15. КОНЕЧНЫЕ ПОСЛЕДОВАТЕЛЬНОСТИ И ИХ ПРЕДСТАВЛЕНИЕ В МАТЕМАТИЧЕСКИХ И КОМПЬЮТЕРНЫХ МОДЕЛЯХ

15.1. Конечные последовательности

Значение называется **конечной последовательностью**, если оно состоит из конечного числа значений, принадлежащих некоторой величине и расположенных в определенном порядке. **Величиной конечных последовательностей** называется множество всех конечных последовательностей, составленных из значений некоторой величины. Каждая величина конечных последовательностей является **сложной величиной**, которая помимо множества конечных последовательностей содержит **базу** - величину, которой принадлежат все элементы этих последовательностей. Величина конечных последовательностей **замкнута** относительно операций конкатенации (сцепления двух последовательностей), следующего и предыдущего элементов последовательности, функций головы (отбрасывания последнего элемента), хвоста (отбрасывания первого элемента), первого и последнего элемента, вычисления длины последовательности, а также отношений подпоследовательности, совпадения с началом и совпадения с концом последовательности.

Например, в вербальном представлении информации о программе на языке МИЛАН значением единственного термина *программа* является конечная последовательность символов – синтаксически правильная программа на языке МИЛАН.

Чтобы установить, какие величины конечных последовательностей используются при вербальном представлении информации, необходимо выяснить, информация о каких объектах с внутренней структурой должна быть представлена, и какие из этих объектов имеют структуру последовательности. Кроме того, как указывалось выше, обоснование системы величин конечных последовательностей ссылается на систему понятий, использованную при вербальном представлении этой информации, где эти величины или их подмножества являются объемами понятий.

15.2. Алгебраические системы конечных последовательностей

Каждая **алгебраическая система конечных последовательностей** определяется как многосортная алгебраическая система, носитель которой состоит из базы - некоторой алгебраической системы, а также множества всех конечных последовательностей, составленных из элементов носителя базы. Алгебраическая система конечных последовательностей **замкнута** относительно операций конкатенации, следующего и предыдущего элементов, функций головы, хвоста и вычисления длины последовательности, а также отношений подпоследовательности, совпадения с началом и совпадения с концом последовательности.

15.3. Модели типов данных конечных последовательностей

Модель любого типа данных **конечных последовательностей** совпадает с подходящей алгебраической системой конечных последовательностей. Каждый тип данных конечных последовательностей является **непримитивным** и содержит бесконечное множество значений.

15.4. Понятия, соответствующие конечным последовательностям

**Понятие** называется **соответствующим конечным последовательностям**, если его объем совпадает с некоторой величиной конечных последовательностей или является подмножеством такой величины. В примере раздела 15.1 понятием, соответствующим конечным последовательностям, является *программа* (объем этого понятия совпадает с множеством всех синтаксически правильных программ на языке МИЛАН). В данном примере множество онтологических соглашений может быть представлено грамматикой языка МИЛАН:

<программа> ::= начало<идентификатор><последовательность операторов>конец<идентификатор>

<последовательность операторов> ::= <оператор> ⏐ <оператор>;<последовательность операторов>

<оператор> ::= <присваивание> ⏐ <условный оператор> ⏐ <цикл> ⏐ <обмен>

<присваивание> ::= <идентификатор>:=<выражение>

<условный оператор> ::= если<логическое выражение>то<последовательность операторов>все ⏐ если<логическое выражение>то<последовательность операторов>иначе<последовательность операторов>все

<цикл> ::= пока<логическое выражение> цк<последовательность операторов>кц

<обмен> ::= ввод<идентификатор> ⏐ вывод<выражение>

<выражение> ::= <фактор> ⏐ <фактор> + <выражение>

<фактор> ::= <первичное> ⏐ <фактор>\*<первичное>

<первичное> ::= <идентификатор> ⏐ <константа> ⏐ (<выражение>)

<логическое выражение> ::= <выражение>=<выражение> ⏐<выражение>≠<выражение>

<константа> ::= <цифра> ⏐ <константа><цифра>

<цифра> ::= 0 ⏐ 1 ⏐ 2 ⏐…⏐ 9

<идентификатор> ::= <буква> ⏐ <идентификатор><буква>

<буква> ::= А ⏐ Б ⏐…⏐ Я

15.5. Термы языка прикладной логики, значениями которых являются множества конечных последовательностей

**Термом специализированного расширения** «ПОСЛЕДОВА- ТЕЛЬНОСТИ» языка прикладной логики, значение которого есть множество конечных последовательностей, является *seq t*, где *t* – терм, значением которого является множество, причем *Jαθ(seq t)* существует, если *Jαθ(t)* есть множество, *Jαθ(seq t)* есть множество всех конечных последовательностей, составленных из элементов множества *Jαθ(t)*. Термы языка прикладной логики, значениями которых являются множества конечных последовательностей, позволяют сопоставлять именам, определяемым прикладной логической теорией, эти множества последовательностей в качестве их сортов с помощью описаний сортов имен.

15.6. Термы языка прикладной логики, значениями которых являются последовательности, термы и формулы, связанные с последовательностями

**Термами специализированного расширения** «ПОСЛЕДОВА- ТЕЛЬНОСТИ» языка прикладной логики, значения которых суть последовательности или их элементы, являются:

*Λ*, причем *Jαθ(Λ)* есть пустая последовательность;

*«t1 … tm»*, причем *Jαθ(«t1 … tm»)* – последовательность, составленная из элементов *Jαθ(t1), …, Jαθ(tm)*;

*t1 || t2*, где *t1* и *t2* – термы, значения которых – последовательности, причем *Jαθ(t1 || t2) = Jαθ(t1) || Jαθ(t2)* (конкатенация);

*next(t1, t2)*, где *t1* и *t2* – термы, значение *t1* – последовательность, а *t2* – элемент этой последовательности (кроме последнего); *Jαθ(next(t1, t2))* есть элемент последовательности *Jαθ(t1)*, следующий за элементом *Jαθ(t2)*;

*prev(t1, t2)*, где *t1* и *t2* – термы, значение *t1* – последовательность, а *t2* – элемент этой последовательности (кроме первого); *Jαθ(next(t1, t2))* есть элемент последовательности *Jαθ(t1)*, предшествующий элементу *Jαθ(t2)*;

*head(t)*, где *t*– терм, значением которого является последовательность; *Jαθ(head(t))* есть последовательность *Jαθ(t)*, из которой удален последний элемент;

*tail(t)*, где *t*– терм, значением которого является последовательность; *Jαθ(tail(t))* есть последовательность *Jαθ(t)*, из которой удален первый элемент;

*first(t)*, где *t*– терм, значением которого является последовательность; *Jαθ(first(t))* есть первый элемент последовательности *Jαθ(t)*;

*last(t)*, где *t*– терм, значением которого является последовательность; *Jαθ(last(t))* есть последний элемент последовательности *Jαθ(t);*

*π(t1, t2)*, где *t1* и *t2* – термы, значение *t2* – последовательность, а *t1* – целое число, не меньшее 1 и не большее числа элементов последовательности *t2*; *Jαθ(π(t1, t2))* есть элемент последовательности *Jαθ(t2)*, имеющий номер *Jαθ(t1).*

Кроме того, термом этого специализированного расширения является:

*length(t)*, где *t* – терм, имеющий значением конечную последовательность, причем *Jαθ(μ(t))* есть длина последовательности *Jαθ(t)*.

**Формулами стандартного расширения** языка прикладной логики являются:

*sub(t1, t2)*, где *t1* и *t2* – термы, значениями которых являются последовательности, причем *Jαθ(sub(t1, t2))* истинно тогда и только тогда, когда *Jαθ(t1)* есть подпоследовательность *Jαθ(t2)*;

*beg(t1, t2)*, где *t1* и *t2* – термы, значениями которых являются последовательности, причем *Jαθ(beg(t1, t2))* истинно тогда и только тогда, когда *Jαθ(t1)* совпадает с началом последовательности *Jαθ(t2)*;

*end(t1, t2)*, где *t1* и *t2* – термы, значениями которых являются последовательности, причем *Jαθ(end(t1, t2))* истинно тогда и только тогда, когда *Jαθ(t1)* совпадает с концом последовательности *Jαθ(t2)*.

Кроме того, в **специализированном расширении** МАТЕМАТИЧЕСКИЕ КВАНТОРЫ языка прикладной логики вводятся следующие **формулы**:

*(∀(v1: t1) (v2: t2)… (vm: tm) f)* – квантор всеобщности, где *(v1: t1) (v2: t2)… (vm: tm)* есть множество описаний переменных (здесь последовательности рассматриваются как упорядоченные множества), переменные *v1, v2, …, vm* связаны в формуле *(∀(v1: t1) (v2: t2)… (vm: tm) f)*, *f* – формула, причем *Jαθ((∀(v1: t1) (v2: t2)… (vm: tm) t))* есть конъюнкция значений *Jαθ(f)* при всех допустимых для множества описаний переменных *(v1: t1) (v2: t2)… (vm: tm)* подстановках в формулу *f*;

*(∃(v1: t1) (v2: t2)… (vm: tm) f)* – квантор существования, где *(v1: t1) (v2: t2)… (vm: tm)* есть множество описаний переменных (и здесь последовательности рассматриваются как упорядоченные множества), переменные *v1, v2, …, vm* связаны в формуле *(∃(v1: t1) (v2: t2)… (vm: tm) f)*, *f* – формула, причем *Jαθ((∃(v1: t1) (v2: t2)… (vm: tm) t))* есть дизъюнкция значений *Jαθ(f)* при всех допустимых для множества описаний переменных *(v1: t1) (v2: t2)… (vm: tm)* подстановках в формулу *f*.

Прикладная логическая теория, моделирующая онтологию синтаксиса языка МИЛАН, имеет следующий вид.

Синтаксис языка МИЛАН(ST, ПОСЛЕДОВАТЕЛЬНОСТИ)

***сорт*** *программа: программы*

*программы ≡ {(v1: идентификаторы)(v2: последовательности операторов) «начало» || v1 || v2 || «конец» || v1}*

*последовательности операторов ≡ {(v1: операторы)(v2: последовательности операторов) v1 || «;» || v2}*

*операторы ≡ присваивания ∪ условные операторы ∪ циклы ∪ обмены*

*присваивания ≡ {(v1: идентификаторы)(v2: выражения) v1 || «:=» || v2}*

*условные операторы ≡ {(v1: логические выражения)(v2: последовательности операторов) «если» || v1 || «то» || v2 || «все»} ∪ {(v1: логические выражения)(v2: последовательности операторов) (v3: последовательности операторов) «если» || v1 || «то» || v2 || «иначе» || v3 || «все»}*

*циклы ≡ {(v1: логические выражения)(v2: последовательности операторов) «пока» || v1 || «цк» || v2 || «кц»}*

*обмены ≡ {(v1: идентификаторы) «ввод» || v} ∪ {(v: выражения) «вывод» || v}*

*выражения ≡ факторы ∪ {(v1: выражения)(v2: факторы) v1 || «+» || v2}*

*факторы ≡ первичные ∪ {(v1: факторы)(v2: первичные) v1 || «\*» || v2}*

*первичные ≡ идентификаторы ∪ константы ∪ {(v: выражения) «(» || v || «)»}*

*логические выражения ≡ {(v1: выражения)(v2: выражения) v1 || «=» || v2} ∪ {(v1: выражения)(v2: выражения) v1 || «≠» || v2}*

*константы ≡ цифры ∪ {(v1: константы)(v2: цифры) v1 || v2}*

*цифры ≡ {«0», «1», «2», …, «9»}*

*идентификаторы ≡ буквы ∪ {(v1: идентификаторы)(v2: буквы) v1 || v2}*

*буквы ≡ {«А», «Б», «В», …, «Я»}*