

Базовая архитектура нечеткой экспертной системы, реализуемой FuzzyClips

FuzzyCLIPS может иметь дело с чёткими, нечёткими и комбинированными аргументациями, что позволяет свободно смешивать нечёткие и нормальные условия в правилах и фактах экспертной системы. Система использует два понятия мягких вычислений: нечёткость (границы фрагмента информации не могут быть четко обозначены) и неопределённость (нельзя с уверенностью что-либо сказать о фрагменте информации).

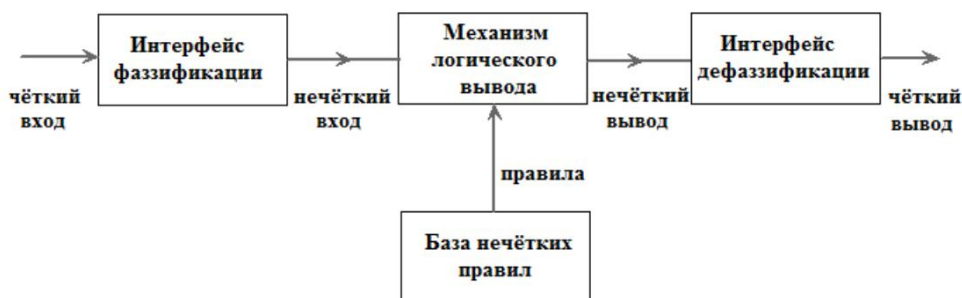


Рисунок : базовая архитектура нечеткой экспертной системы

Основными компонентами являются интерфейс фаззификации, база нечётких правил (база знаний), механизм логического вывода (логика принятия решений) и интерфейс дефаззификации. Входные переменные преобразуются в нечёткие посредством функций принадлежности, определённых на входных переменных и применяемых к фактическим значениям входных переменных для установления степени принадлежности для каждого правила.

Реализация факторов уверенности в FuzzyClips и нечеткость

В библиотеке FuzzyClips неточная степень уверенности, основанная на факторах может быть смоделирована одновременно с нечеткостью системы в целом. Сначала определяются факторы уверенности. FuzzyClips позволяет также использовать “определенные” факторы, например:

```
(defrule flight-rule
(declare (CF 0.95)); declares certainty factor of the rule
(animal type bird)
=>
(assert (animal can fly))
)
```

Это правило подчеркивает, что степень уверенности в правиле, что если животное - птица, то оно может летать составляет определенную величину, а именно 95% . Другие примеры FuzzyClips программ, генерирующие определенные факты представлены ниже:

```
(defrule GetFever
(declare (salience 100))
=>
(printout t "Enter confidence that patient has a fever (0 to 1):
")
(bind ?cf (read))
(assert (fever yes) CF ?cf))

(defrule GetSpots
(declare (salience 100))
=>
(printout t "Enter confidence that patient has spots (0 to 1):
")
(bind ?cf (read))
(assert (spots yes) CF ?cf))
```

```

(defrule GetSore
  (declare (salience 100))
  =>
  (printout t "Enter confidence that patient has a sore throat (0
to 1): ")
  (bind ?cf (read))
  (assert (sorethroat yes) CF ?cf))

```

Более того в исходный код CLIPS может быть внедрена нечеткость. Нечеткость возникает, когда граница фрагмента информации нечеткая. Например, такие понятия, как молодые, высокие, хорошие или большие, являются нечеткими. Не существует единого значения, определяющего термин «молодой». Для некоторых людей 25 лет - молодой, а для других 35 лет - молодой. На самом деле понятие «молодой» не имеет четких границ. Возраст 1 год определенно значит “молод”, а возраст 100 лет - точно “не молод”. Однако возраст 35 лет может восприниматься человеком как возраст молодого или не молодого и обычно зависит от контекста, в котором он рассматривается. Представление такого рода информации в FuzzyCLIPS основано на теории нечетких множеств. В отличие от классической теории множеств, где речь идет о объектах, принадлежность которых к множеству может быть четко описана, в теории нечетких множеств принадлежность элемента ко множеству может быть выражено степенью принадлежности. Например, нечеткий термин «молодой» может определяться следующим образом:

Возраст	Степень принадлежности
25	1.0
30	0.8
35	0.6

40	0.4
45	0.2
50	0.0

Это может быть выражено формулой:

$$\mu_{\text{young}}(25) = 1, \mu_{\text{young}}(30) = 0.8, \dots, \mu_{\text{young}}(50) = 0$$

Или графически:

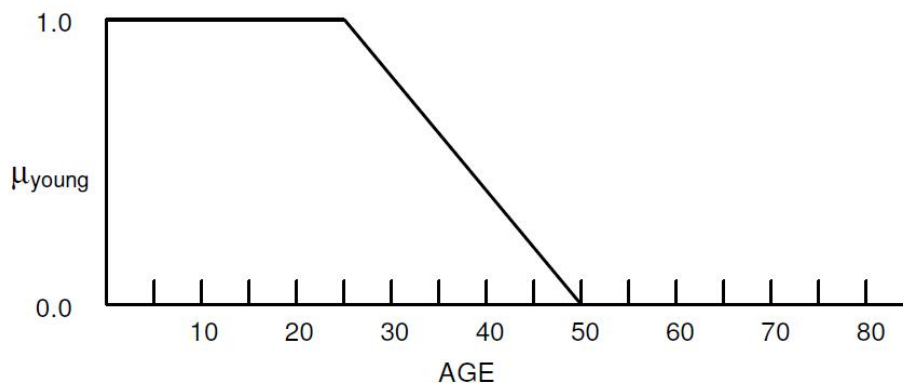


Рисунок :

Нечеткие переменные и функции принадлежности в FuzzyClips

В FuzzyClips все нечеткие переменные должны быть определены до использования конструктора `deftemplate`. Расширенный синтаксис приведен ниже:

```
(deftemplate <name> ["<Comments>"]
  <from> <to> [<unit>] ; Предметная область
  (
```

```

        t1
        .
        . ; Список первичных термов
        .
        tn
    )
)

```

Где <name> - идентификатор, который используется для нечеткой переменной. <from> и <to> - числа с плавающей запятой, а первичный терм t_i ($i = 1..n$) имеет следующий вид:

(<pname> <описание нечеткого множества>). Параметр pname определяет имя нечеткого множества а описание - соответствующую функцию принадлежности.

FuzzyClips предоставляет несколько встроенных функций принадлежности, которые могут применяться при решении различных задач. А именно, S- Функция, Z- Функция и П-Функция.

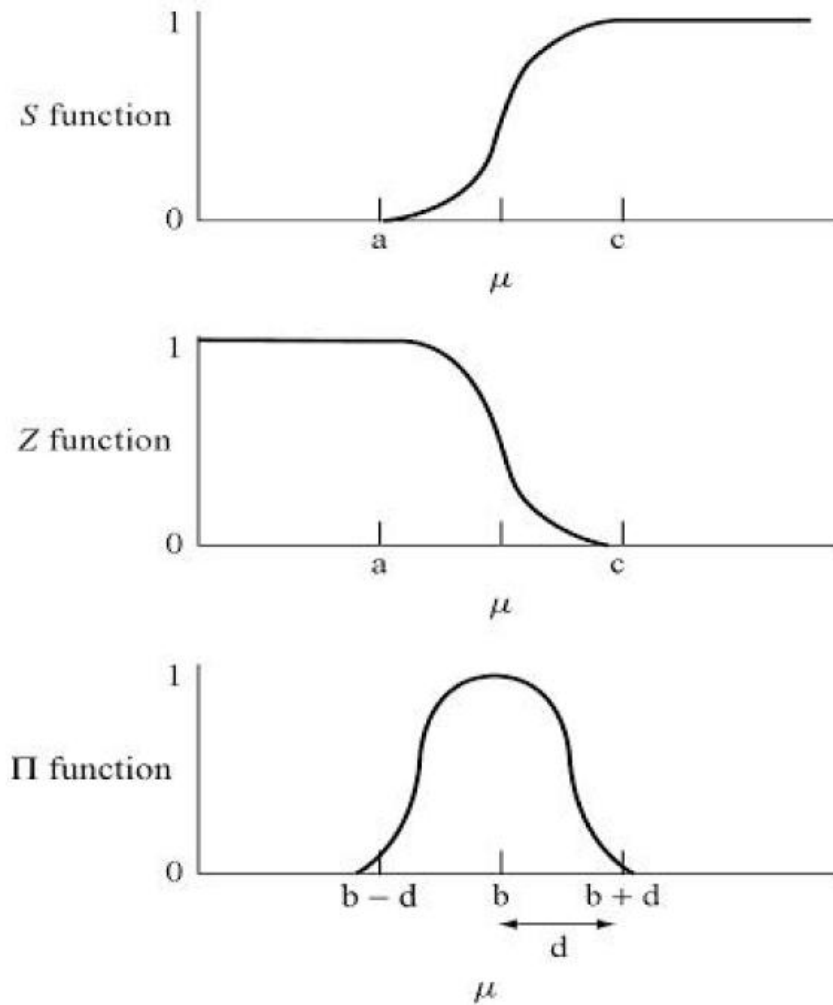


Рисунок : виды функции принадлежности

Функция принадлежности в библиотеке FuzzyClips может представляться 3 разными способами, о которых пойдет речи ниже.

Представление в виде синглтона

Степень принадлежности $\mu_A(x)$ в нечетком множестве A является положительным числом, а пара $(\mu_A(x), x)$ называется синглтоном (часто эти пары представлены $\mu_A(x)/x$ или $(x)/x$ для краткости). Нечеткое множество A в универсуме дискурса U можно описать следующим образом:

$$A = \int_{x \in U} \mu_A(x) / x$$

где интегральный символ обозначает объединение синглтонов.

Если универсум дискурса U является конечным множеством, то A выражается следующим образом:

$$A = \bigcup_{i=1}^n \mu(x_i) / x_i = \mu(x_1) / x_1 + \mu(x_2) / x_2 + \dots + \mu(x_n) / x_n$$

В FuzzyCLIPS мы рассматриваем универсум дискурса как диапазон действительной числовой линии и не имеем дело с конечными множествами для U . Синглтон будет представлен здесь как пара $(x_i \quad \mu_i)$. Нечеткое множество A будет представлено в виде списка синглтонов

$$\langle \text{singletons} \rangle ::= (x_1 \quad \mu_1) (x_2 \quad \mu_2) \dots (x_n \quad \mu_n)$$

где:

$$\begin{aligned} & x_i \leq x_{i+1} \text{ for } i = 1, 2, \dots, n-1 \\ & x_i \text{ элемент } U \\ & \mu_i \text{ число, обозначающее степень принадлежности } x \text{ в нечетком множестве } A \end{aligned}$$

Как уже упоминалось в разделе 4.1, нечеткое множество представлено упорядоченным множеством точек, соединенных сегментами прямой линии. Степень принадлежности значения x , не указанного в списке синглтонов, будет рассчитываться на основе интерполяции по следующей формуле (для точек, которые не имеют нескольких значений принадлежности, в этих случаях значение принадлежности определяется как максимальное для всех значений при одном и том же значении x):

$$\begin{aligned} \mu(x) &= \mu(x_1), & x \leq x_1 \\ \mu(x) &= \mu(x_i) + \frac{\mu(x_{i+1}) - \mu(x_i)}{x_{i+1} - x_i} (x - x_i), & x_i < x \leq x_{i+1} \\ \mu(x) &= \mu(x_n), & x_n < x \end{aligned}$$

Example

Let $U = \{x \mid 0 \leq x \leq 9\}$

Мы можем определить нечеткое множество следующим образом

$$\begin{aligned} \mu(0) &= 0, & \mu(1) &= 0, & \mu(2) &= 0.3, & \mu(3) &= 0.9, \\ \mu(4) &= 1, & \mu(5) &= 0.8, & \mu(6) &= 0.5, & \mu(7) &= 0, \\ \mu(8) &= 0, & \mu(9) &= 0 \end{aligned}$$

Можно представить это нечеткое множество следующим списком синглтонов:

(1 0) (2 0.3) (3 0.9) (4 1) (5 0.8) (6 0.5) (7 0)

Можно также показать этот набор графически, как на рисунке 15: Нечеткий набор из нескольких групп.

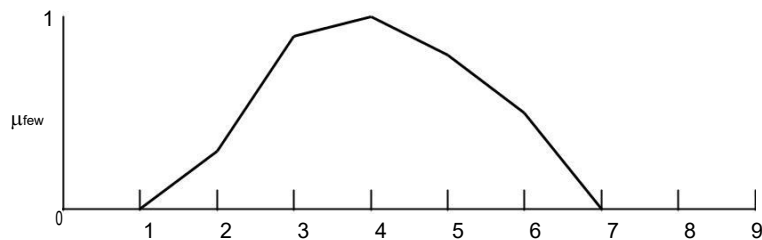


Рисунок : набор значений синглтонов

В FuzzyCLIPS можно определить группу лингвистических переменных, которая имеет универсум дискурса от 0 до 9 (единицы неопределенные).

```
(deftemplate group                ;a linguistic variable declaration
  0 9                            ;universe of discourse limits (no units)
  (                               ;start of primary term declarations
    ; a primary term few described in singleton notation
    (few (1 0) (2 0.3) (3 0.9) (4 1) (5 0.8) (6 0.5) (7 0))
  )                               ;end of primary term declarations
)                                 ;end of fuzzy deftemplate
```

Обратите внимание, что для последовательных x значения могут быть одинаковыми. Это описывает четкую границу в нечетком множестве (вертикальная линия). Если более трех точек имеют одинаковое значение, то останется только три (четвертую всегда заменит третья). Если две точки имеют одинаковые значения x и y , то одна из них будет отброшена. Рассмотрим следующие два примера нечетких множеств с четкими границами.

Example

Единичный набор, описанный как (три (3 0) (3 1) (3 0))

Можно представить четкую концепцию 3 как нечеткое множество. Он изображен графически на рисунке 16.

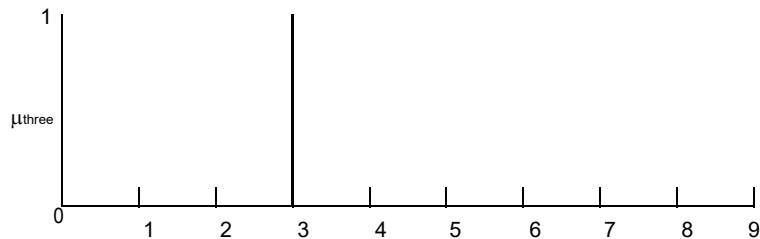


Рисунок : четкая концепция числа “три”

Example 16

Другой более сложный (и, вероятно, нереалистичный) набор может быть определен с помощью следующего набора значений синглтонов

(weird (1 0) (1 1) (1 0) (3 0) (4 .25)
(4 1) (4 .4) (4 .5) (6 1) (8 0))

График отображает набор выше. Заметим, что в этом случае точка (4 .4) отбрасывается.

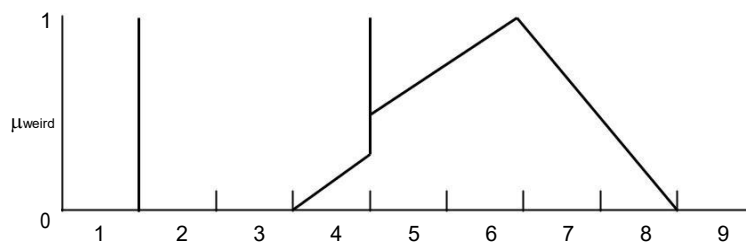


Рисунок : набор значений синглтонов

Представление в виде стандартной функции.

Часто бывает полезно описать функцию принадлежности, используя один из наборов стандартных функций S, PI, или Z (см стр. 5). Параметры этих функций могут быть выбраны в зависимости от приложений.

$$\begin{aligned} S(u, a, c) &= 0, \quad u \leq a, \quad u \in U \\ S(u, a, c) &= 2 \frac{u - a}{c - a}, \quad a < u \leq \frac{a + c}{2} \\ S(u, a, c) &= 1 - 2 \frac{c - u}{c - a}, \quad \frac{a + c}{2} < u \leq c \\ S(u, a, c) &= 1, \quad c < u \\ Z(u, a, c) &= 1 - S(u, a, c) \end{aligned}$$

$$\begin{aligned} \Pi(u, d, b) &= S(u, b - d, d), \quad u \leq b \\ \Pi(u, d, b) &= Z(u, b, b + d), \quad b < u \end{aligned}$$

Стандартное представление функции

принадлежности имеет следующий формат :

$\langle \text{standard} \rangle ::= (S\ a\ c) \mid (s\ a\ c) \mid (Z\ a\ c) \mid (z\ a\ c) \mid (PI\ d\ b) \mid (pi\ d\ b)$

где : a, b, c, d числа, представляющие параметры соответствующих функций.

Example

```
(deftemplate Tx "output water temperature"
  5 65 Celsius
  ( (cold (z 10 26))      ;standard set representation
    (OK (PI 2 36))
    (hot (s 37 60))
  )
)
```

FuzzyCLIPS преобразует все стандартные обозначения в синглтон. Для представления функций выбраны девять точек, равномерно расположенных вдоль оси x (см. Рис. 19: Аппроксимация стандартных функций). Количество точек (9) можно изменить, изменив значение `ArraySIZE`, которое определено в файле *fuzzyprsr.h*, а затем перекомпилировать FuzzyCLIPS. Обратите внимание, что увеличение этого размера увеличит вычислительную нагрузку во время нечеткого вывода. Также обратите внимание, что во многих случаях простой набор синглтонов из 2 или 3 точек, который аппроксимирует эти функции будет приемлемым и приведет к меньшему времени вычисления во время вывода.

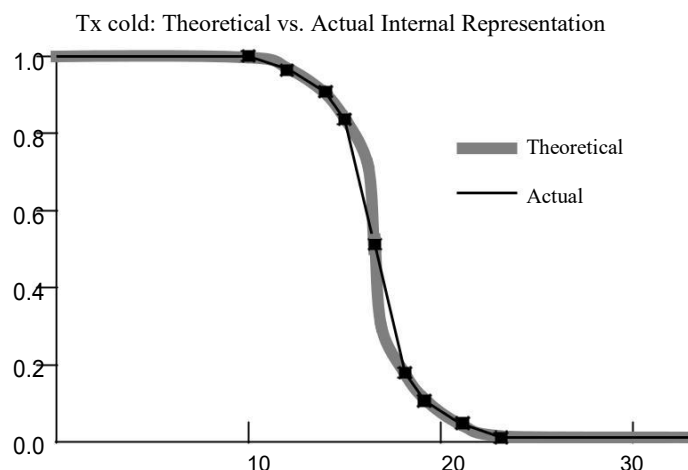


Рисунок : Аппроксимация стандартных функций

В качестве особого замечания рассмотрим случай, когда параметр a равен c для S- или Z-функций. В этом случае для представления функции будут использованы две точки.

Example 18

```
(S 10 10) будет представлено (10 0) (10
1)
(Z 10 10) будет представлено (10 1) (10
0)
```

Для функции PI, если параметр d равен нулю, функция будет представлена тремя точками.

Example

```
(PI 0 10) будет представлено (10 0) (10 1) (10 0)
```

Представление языков выражений

Example

```
(deftemplate temperature
  0 100 C
  ( (cold (z 10 26))           ;standard set representation
    (hot (s 37 60))           ;standard set representation
    (warm not [ hot or cold ]) ; linguistic expression
  )
)
```

Обратите внимание, что термин «теплый» описывается как не горячий или не холодный. В нем используются термины «горячий»

и «холодный», которые ранее были определены в этой модели для описания теплой концепции. Могут использоваться только термины, описанные в этом шаблоне (до определения определения термина, вместе с любыми доступными модификаторами (и, или) и операциями).

Команды в FuzzyClips

Ряд команд, поставляемых с CLIPS, можно использовать для просмотра нечетких фактов и шаблонов, так же как и обычных фактов (см. Команды факты, `ppdeftemplate`, `list-deftemplates`, `undeftemplate`, `ppdeffacts`, `list-deffacts`, `undeffacts`). Следующие команды или функции добавляют возможность доступа к компонентам нечетких фактов, управляют порогами для обхода правил, задают точность отображения нечеткого множества, задают тип нечеткого вывода, устанавливают порог для нечеткого сопоставления с образцом, создают и управляют нечеткими значениями и т. д.

1. Доступ к универсуму дискурса (`get-u`, `get-u-from`, `get-u-to`, `get-u-units`):

Команда: `get-u`

Использование: (`get-u` ?<fact-var>) or (`get-u` <integer>) or (`get-u` <fuzzy-template-name>) or (`get-u` <fuzzy-value>)

Где:

- ?<fact-var> - это переменная факта, обычно связанная в LHS с правилом
- <integer> это число фактов в списке фактов (константа или выражение)

- `<fuzzy-template-name>` это символ, который представляет имя нечеткой шаблона `deftemplate`
- `<fuzzy-value>` является элементом типа `FUZZY-VALUE`

Цель: Возвращает строку вида “`<from> - <to> <units>`” пределы универсума дискурса и единиц (если указано). Если в выражении `deftemplate` не было объявлено никаких единиц, функция возвращает “`<from> - <to>`”.

get-u-from - возвращает нижнюю границу универсума дискурса в формате с плавающей точкой.

get-u-to - возвращает верхний предел универсума дискурса в формате с плавающей точкой

get-u-units - возвращает единицы универсума дискурса в строковом формате. Если никакие единицы не указаны, то возвращается пустая строка.

2. Доступ к нечеткому множеству (`get-fs`, `get-fs-x`, `get-fs-y`, `get-fs-length`, `get-fs-lv`, `get-fs-value`)

Команда: `get-fs` — возвращает все нечеткое множество в одноэлементном представлении в строковом формате

get-fs-length - возвращает число пар в описании нечеткого множества как целое число

get-fs-x - возвращает x-координату *i*-ой пары в нечетком множестве, где пары нумеруются слева направо от 0 до (*n*-1), а *n* - общее количество пар в наборе. Если выражение `<i>` оценивает нецелое значение, оно усекается до ближайшего целого числа. Координата *x* возвращается как значение с плавающей запятой.

Используется следующим образом: (`get-fs-x` ?`<fact-var>` `<i>`) or (`get-fs-x` `<integer>` `<i>`) or (`get-fs-x` `<fuzzy-value>` `<i>`), где `<i>` - целое, переменное или функциональное выражение.

get-fs-y - возвращает y-координату i-той пары в нечетком множестве как значение с плавающей запятой.

get-fs-lv - возвращает лингвистическое значение, связанное с нечетким множеством.

get-fs-value - возвращает значение нечеткого набора с указанным значением x (<число>). <Число> - это значение, которое должно лежать между нижним и верхним пределами универсума дискурса для нечеткого множества.

3. Доступ к коэффициенту достоверности

get-cf - возвращает фактор достоверности факта как число с плавающей запятой.

4. Включение и отключение вычислений фактора достоверности в правилах

enable-cf-rule-calculation - после выполнения этой команды любые факты, утверждаемые в правиле, будут использовать соответствующие вычисления для определения значения достоверности для факта.

disable-cf-rule-calculation - после выполнения этой команды любые факты, утверждаемые в правиле, всегда будут иметь значение достоверности, указанное для факта, и если оно не указано, оно будет иметь значение 1.0. Значение достоверности правила или сопоставленных фактов не будет считаться.

5. Доступ к пороговому коэффициенту достоверности

set-threshold - устанавливает пороговый коэффициент достоверности в значение <NUMBER>. <NUMBER> должен

вычислять до плавающего значения между 0.0 и 1.0. По умолчанию пороговое значение равно 0.0

get-threshold - возвращает значение с плавающей запятой порогового коэффициента достоверности, если пороговое значение включено. Если он выключен, возвращается значение 0.0.

6. Настройка поведения оценки правила CF

set-CF-evaluation - устанавливает поведение для оценки CF правил в значение <value>. Значение должно быть одно из: когда определено (по умолчанию) или когда активировано. Это похоже на функцию SETAL-функции оценки CLIPS. Значение when-defined заставляет фактор достоверности правила оцениваться во время определения правила (компиляция). Значение when-activated заставляет фактор достоверности правила определяться во время определения правила и когда правило активируется (добавляется в повестку дня).

get-CF-evaluation - возвращает текущую настройку поведения для оценки CF правил. Возвращаемое значение является либо когда-либо определенным, либо когда-активированным (аналогично функции get-salience-evaluation).

7. Контроль точности отображения нечетких множеств

set-fuzzy-display-precision - когда отображаются нечеткие факты, значения нечетких множеств отображаются в формате с плавающей запятой. Эта функция позволяет отображать количество значащих цифр после десятичной точки. Аргумент <integer> представляет собой целое число от 2 до 16. Если оно меньше 2, то устанавливается значение 2, а если оно больше 16, устанавливается

значение 16. Значение по умолчанию равно 4. Обратите внимание, что сброс не сбрасывается это значение равно 4.

get-fuzzy-display-precision - возвращает целочисленное значение, которое является текущей отображаемой точностью.

8. Управление методом нечетких выводов

set-fuzzy-inference-type - устанавливает текущий тип вывода в один из макс-мин или max-prod. По умолчанию используется значение max-min. Обратите внимание, что clear не будет сбросить это значение до max-min.

get-fuzzy-inference-type - возвращает символ, который является одним из max-min или max-prod, указывающим текущий тип вывода.

9. Установка порога соответствия нечеткого шаблона

set-alpha-value - когда нечеткие слоты соответствуют нечетким шаблонам в правилах LHS, совпадение считается успешным, если есть какое-либо перекрытие (пересечение) между двумя нечеткими множествами, участвующими в сопоставлении. Эта функция позволяет совпадению достичь успеха только в том случае, если максимальное значение набора пересечений имеет значение принадлежности, большее или равное этому порогу. Альфа-значение по умолчанию равно 0.0. Если альфа-значение равно 0.0, максимальный набор пересечений должен быть больше 0.0. Обратите внимание, что сброс не сбрасывает значение альфа в 0.0.

get-alpha-value - возвращает значение с плавающей запятой, которое является текущим альфа-значением.

10. Функция предиката нечетких значений

fuzzyvaluep - эта функция возвращает TRUE, если аргумент имеет тип FUZZY-VALUE, в противном случае он вернет FALSE.

11. Создание и работа с FUZZY-VALUE

create-fuzzy-value - эта функция позволяет создать нечеткое значение. Нечеткое значение - это нечеткое множество, которое связано с конкретной нечеткой дефлему. Нечеткий deftemplate определяет универсум дискурса для нечеткого множества и терминов, которые могут быть использованы для описания нечеткого множества. Первый аргумент, <fuzzy-deftemplate-name>, - это имя нечеткого шаблона deftemplate. Остальные части описывают нечеткое множество, как это делается для нечеткого слота, когда утверждается нечеткий факт. Это может быть лингвистическое выражение, одноэлементная спецификация или стандартное функциональное выражение.

fuzzy-union - возвращает новое нечеткое значение, которое является объединением двух других нечетких значений. Оба аргумента должны иметь тип FUZZY-VALUE.

fuzzy-intersection - возвращает новое нечеткое значение, которое является пересечением двух других нечетких значений. Оба аргумента должны иметь тип FUZZY-VALUE.

fuzzy-modify - возвращает новое нечеткое значение, которое является модификацией аргумента нечеткого значения. Модификация выполняется с помощью аргумента-модификатора. Этот модификатор может быть любым активным модификатором (очень, слегка и т. д.).

12. Доступ к нечеткому слоту в факте

get-fuzzy-slot - эта функция будет получать нечеткое значение, связанное с нечетким слотом, по факту. Первым аргументом может быть переменная, которая связана с адресом факта или целым числом, которое является фактом для факта. Если факт является нечетким фактом `deftemplate` (тот, чье имя отношения является нечетким именем `deftemplate`), тогда второй аргумент не нужен, поскольку единственным слотом для этого факта является нечеткое значение. Если факт является стандартным фактом `deftemplate` с нечеткими интервалами, то вторым аргументом является символ, который идентифицирует доступный слот. (Обратите внимание: слот с нечеткими фактами `deftemplate` всегда называется «`GenericFuzzySlot`», и к нему можно получить доступ, используя это имя.).

13. Отображение нечеткого значения в функции форматирования

Это не функция или команда, а дополнение к CLIPS, позволяющее форматировать нечеткие значения в функции форматирования. Используется указатель `%F`.

14. Построение нечеткого значения

plot-fuzzy-value – эта функция используется для построения нечетких множеств. Имеет следующие аргументы:

`<logicalName>` - любой открытый маршрутизатор для направления вывода (например, `t` для стандартного вывода).

`<plot-chars>` - указывает символы, которые будут использоваться при построении графика (например, `*` или `"* +."`): для каждого заданного нечеткого значения соответствующий символ из строки или символа используется в качестве символа построения

- если задано больше нечетких значений, чем символов, то последний символ используется для оставшихся участков.

<low-limit> - представляет собой числовое значение, которое задает наименьшее значение x , которое будет отображаться на графике, или если оно не является числовым значением, тогда по умолчанию будет использоваться нижний предел универсума дискурса.

<high-limit> - представляет собой числовое значение, которое указывает наивысшее значение x , которое будет отображено в графике ИЛИ, если это не числовое значение, тогда по умолчанию будет установлен верхний предел универсума дискурса.

<fuzzy-value> - является одной из 3 вещей:

- Целое число, которое идентифицирует факт нечеткого deftemplate (в этом случае нечеткое значение из факта извлекается и используется)
- Переменная с нечетким фактографическим адресом факта (в этом случае нечеткое значение из факта извлекается и используется)
- Переменная с нечетким значением

Символ + идентифицирует, что может присутствовать один или несколько аргументов <fuzzy-value>.

deftemplate, связанная со ВСЕМИ нечеткими значениями, которые должны быть построены на одном и том же участке, должна быть одинаковой. Это необходимо, так как ось x должна иметь то же значение.

Значения <high-limit> и <low-limit> позволяют отображать окно универсума дискурса и обеспечивают масштабирование графика по оси x .

15. Управление результатом деактивизации

Функции defuzzification определены для возврата значений, которые зависят от нечеткого значения, которое они предоставляют. Функция момента-defuzzify не определена в случае нечеткого множества с площадью, равной нулю. Это может произойти, когда нечеткое множество является четким значением, или нечеткое множество является плоским при значении принадлежности 0.0. В этих случаях по умолчанию функция возвращает среднюю точку универсума дискурса. Для того чтобы пользователь мог обнаружить эти ситуации, предусмотрена специальная функция.

is-defuzzify-value-valid - эта функция используется, чтобы проверить, действительно ли значение, возвращаемое последней функцией defuzzify, или нет. Он возвращает TRUE, если верное значение было возвращено или FALSE, если нет. В текущей реализации только функция-defuzzify может возвращать недопустимое значение по умолчанию.

Если функции defuzzify еще не вызывались, возвращаемое значение TRUE.

Обычно эта функция будет использоваться сразу же после вызова функции defuzzify.

16. Пример использования CLIPS:

Здесь мы рассмотрим описание вывода просмотра фактов и правил для примера, найденного в файле simplTst.clp. Обратите внимание, что это не было реальным примером использования FuzzyCLIPS.

```
CLIPS>(load "simplTst.clp")  
Defining deftemplate: speed_error
```

```

Defining deftemplate: speed_change
Defining deffacts: my_facts
Defining defrule: speed-too-fast +j
Defining defrule: speed-ok +j
Defining defrule: get-crisp-value-and-print-rslt +j
TRUE
CLIPS> (reset)
==> f-0 (initial-fact) CF 1.00
==> f-1 (speed_error zero) CF 0.90 ;linguistic
description of fuzzy set
( (0.0 1.0) (0.11 0.0) ) ;singletons describe fuzzy set
in detail
==> Activation 0 speed-ok: f-1
==> Activation 0 speed-too-fast: f-1
CLIPS> (run)
FIRE 1 speed-too-fast: f-1
==> f-2 (speed_change ???) CF 0.63 ;CF = 0.9 * 0.7 for
fuzzy-fuzzy rule
( (0.1 0.0) (0.1495 0.0991) )
==> Activation -1 get-crisp-value-and-print-rslt: f-2
FIRE 2 speed-ok: f-1
<== f-2 (speed_change ???) CF 0.63 ;retraction of fuzzy
fact and ...
( (0.1 0.0) (0.1495 0.0991) )
<== Activation -1 get-crisp-value-and-print-rslt: f-2
==> f-3 (speed_change ???) CF 0.63 ;reassertion as fact
is modified
( (0.0 1.0) (0.1 0.1) (0.1333 0.06667) (0.1495 0.0991) )
==> Activation -1 get-crisp-value-and-print-rslt: f-3
FIRE 3 get-crisp-value-and-print-rslt: f-3
Change speed by a factor of: 0.3553202565269306
3 rules fired Run time is 0.06400000000212458 seconds.
46.87499999844391 rules per second.
3 mean number of facts (3 maximum).

```

```
1 mean number of instances (1 maximum).  
1 mean number of activations (2 maximum).  
CLIPS>
```

Пример определения нечеткого множества в FuzzyClips и манипуляции с ним.

Рассматривая вышеизложенное понятие “Возраст” , должны согласовываться между собой два принципа:

1. age - нечеткая переменная, которая принадлежит нечетким множествам “young”, “middle” и “old”.

2. Сущность (в данном случае Person) имеет четкий слот со значением “name” и нечеткий слот, который представляет нечеткую переменную age.

Таким образом код на FuzzyClips может быть написан следующим образом:

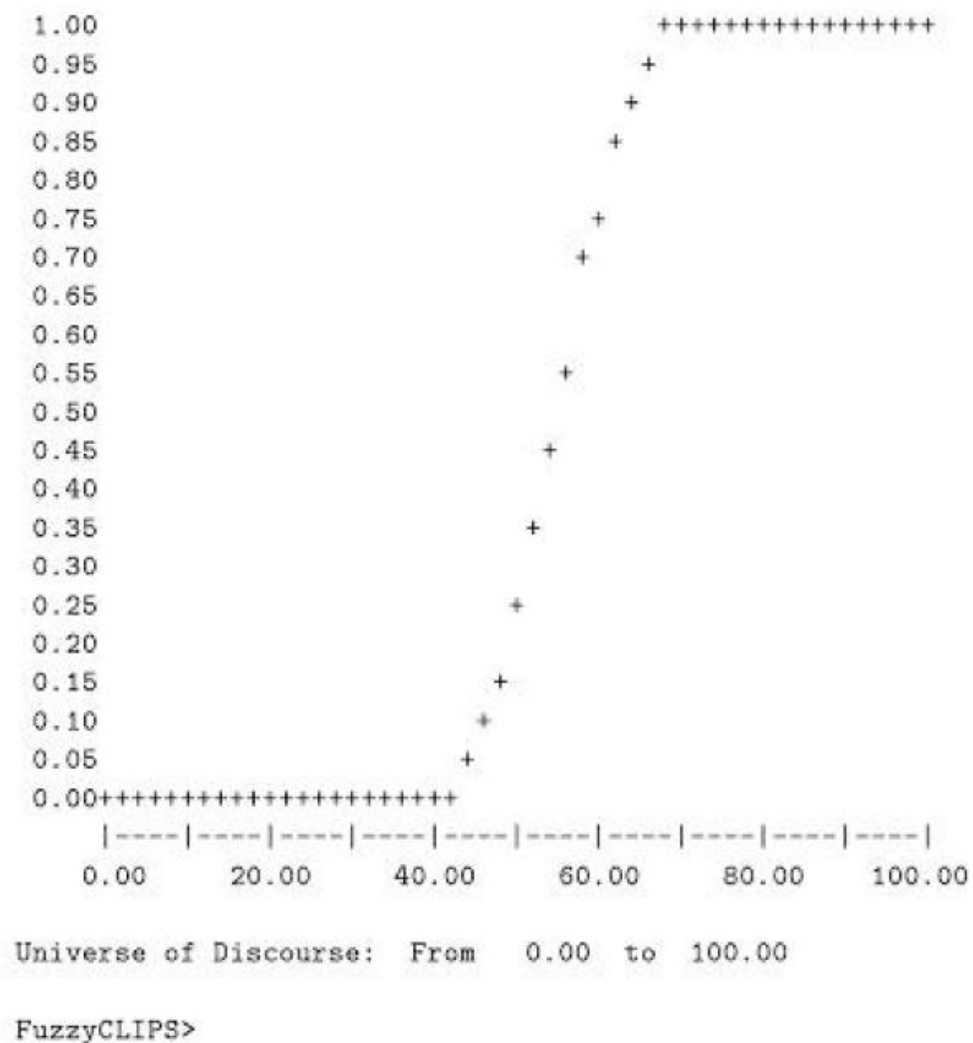
```
(deftemplate age  
  0 100 ; universe  
  ( (young (0 1) (25 1) (40 0.5) (55 0))  
    (middle(0 0) (25 0.5) (40 1.0) (55 0.5) (70 0))  
    (old    (0 0) (40 0) (55 0.5) (70 1) (80 1))  
  )  
  
(deftemplate sfage  
  0 100 ; universe  
  ( (young (z 30 55))  
    (middle(pi 15 40)  
    (old    (s 40 70))  
  )
```

```
(deftemplate person
  (slot name)
  (slot age (type FUZZY-VALUE age))
)
```

```
(deffacts startup
  (person (name bob) (age middle))
  (person (name katie) (age young))
)
```

FuzzyClips позволяет проводить многочисленные манипуляции с нечеткой базой знаний из командной оболочки, например:

```
FuzzyCLIPS> (get -u age)
0.00 - 100.00
FuzzyCLIPS> (get-fuzzy-inference-type)
max-min
FuzzyCLIPS> (plot-fuzzy-value t + nil nil
(create-fuzzy-value age middle))
)
```

Вывод четкого результата используя нечеткое множество в FuzzyClips

Функция *FuzzifyTemperature* фаззифицирует четкое входное значение с учетом возможной погрешности + или - пару градусов, тем самым генерирую нечеткое треугольное число.

```
(deftemplate Fever
  96 104
  ((fever (99 0) (103 1))))
```

```

(defrule getTemperature
  (declare (salience 100))
  =>
  (printout t "Enter temperature: ")
  (bind ?t (read))
  (assert (temperature ?t)))

(defrule FuzzifyTemperature
  (temperature ?t)
  =>
  (bind ?t1 (- ?t 2))
  (bind ?t2 (+ ?t 2))
  (assert (Fever (?t1 0) (?t 1) (?t2 0))))

(defrule GetFlu
  (Fever fever)
  =>
  (assert (flu yes)))

(defrule ShowFlu
  ?f <- (flu yes)
  ?fev <- (Fever ?)
  =>
  (printout t "Confidence in flu: " (get-cf ?f) crlf)
  (plot-fuzzy-value t "*+" nil nil (create-fuzzy-value Fever
fever) ?fev))

```

Предопределенные модификаторы в FuzzyClips

FuzzyCLIPS имеет набор предопределенных модификаторов, которые могут быть использованы в любое время для описания нечетких понятий. Описание модификаторов и пример работы с ними приведен ниже:

<u>Modifier Name</u>	<u>Modifier Description</u>
not	$1-y$
very	y^{**2}
somewhat	$y^{**0.333}$
more-or-less	$y^{**0.5}$
extremely	y^{**3}
above	(see [12])
below	(see [12])
intensify2	(y^{**2}) if y in $[0, 0.5]$ $1 - 2(1-y)^{**2}$ if y in $(0.5, 1]$
plus	$y^{**1.25}$
norm	normalizes the fuzzy set so that the maximum value of the set is scaled to be 1.0 ($y = y * 1.0 / \text{max-value}$)
slightly	intensify (norm (plus A AND not very A))

```

(deftemplate Fever
  98.6 104
  ((fever (S 98.6 103))
   (very_mild_fever somewhat fever)
   (mild_fever more-or-less fever)
   (high_fever very fever)
   (very_high_fever extremely fever)
   (not_fever not fever)))

(defrule GetValue
  (declare (salience 100))
  =>

```

```
(printout t "Enter one of the following:" crlf)
(printout t "fever, mild_fever, very_mild_fever,
high_fever, very_high_fever, or not_fever? ")
(bind ?m (read))
(assert-string (format nil "(Fever %s)" ?m)))
```

; The following rule prints a graph corresponding to the
; fuzzy set chosen by the user.

```
(defrule ShowLevel
  ?f <- (Fever ?l)
  =>
  (plot-fuzzy-value t "*" nil nil ?f))
```