного вида. В частности, содержимым может быть любая символьная информационная конструкция, представление какого-либо числа в той или иной системе счисления. Далеко не все хранимые в памяти константные sc-узлы могут иметь содержимое. И далеко не для всех константных sc-узлов, у которых содержимое в принципе существует, это содержимое реально присутствует в текущем состоянии памяти, т.е. является вычисленным (сформированным) в текущий момент. Кроме того, содержимое sc-узла, независимо от того, сформировано оно или нет, может быть стационарным (фиксированным, не меняющимся в процессе переработки информации) и нестационарным (нефиксированным, меняющимся в процессе переработки информации). Узел с нестационарным содержимым – это аналог адресуемой области памяти традиционного компьютера, которая в процессе переработки информации меняет свое состояние.

К 10-й группе ключевых узлов языка SC относятся константные узлы со следующими идентификаторами: ***formIdtf, existCont, formCont, fixCont, comprCont, ComprEqCont, EqCont, пояснение, текст_, wordDef, wordSem, word_, wordIncl.***

Ключевой узел ***formIdtf*** является знаком унарного отношения “быть sc-элементом, у которого в текущий момент времени имеется идентификатор”. Проведение в некоторый sc-элемент константной негативной sc-дуги из узла ***formIdtf*** означает то, что у указанного sc-элемента идентификатор в текущий момент времени отсутствует.

Ключевой узел ***existCont*** является знаком унарного отношения ”быть константным sc-узлом, у которого существует содержимое, но не обязательно в текущий момент времени”. Проведение в некоторый sc-узел константной негативной sc-дуги из узла ***existCont*** означает то, что у указанного sc-узла содержимого в принципе быть не может.

Ключевой узел ***formCont*** является знаком унарного отношения ”быть константным sc-узлом, у которого в текущий момент времени имеется содержимое”. Проведение в некоторый sc-узел константной негативной sc-дуги из узла ***formCont*** означает то, что у указанного sc-узла содержимое в текущий момент времени отсутствует.

Ключевой узел ***fixCont*** является знаком унарного отношения ”быть константным sc-узлом, содержимое которого является фиксированным и при этом не обязательно сформировано в текущий момент времени”. Проведение в некоторый sc-узел, относящийся к классу ***existCont,*** константной

негативной sc-дуги из узла ***fixCont*** означает то, что у указанного sc-узла содержимое является нефиксированным, т.е. меняющимся в процессе переработки информации.

Ключевой узел ***EqCont*** является знаком симметричного отношения нефиксированной арности, каждая связка которого обозначает множество sc-узлов, имеющих равные содержимые.

Ключевой узел ***пояснение*** является знаком бинарного асимметричного отношения, связывающего множество sc-элементов любого вида с константным sc-узлом, содержимым которого является текст, представляющий собой пояснение указанного множества. Ключевой узел ***текст_*** является атрибутом донного отношения.

Ключевой узел ***wordDef*** является знаком бинарного асимметричного отношения, связывающего множество sc-элементов любого вида с константным sc-узлом, содержимым которого является текст, представляющий собой определение понятия, обозначаемого указанным sc-элементом. Ключевой узел ***текст_*** является атрибутом донного отношения. Семантика этих ключевых узлов более подробно рассмотрена в пункте 6.6.

Ключевой узел ***text_*** является знаком атрибута некоторых отношений, указывающего на константный sc-узел, содержимым которого является некоторый текст. Семантика этих ключевых узлов более подробно рассмотрена в пункте 6.6.

Ключевой узел ***wordSem*** является знаком бинарного асимметричного отношения, связывающего множество sc-элементов любого вида с константным sc-узлом, имеющим атрибут ***word_,*** содержимым которого является текст, трактуемый как дополнительный (вспомогательный, альтернативный) идентификатор вышеуказанного sc-элемента.

Ключевой узел ***word_*** является знаком атрибута отношения ***wordSem,*** указывающим на константный sc-узел, содержимым которого является слово или словосочетание, определяющее семантику некоторого понятия.

Ключевой узел ***wordIncl*** является знаком асимметричного отношения нефиксированной арности, которое описывает связь между вхождениями каких-либо терминов (термина) в какой-либо текст. Последним может быть раздел данного текста, либо библиографическая ссылка.

Семантика ключевых узлов, относящихся к 10-й группе также рассмотрена в пункте 6.6.

Ключевые узлы языка SC, относящиеся к 11-й группе ключевых узлов, обеспечивают запись:

- команд ассоциативного поиска и отображения sc-конструкций, соответствующих заданному образцу, который может иметь произвольный размер и произвольную конфигурацию;
- команд отображения содержимого указываемого sc-узла;
- команд удаления указываемого sc-элемента;
- команд ассоциативного поиска sc-конструкций, соответствующих заданному образцу, с последующим удалением тех sc-элементов, которые соответствуют указываемым элементам образца;
- команд ассоциативного поиска sc-конструкций, соответствующих другому заданному образцу (см. раздел 7).

4.6. Понятие sc-подъязыка. Семейство графовых языков, построенных на базе языка SC

Ключевые понятия: sc-подъязыки, семейство абстрактных sc-машин (информационных машин, внутренними языками которых являются различные sc-подъязыки)**, семейство sc-моделей** (формальных моделей, реализуемых на различных абстрактных sc-машинах)**.**

Понятие семейства sc-подъязыков, т.е. языков, являющихся подъязыками графового языка SC (см. определение 4.6.1), понятие семейства абстрактных sc-машин, т.е. абстрактных информационных машин, внутренними языками которых являются различные sc-подъязыки (см. определение 4.7.1), а также понятие семейства sc-моделей, т.е. формальных моделей, реализуемых на различных абстрактных sc-машинах (см. определение 4.8.1), являются ключевыми понятиями всей данной работы. Это обусловлено тем, что:

- sc-подъязыки являются удобным средством семантического представления текстов самых различных языков (как символьных, так и графовых);
- абстрактные sc-машины являются удобным средством рассмотрения на семантическом уровне самых различных способов организации переработки информации;
- sc-модели являются удобным средством семантического представления самых различных формальных моделей и удобным средством приведения различных формальных моделей к общему виду.

Поэтому все рассматриваемые в данной работе графовые языки относятся к классу sc-подъязыков, все рассматриваемые абстрактные машины – к классу абстрактных sc-машин, а формальные модели – к классу sc-моделей.

Важнейшим достоинством семейства всевозможных sc-подъязыков, семейства абстрактных sc-машин и семейства формальных sc-моделей является то, что любой язык, любую абстрактную машину и любую формальную модель можно достаточно легко представить в виде эквивалентного sc-подъязыка, эквивалентной абстрактной sc-машины, эквивалентной формальной sc-модели. Такая уникальная возможность приведения к общему виду самых различных языков, абстрактных машин и формальных моделей создает все необходимые предпосылки для эффективной интеграции любых языков, абстрактных машин и формальных моделей, так как интегрировать различные подъязыки языка SC, различные абстрактные sc-машины, различные формальные sc-модели существенно проще, чем интегрировать произвольные языки, произвольные абстрактные машины и произвольные формальные модели.

Итак, sc-подъязыки являются удобным средством уточнения денотационной семантики самых различных (в первую очередь графовых) языков, а абстрактные sc-машины являются удобным средством уточнения операционной семантики всевозможных (и в первую очередь графовых) языков.

Введенные нами понятия sc-конструкции, sc-подъязыка, абстрактной sc-машины и формальной sc-модели дают возможность уточнить такие понятия, как:
- эквивалентность информационных конструкций, языков, абстрактных машин и формальных моделей;
- интеграция информационных конструкций, языков, абстрактных машин и формальных моделей;
- интерпретация абстрактных машин и формальных моделей.

Графовый язык SC рассматривается нами как ядро целого семейства графовых языков, являющихся подъязыками языка SC и называемых в соответствии с этим sc-подъязыками.

Будем называть **sc-подъязыком** такой графовый язык, каждая конструкция которого является sc-конструкцией.

Благодаря тому, что язык SC обладает неограниченной семантической мощностью, семейство графовых языков, создаваемых на его базе, заполняет абсолютно весь семантический спектр языков, т.е. в число sc-подъязыков входят и sc-подъязыки, являющиеся языками программирования (т.е. языками представления программ) и sc-подъязыки, являющиеся языками представления данных, перерабатываемых программами, и sc-подъязыки, являющиеся языками представления фактографических высказываний, и sc-подъязыки, являющиеся языками представления логических высказываний (т.е. знаний о свойствах и закономерностях предметной области). Примеры таких языков см. в разделах 5, 6.

Синтаксис и семантика графовых языков опираются на понятие ключевого узла. У графовых языков, конструкции которых имеют метки (унарные отношения, заданные на множестве элементов конструкции), ключевую роль могут иметь также и метки. У языка SC набор меток фиксирован, поэтому основную роль в синтаксисе и семантике языков, построенных на базе языка SC, выполняет определенный набор ключевых узлов языка SC, знание семантики которых обеспечивает определение смысла (прочтение, расшифровку, понимание) любой конструкции такого языка. Следовательно, определение денотационной семантики ключевых узлов графового языка, сделанное на каком-либо метаязыке (в качестве которого может быть использован и естественный язык), составляет основу описания денотационной семантики каждого графового языка. При этом обратим внимание на то, что среди ключевых узлов графового языка отсутствуют аналоги разделительных и ограничительных ключевых лексем символьных языков. Разделителям и ограничителям трудно поставить в соответствие какую-либо денотационную семантику, поскольку они используются только как средство структуризации линейных символьных конструкций.

Итак, каждый конкретный sc-подъязык имеет свои семантические и синтаксические особенности в дополнение к базовым семантическим и синтаксическим особенностям самого языка SC (см. подразделы 4.2 – 4.4). При этом дополнительные (к языку SC) семантические особенности каждого конкретного sc-подъязыка в полной мере могут быть описаны путем описания денотационной семантики всех ключевых узлов этого sc-подъязыка, точнее, ключевых узлов, вводимых в дополнение к ключевым узлам базового языка SC. Подчеркнем, что все ключевые узлы языка SC входят в число ключевых узлов каждого графового языка, построенного на его базе. Таким образом, создание на базе языка SC каждого конкретного sc-подъязыка, по существу, сводится к определению набора всех ключевых узлов создаваемого языка.

Перейдем к рассмотрению вопросов, связанных с интеграцией sc-конструкций и с интеграцией sc-подъязыков. **Интеграция двух sc-конструкций** – это их конкатенция, предполагающая склеивание синонимичных sc-элементов, принадлежащих разным интегрируемым sc-конструкциям. Напомним, что в рамках каждой sc-конструкции (в том числе и той, которая является результатом интеграции) неявная синонимия sc-элементов запрещена. Таким образом, основная проблема интеграции sc-конструкций заключается в поиске синонимичных sc-элементов, подлежащих склеиванию. Интегрировать различные языки – значит, привести к некоторому общему виду конструкции интегрируемых языков и обеспечить их совместное хранение и совместную переработку в памяти одной абстрактной машины. Приведение различных языков к общему виду может быть осуществлено путем построения sc-подъязыков, эквивалентных этим языкам. Интеграция различных sc-подъязыков сводится 1) к склеиванию ключевых узлов, принадлежащих разным sc-подъязыкам и оказавшихся синонимичными, и 2) к определению правил (операций) склеивания неключевых узлов, принадлежащих конструкциям разных sc-подъязыков и являющихся синонимичными. При этом возможна и более тонкая интеграция sc-подъязыков, заключающаяся в таком эквивалентном изменении интегрируемых sc-подъязыков, которое приводит к максимально возможному количеству синонимичных (и соответственно склеиваемых) ключевых узлов. Такая тонкая интеграция sc-подъязыков заключается в сближении семантически близких ключевых узлов, принадлежащих разным sc-подъязыкам, и приводит к сокращению общего количества ключевых узлов того sc-подъязыка, который является результатом интеграции. Очевидно, что из двух семантически эквивалентных sc-подъязыков лучшим следует считать тот, который имеет меньшее количество ключевых узлов.

4.7. Понятие абстрактной sc-машины

Ключевые понятия и идентификаторы ключевых узлов: абстрактная sc-машина, sc-память, sc-операция, микропрограмма sc-машины, интерпретация sc-машины, интеграция sc-машин, *node*, *const*, *arc*, *elem*, *var*, *pos*, *neg*, *fuz*, s-блокировка, e-блокировка, g-блокировка.

Абстрактная sc-машина (графодинамическая абстрактная машина, ориентированная на переработку sc-конструкций) представляет собой абстрактную графодинамическую параллельную асинхронную информационную машину, у которой внутренним языком является графовый язык SC и в графодинамической памяти в виде sc-конструкций хранится полная информация, необходимая не только для операций абстрактной sc-машины, но и для всех ее микропрограмм, т.е. все вспомогательные информационные конструкции, необходимые для реализации микропрограмм абстрактной sc-машины, также должны быть представлены в памяти этой машины.

Память абстрактной sc-машины, которую будем также называть **sc-памятью**, обеспечивает хранение текущего состояния перерабатываемой sc-конструкции. Никаких других информационных конструкций sc-память не содержит (точнее говоря, любые информационные конструкции, не являющиеся sc-конструкциями, могут храниться в sc-памяти, но только в том случае, если каждая из этих информационных конструкций является содержимым соответствующего константного sc-узла). Таким образом, sc-память представляет собой динамическую (меняющуюся во времени) sc-конструкцию и относится к классу графовых структурно-перестраиваемых (т.е. графодинамических) запоминающих сред. Переработка информации в sc-памяти сводится к следующему:

- генерации новых sc-узлов, которым сразу приписываются метка ***node*** и одна из меток, принадлежащих множеству **{*const*, *var*};**
- генерации новых sc-элементов неопределенного типа, которым сразу приписываются метка ***elem*** и метка из множества **{*const*, *var*};**
- генерации новых sc-дуг, каждая из которых проводится между двумя имеющимися sc-элементами; сгенерированной sc-дуге сразу приписываются метка ***arc***, а также метки из множеств **{*pos*, *neg*, *fuz*}** и **{*const*, *var*};**

- удалению (стиранию) имеющихся sc-элементов (sc-узлов, sc-элементов неопределенного типа, sc-дуг);
- формированию, удалению и модификации содержимого sc-узлов;
- формированию, удалению и модификации идентификаторов sc-элементов.

Следует подчеркнуть то, что в процессе переработки sc-конструкций метки sc-элементов (к числу которых относятся ***node*** , ***arc*** , ***elem*** , ***const*** , ***var*** , ***pos*** , ***neg*** , ***fuz***) меняться не могут в том смысле, что изменение метки по существу означает замену этого элемента на другой элемент, имеющий совершенно другую денотационную семантику.

Итак, в sc-памяти хранятся:

- sc-элементы, каждому из которых приписываются соответствующая метка и его принадлежность к конкретному типу (константа или переменная, а для дуг дополнительно – позитивная, негативная или нечёткая). При этом для sc-дуги дополнительно указываются соединяемые ею sc-элементы;
- информационные конструкции, каждая из которых является содержимым некоторого обязательно указываемого sc-узла;
- символьные информационные конструкции, каждая из которых является идентификатором некоторого обязательно указываемого sc-элемента.

Операции абстрактной sc-машины будем также называть **sc-операциями**, операциями над sc-конструкциями, операциями над sc-памятью. Разные абстрактные sc-машины имеют разные системы операций. Следовательно, понятие абстрактной sc-машины определяет целый класс (семейство) абстрактных машин, имеющих одинаковую организацию памяти, но разные системы операций.

Каждой sc-операции взаимно однозначно соответствует **микропрограмма**, определяющая операционную семантику этой sc-операции, т.е. уточняющая то, какое преобразование эта операция осуществляет над sc-памятью. Микропрограммы операций абстрактной sc-машины, как и других абстрактных машин являются демоническими, т.е. автономными, активными, самоинициируемыми программами. Демонические микропрограммы могут вызывать другие микропрограммы, которые являются уже пассивными, т.е. инициируемыми только в результате явного обращения к ним из других микропрограмм либо (в случае рекурсивной микропрограммы) из нее же самой. Таким образом, множество микропрограмм абстрактной sc-машины (как, впрочем, и любой другой абстрактной машины) может быть условно разбито на множества демонических микропрограмм и на множество пассивных микропрограмм. Преобразования, выполняемые пассивными микропрограммами абстрактной машины, будем условно называть базовыми преобразованиями этой машины.

Следует отметить, что одинаковая организация памяти для разных абстрактных sc-машин и запрет их микропрограммам использовать информационные конструкции, хранимые не в sc-памяти, создают все необходимые предпосылки для создания общего для всех sc-машин языка микропрограммирования. В качестве такого языка предлагается язык SCP (Semantic Code Programming) [411] (***ПрогрВАМ-2001кн***). В свою очередь, наличие языка микропрограммирования, общего для всех sc-машин, создает необходимые предпосылки для создания универсальной sc-машины, которая обеспечивает интерпретацию любой абстрактной sc-машины. В качестве такой универсальной sc-машины предлагается SCP-машина, рассматриваемая в [411] (***ПрогрВАМ-2001кн***).

Будем говорить, что sc-машина ***Cj*** **интерпретирует** sc-машину ***Ci*** в том и только в том случае, если:

- в памяти sc-машины ***Cj*** содержатся все sc-конструкции, перерабатываемые в памяти sc-машины ***Ci*** ;
- в памяти sc-машины ***Cj*** дополнительно хранятся все микропрограммы sc-машины ***Ci*** , представленные соответственно в виде sc-конструкций. Микропрограммы машины ***Ci*** в машине ***Cj*** становятся уже не микропрограммами, а хранимыми в памяти программами;
- в памяти sc-машины ***Cj*** дополнительно хранятся все вспомогательные данные, необходимые для реализации микропрограмм машины ***Ci*** ;
- система операций sc-машины ***Cj*** обеспечивает реализацию любой хранимой в ее памяти микропрограммы sc-машины ***Ci*** .

Графодинамическая абстрактная sc-машина является весьма перспективным (как в теоретическом, так и практическом плане) уточнением понятия абстрактной машины, так как хорошие метаязыковые возможности языка SC и его открытость дают основания претендовать абстрактной sc-машине на роль

мощного средства реализации самых различных формальных моделей, использующих самые различные языки. Но особенно важно то, что абстрактные sc-машины являются мощным средством описания операционной семантики самых различных языков (языков программирования, языков представления знаний). Для того чтобы описать операционную семантику какого-либо языка с помощью абстрактной sc-машины, необходимо:

1) разработать способ отображения произвольных конструкций описываемого языка в семантически эквивалентные sc-конструкции;
2) для каждой операции абстрактной машины, определяющей операционную семантику описываемого языка, построить эквивалентную ей операцию абстрактной sc-машины, описывающую определенный класс преобразований sc-конструкций, эквивалентных соответствующим конструкциям описываемого языка.

Другими словами, речь идет о моделировании на абстрактной sc-машине всевозможных абстрактных машин, но при взаимно однозначном отображении множества операций моделируемой абстрактной машины во множество операций абстрактной sc-машины. Последнее обстоятельство, обусловленное открытым характером операций абстрактной sc-машины, является принципиально важным. Принципиально важной является также возможность разработки общих принципов кодирования конструкций всевозможных языков с помощью конструкций языка SC, являющихся специальным видом однородных семантических сетей. Такие общие принципы сводятся к разработке системы ключевых sc-узлов, используемых при представлении конструкций любых языков. Наличие общих принципов кодирования конструкций всевозможных языков на языке SC обеспечивает не только совершенствование методов описания семантики этих языков, но и упрощение интеграции абстрактных машин, определяющих операционную семантику самых различных языков. Наличие таких общих принципов кодирования должно привести к тому, чтобы общность семантики различных языков проявлялась бы не только в совпадении некоторых ключевых sc-узлов в соответствующих им абстрактных sc-машинах, но и в совпадении некоторых операций этих абстрактных sc-машин.

Элементарные процессы абстрактной sc-машины выполняются параллельно и асинхронно над общей для них памятью (sc-памятью). Если области памяти, которые обрабатывают ("захватывают") элементарные процессы, не пересекаются, то эти процессы никак на влияют друг на друга. Взаимодействие элементарных процессов, обрабатывающих пересекающиеся области памяти, осуществляется следующим образом. При параллельном выполнении элементарных процессов над общими фрагментами sc-памяти действия, выполняемые над каждым хранимым в памяти sc-элементом, выполняются строго последовательно. Каждый элементарный процесс блокирует элементы обрабатываемой им sc-конструкции (хранимой в памяти) с помощью различных типов блокировки (**s-блокировки**, **e-блокировки** и **g-блокировки** – см. пояснения 4.6.2.1 – 4.6.2.3).

Пояснение 4.7.1. Блокировка sc-элемента, имеющая тип **s-блокировки** sc-элемента, запрещает другим элементарным процессам удалять этот элемент, а точнее, приписывать ему e-блокировку до тех пор, пока эта s-блокировка с указанного sc-элемента не будет снята. Следовательно, каждый элементарный процесс sc-машины, попытавшийся e-блокировать sc-элемент, s-блокированный другим элементарным процессом, прерывается и переходит в состояние ожидания снятия s-блокировки с указанного sc-элемента. Прерванный процесс продолжится после того, как он дождется указанной ситуации. Таким образом, sc-элемент, s-блокированный элементарным процессом абстрактной sc-машины, – это sc-элемент, существовавший до этого элементарного процесса и сохраняемый им до своего завершения. Один и тот же sc-элемент может быть s-блокирован несколькими элементарными процессами.

Пояснение 4.7.2. SC-элемент, **e-блокированный** элементарным процессом абстрактной sc-машины, – это sc-элемент, существовавший до этого элементарного процесса и удаляемый им на завершающей стадии своего выполнения. sc-элемент может быть e-блокирован только одним элементарным процессом. Для всех остальных элементарных процессов e-блокированный sc-элемент "невидим", т.е. для них этого sc-элемента уже не существует.

Пояснение 4.7.3. SC-элемент, **g-блокированный** элементарным процессом абстрактной sc-машины, порожден этим процессом и на завершающей стадии выполнения процесса либо удаляется (в случае, если это вспомогательный sc-элемент), либо освобождается от g-блокировки. SC-элемент может быть g-блокирован только одним элементарным процессом. Для всех остальных элементарных процессов g-блокированный sc-элемент "невидим", т.е. для них этого sc-элемента еще не существует.

Рассмотренные принципы разрешения конфликтов между параллельными элементарными процессами sc-машины, выполняемыми над общей sc-памятью, могут быть также использованы и для разрешения конфликтов между абстрактными машинами, параллельно работающими над общей памятью. Это оэначает, что проблемы интеграции абстрактных sc-машин фактически не существует – после

интеграции sc-подъязыков, соответствующих интегрируемым sc-машинам, достаточно просто объединить множества операций этих машин.

Такая легкая интегрируемость абстрактных sc-машин и обусловливает высокий уровень их гибкости (открытости, модифицируемости, наращиваемости).

Каждой sc-операции однозначно соответствует также некоторое множество sc-узлов, хранимых в памяти абстрактной sc-машины и являющихся константами всех микропрограмм, используемых для реализации указанной sc-операции (сюда входят соответствующая этой sc-операции демоническая микропрограмма, микропрограммы, к которым она обращается, микропрограммы, к которым обращаются эти микропрограммы, и т.д.). Заметим, что константами микропрограмм абстрактной sc-машины могут быть только константные sc-узлы, т.е. таковыми не могут быть ни переменные sc-узлы, ни sc-дуги любого вида, ни sc-элементы неопределенного типа. Множество всех sc-узлов, являющихся константами всех микропрограмм, используемых для реализации некоторой sc-операции, будем называть ключевыми узлами этой sc-операции. Образно говоря, это те sc-узлы, к которым соответствующая sc-операция как бы "привязывается". SC-узел, хранимый в памяти абстрактной sc-машины и являющийся ключевым по крайней мере для одной из ее операций, будем называть ключевым узлом этой sc-машины.

Очевидно, что все ключевые узлы абстрактной sc-машины должны входить в число ключевых узлов внутреннего языка этой машины. Заметим при этом, что ключевые узлы внутреннего языка абстрактной sc-машины кроме ключевых узлов самой машины могут включать в себя и другие узлы. То есть понятие ключевого узла графового языка несколько шире понятия ключевого узла графодинамической машины, работающей на этом графовом языке.

Поскольку разные абстрактные sc-машины отличаются разным набором операций, то и наборы ключевых узлов у них в общем случае будут разными. Следовательно, строго говоря, разные абстрактные sc-машины работают на разных внутренних языках, имеющих разные наборы ключевых узлов, но являющихся при этом подъязыками одного и того же базового языка – языка SC.

Итак, для всего семейства абстрактных sc-машин фиксируются:

- ядро внутренних языков этих машин – язык SC;
- принципы организации памяти, обеспечивающей хранение и переработку sc-конструкций. Память всех sc-машин является графодинамической (структурно-перестраиваемой) и ассоциативной, поддерживающей ассоциативный доступ по произвольному образцу (по образцовой sc-конструкции произвольного размера и произвольной структуры);
- параллельный, асинхронный характер реализации элементарных процессов;
- принципы разрешения конфликтов между элементарными процессами, параллельно выполняемыми над общей sc-памятью;
- принципы построения микропрограмм, в частности, запрещающие использование микропрограммами каких-либо вспомогательных данных, не хранимых в sc-памяти в виде соответствующих sc-конструкций.

Таким образом, в рамках всего семейства абстрактных sc-машин не фиксируются ни конкретный подъязык языка SC (разные sc-машины в качестве своего внутреннего языка могут использовать разные подъязыки языка SC), ни набор операций и соответствующих им микропрограмм (разные sc-машины могут иметь разный набор операций).

Общим для всех абстрактных sc-машин является только небольшой набор специальных операций, обеспечивающих просмотр и редактирование sc-конструкций, хранимых в графодинамической памяти. Ключевые узлы языка SC, используемые для записи соответствующих команд просмотра и редактирования, рассмотрены в подразделе 2.6.

Перечисленные выше общие принципы организации абстрактных sc-машин дают возможность существенно упростить решение проблемы интеграции абстрактных машин и формальных моделей (абстрактные машины, относящиеся к классу sc-машин, очевидно, существенно проще интегрировать, чем разнородные абстрактные машины), а также решение проблемы интерпретации абстрактных машин и формальных моделей (очевидно, что для интерпретации абстрактной sc-машины, особенно сложной, проще использовать абстрактную машину, также относящуюся к классу sc-машин).

4.8. Понятие формальной sc-модели

К л ю ч е в ы е п о н я т и я : формальная модель, формальная sc-модель.

Общее определение формальной модели приведено в пункте.1.1.1 (определение 1.1.).

Формальную модель, задаваемую языком, относящимся к классу sc-подъязыков, и абстрактной машиной, относящейся к классу абстрактных sc-машин, будем называть **формальной sc-моделью**.

Открытый (гибкий) характер формальных sc-моделей является важнейшим их достоинством и определяется легкой интегрируемостью формальных sc-моделей, которая, в свою очередь, обусловлена легкой интегрируемостью sc-подъязыков, sc-конструкций и абстрактных sc-машин. Интеграция формальных sc-моделей сводится к интеграции их языков (sc-подъязыков), к интеграции их аксиом (начальных sc-конструкций) и к интеграции их абстрактных машин (sc-машин). От интеграции формальных sc-моделей можно перейти к интеграции произвольных формальных моделей, которая сводится к следующим этапам:

1) построение sc-подъязыков, семантически эквивалентных языкам, используемым в интегрируемых формальных моделях, т.е. приведение указанных языков к общему каноническому виду – к некоторому способу представления текстов этих языков в виде sc-конструкций;
2) интеграция построенных sc-подъязыков с возможной корректировкой набора ключевых узлов для каждого из этих языков;
3) построение абстрактных sc-машин, семантически эквивалентных абстрактным машинам, используемым в интегрируемых формальных моделях;
4) интеграция построенных абстрактных sc-машин с возможной корректировкой набора ключевых узлов в каждой из этих машин;
5) построение sc-конструкций, семантически эквивалентных аксиомам интегрируемых формальных моделей и оформленных на указанных выше sc-подъязыках;
6) интеграция построенных sc-конструкций.

Выводы к разделу 4

Отличие языка SC от его базового подмножества (языка SCB) сводится:

- к расширению понятия scb-элемента путем включения в число элементов текста не только знаков множеств, но и простых переменных, значениями которых являются знаки множеств, а также метапеременных, значениями которых являются простые переменные. При этом значениями простых переменных могут быть знаки не только узловых множеств (в том числе знаки кортежей), но и знаки пар принадлежности;
- к расширению понятия множества путем включения в число возможных элементов множества не только знаков множеств, но и переменных.