

Кукса П.П.

Московский Государственный Технический Университет
им. Н.Э. Баумана

E-mail: kouxa@online.ru

WWW: <http://www.geocities.com/pkouxa>

РАЗРАБОТКА ЯЗЫКА ОПИСАНИЯ НЕЧЕТКИХ СИСТЕМ

Маршрут проектирования нечетких систем (НС) включает этапы алгоритмического, архитектурного и схемного проектирования. На алгоритмическом уровне определяется функциональное поведение НС, определяются лингвистические переменные, их функции принадлежности (ФП), выбирается (разрабатывается) набор базовых операций (Т- и S-нормы, импликация $I(\cdot, \cdot)$, агрегации, дефазификации), осуществляется разработка концептуальной (качественной) модели системы или процесса. Нечеткая модель определяется множеством входов $x = \{x_1, \dots, x_p\}$, выходов $y = \{y_1, \dots, y_m\}$, множеством термов $T(x_i)$, множеством правил $RS = \bigcup R_i$ (и возможно ассоциированными с ними весами w_i), формами математических определений базовых операций. Целью разработки языка описания НС является моделирование НС, документирование, проведение иерархического функционально-структурного описания нечетких систем.

Основные требования, предъявляемые к языку:

- язык должен иметь средства для описания лингвистических переменных, функций принадлежности и нечетких правил
- язык должен позволять описывать поведение (алгоритм функционирования) НС в удобной форме: алгоритмической (оформленной в виде отдельной процедуры)
- иметь средства для определения структуры набора правил (блоков правил и связей между ними (интерфейсов))
- структура набора правил может быть произвольной (последовательное, параллельное, последовательно-параллельное включение отдельных блоков правил (модулей))
- обладать широким набором поддерживаемых нечетких операций с возможностью объявления новых математических определений
- язык должен позволять определять новые типы данных
- обеспечивать естественную запись и интерпретацию правил
- описание НС должно иметь модульную структуру
- наличие встроенных типов ФП
- возможность задания произвольной ФП
- возможность определения сложных понятий на основе более простых
- возможность определения *структурированных* лингвистических переменных (содержащих явный алгоритм (синтаксическое правило) порождения элементов термножества $T(X)$ и вычисления смысла $M(X)$ нечеткой переменной X)
- условная часть правил может иметь произвольную структуру (сочетание нечетких предикатов)
- метод описания должен быть независим от применяемого набора операций, содержание набора должно задаваться с помощью средств, основанных на этом методе

Многие программные средства [1, 2] разработки нечетких моделей представляют собой замкнутый программный продукт, имеют ограниченный и нерасширяемый набор нечетких операций (различный для разных программных сред). Это затрудняет сравнение подобных нечетких моделей, но разработанных в разных программных средах, а описание нечеткой

системы получается реализационно-ориентированным, “погруженным” в детали конкретного программного продукта, что скрывает внутреннюю (истинную) структуру нечеткой системы. Преодолеть указанные недостатки можно, если метод определения НС, заложенный в язык описания, во-первых, будет независим от применяемого набора нечетких операций, во-вторых, не будет накладывать ограничений на структуру системы.

В основе спецификации НС на языке описания лежат следующие основные принципы:

- конструкции языка, позволяющие описывать нечеткие правила, лингвистические переменные, термы (терм-множества), являются базовыми, на основе которых определяются более сложные компоненты модели: составные лингвистические значения, блоки правил, системные компоненты (блоки правил + интерфейс), составленная из блоков структура (набор, композиция), система в целом (системные компоненты, связанные между собой + интерфейс)
- алгоритм функционирования системы задается в спецификации системы в виде отдельной процедуры
- нечеткий логический вывод осуществляется в соответствии с выбранной схемой и набором нечетких операций
- спецификация НС является исходной для различных программных модулей [3] (моделирования, трансляции и пр.)

Рассмотрим структуру описания НС.

1. Объявление переменных

```
Идентификатор базовый тип {
    <Терм>
    <Терм>
    ... }
```

Смысл термина может быть определен посредством других термов: терм1 {если терм2 и терм3}.

2. Описание набора правил

```
Блок правил {
    <правило>
    ... }
```

3. Описание системных компонентов

```
Компонент (интерфейс){
    Блок правил
}
```

4. Описание системы

```
Система (интерфейс) {
    Компоненты {
        ...
    }
}
```

5. Алгоритм функционирования

```
{
    <оператор>
    <оператор>
    ... }
```

Конфигурацию механизма логического вывода (используемый набор операций, параметры вывода) представляется целесообразным вынести в отдельный секционированный блок, где каждая секция отвечает за определение того или иного элемента.

Для обеспечения возможности использования нечеткой модели в одном контексте с другими программными модулями спецификацию НС с языка описания можно оттранслировать в программу на другом языке высокого уровня.

Разработанные концепции языка описания и структура описания НС в значительной степени облегчают перевод спецификации НС в другой язык. Рассмотрим подробнее процедуру трансляции на примере языка С.

1. Описание набора элементарных операций (с использованием макросов)

`#define fz_or(a, b) ((a)>(b) ? (a) : (b))` – операция “или” над двумя нечеткими значениями
`#define fz_and(a, b) ((a)<(b) ? (a) : (b))` – операция “и” над двумя нечеткими значениями

2. Реализация ФП

```
unsigned int MFNAME (value) {
    /*реализация трапецеидальной ФП {p1, p2, p3, p4}*/
    if (value < p1) return(0);
    else {
        if (value < p2) return((value-p1)/(p2-p1));
        else {
            if (value < p3) return (1)
            else {
                return((p4-value)/(p4-p3));
            }
        }
    }
}
```

3. Реализация нечетких правил

Каждое правило реализуется следующей последовательностью действий:

`/*if x is small and dx is big then y is NB*/`

`v1 = x_small(x); /*степень принадлежности $\mu_{small}(x)$ */`

`v2 = dx_big(x); /* степень принадлежности $\mu_{big}(dx)$ */`

`v = fz_and(v1, v2); /* степень истинности предпосылки правила*/`

`y_NB(v);`

Последняя процедура работает в соответствии с выбранным методом дефазификации.

Для метода взвешенного нечеткого среднего:

```
void y_NB(w){
    denom+=w;
    nom+=(w*c);
}
```

Итоговое значение выходной переменной y определяется как $nom/denom$.

4. Реализация набора правил

```
void ruleb (void) {
    nom=denom=0;
    <правило 1>
    <правило 2>
    ...
    <правило r>
    y=nom/denom
}
```

5. Алгоритм функционирования системы

```
void alg(void) {
    while(1) {
        ...
        ruleb();
    }
}
```

... }
}

Литература

1. Inform FuzzyTech 5.5 User's manual, 2001, 257 p.
2. Jan E. Mortensen, JFS - development environment for the programming language JFL, 2001.
3. Кукса П.П. Система моделирования нечетких систем на алгоритмическом уровне //Информатика и системы управления в XXI веке. Сборник научных трудов. – М: Изд-во Эликс+, 2002, с. 201-204.