

Продукционные системы

Александр Туренко

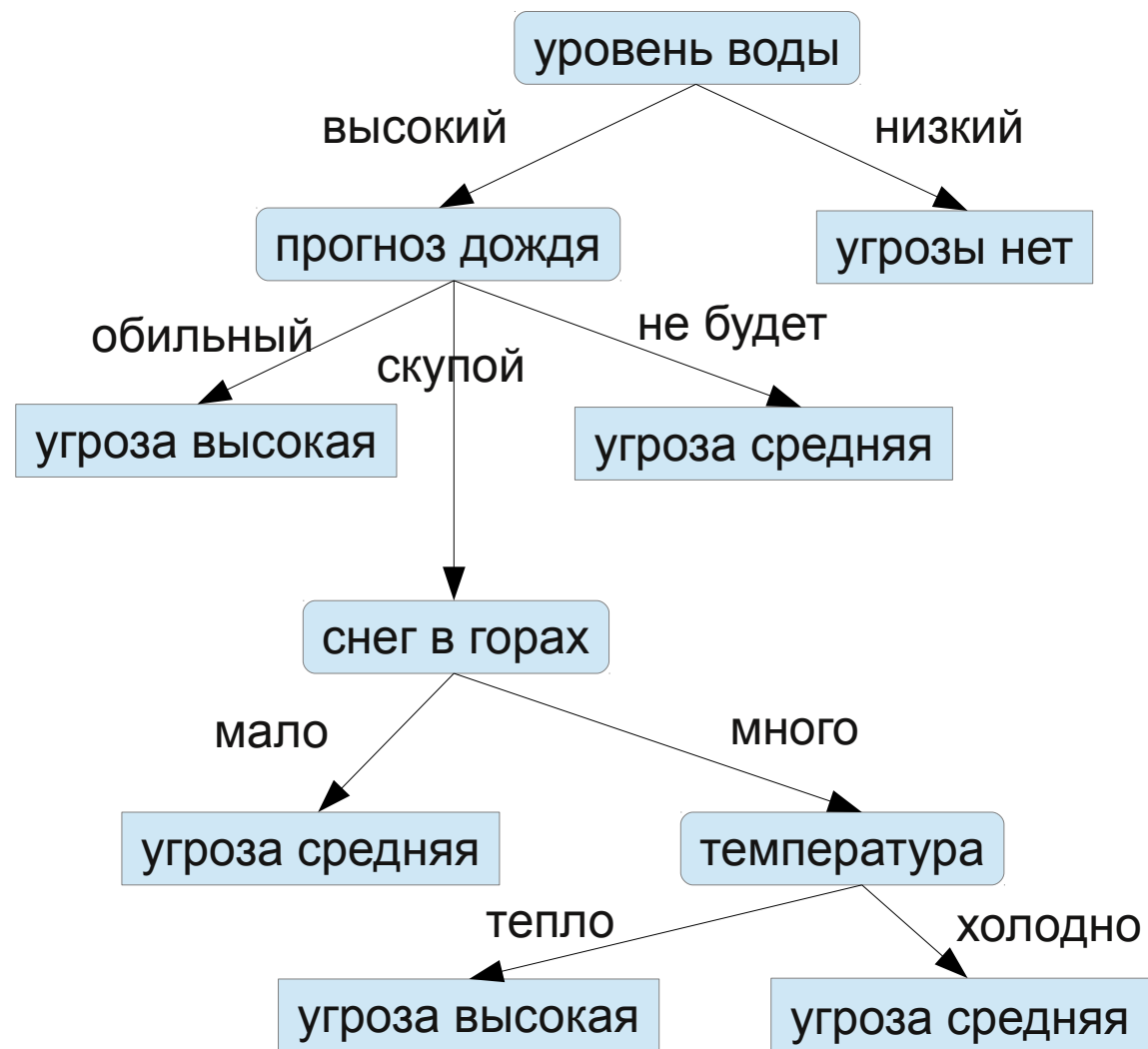
ВМК МГУ
8 ноября 2012

Продукции

- Продукция (продукционное правило) — правило вида ЕСЛИ <условия>, ТО <действия>.
- Примеры:
 1. «Если Вы сели в троллейбус, то нужно заплатить за проезд».
 - 2.«Если вода кипит, то часть воды должна превращаться в пар».
- Условия иногда называют посылками или предпосылками, а действия — заключениями.

Генерация продукций. Деревья решений.

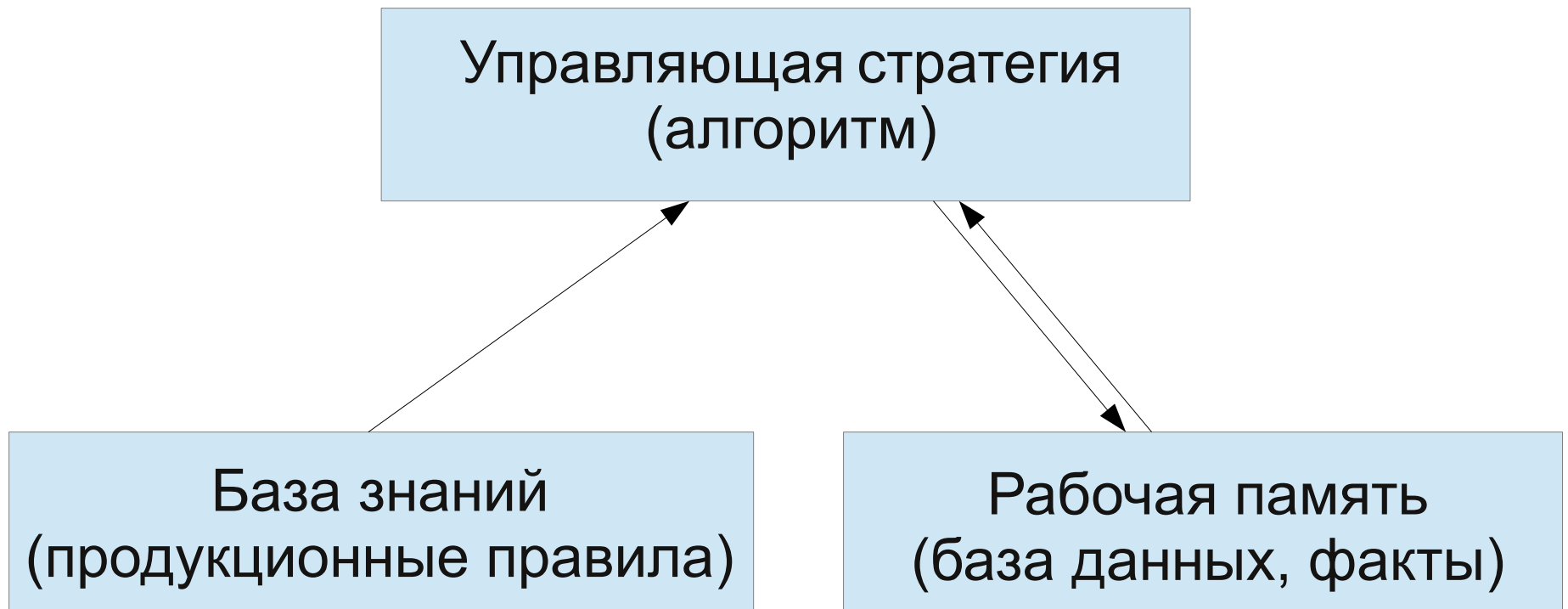
- В нелистовых вершинах: факторы, влияющие на принятие решения.
- На ветвях: значения факторов.
- В листовой вершине: вывод, определяющийся значениями факторов на пути к этому листу от корня.
- Количество генерируемых продукций — по числу листов.



Продукционная система

- Продукционная система — это модель вычислений, применяется в задачах создания (эффективных) алгоритмов поиска и моделирования человеческого мышления.
- Состоит из набора продукционных правил (продукций), рабочей памяти и управляющей стратегии.
- Аналогии и термины: долговременная память (навыки, база знаний), кратковременная память (база данных, набор фактов, состояние окружающего мира), алгоритм выбора хода (модель управления выводом).

Схема продукционной системы



Продукционные правила в системе

- Действие продукционного правила (продукции) направлено на изменение рабочей памяти.
- Применимость продукционного правила определяется текущим состоянием рабочей памяти.
- Управляющая стратегия — алгоритм выбора продукционного правила из конфликтного множества (т. е. из допустимых в данный момент).

Прямой вывод. Пример.

- Правила:

1. « $ba \rightarrow ab$ ».

2. « $ca \rightarrow ac$ ».

3. « $cb \rightarrow bc$ ».

- Рабочая память:

Строка из символов 'a', 'b' и 'c'.

- Управляющая стратегия:

Первое применимое.

Ит. №	Рабочая память	Применимы	Применяется
0	cbaca	1, 2, 3	1
1	cabca	2	2
2	acbca	2, 3	2
3	acbac	1, 3	1
4	acabc	2	2
5	aacbc	3	3
6	aabcc	∅	Останов

- Используется прямой вывод.

Прямой и обратный вывод

- Прямой вывод — от исходного состояния к целевому. «Выводим из известных фактов все возможные, пока не получим то, что требуется». Применимость правил — в соответствии с левыми частями.
- Обратный вывод — от целевого состояния к исходному. «Смотрим, что нужно доказать, чтобы цель была выполнена». Применимость правил — в соответствии с правыми частями.

Обратный вывод. Примеры.

- Предыдущий пример. Известно количество 'a', 'b', 'c' в исходном состоянии, значит целевое состояние известно. Путь к цели можно искать с помощью обратного вывода, поменяв левые и правые части правил, поменяв исходное и целевое состояния и применяя, например, ту же самую управляющую стратегию.

Путь, кстати, будет не тем, что при прямом выводе.

- Доказательство теорем в терминах логики предикатов первого порядка.

Управляющие стратегии

- Исчерпывающий перебор (поиск с возвратами, backtracking).
- Выбор одного правила с помощью оценки.
- Управление с помощью планировщика «классной доски».
- Управление с помощью метаправил.

Алгоритмы выбора правила

- Динамическая оценка в зависимости от вклада в достижение цели.
 - В зависимости от важности используемых фактов.
 - На основе статических приоритетов правил.
 - Наиболее специализированное (specificity) правило.
1. «Если является птицей и является пингвином, то не умеет летать».
 2. «Если является птицей, то умеет летать».

Алгоритмы выбора правила

- В соответствии с отношением частичного порядка правил.
- Правило, в котором задействован последний появившийся в рабочей памяти факт (стратегия новизны, recency, оппортунистический поиск). Позволяет сосредоточиться на одной линии рассуждения.
- Не применять правило, применённое последним, пока не изменятся факты, соответствующие его условиям (рефракция, refraction).

Общая последовательность действий

1. Нахождение применимых правил: соответствие левой части одному из фактов (прямой вывод), либо соответствие правой части одной из целей (обратный вывод). Если ни одно правило применить нельзя, то задача не имеет решения — завершаем работу.
2. Разрешение конфликтов. Применение одного из алгоритмов выбора правила.
3. В некоторых продукционных системах требуется удалить из рабочей памяти все противоречащие выбранному правилу.

Общая последовательность действий

4. Выполнение действий из правой части правила (прямой вывод), либо замена в рабочей памяти факта, соответствующего правой части правила, условиями из левой.
5. Проверка, есть ли целевой факт в рабочей памяти (прямой вывод), либо есть ли требующие доказательства факты (обратный вывод).
6. Если ответ «да», то переходим к шагу 1, иначе — завершаем работу (или, если продукционная система позволяет, возвращаемся к одному из предыдущих состояний и переходим к шагу 1).

Устаревающие факты

- Если выведенный факт противоречит некоторым другим фактам из рабочей памяти?
 1. Убираем из рабочей памяти факты, противоречащие выведенному.
 2. Добавляем выведенный факт.
- Такие ситуации возможны не для всех продукционных систем.
- Проблема отделения устаревших фактов от актуальных называется проблемой рамок и границ (frame problem).

Устаревающие факты. Пример.

- Система с правилами вида:

«Если конь на клетке X и есть ход конём в клетку Y, то убрать факт конь на X, добавить факт конь на Y, устранить противоречия».

Унификация

- Правила могут содержать переменные, например, «путь из X в Рим существует».
- Унификация — поиск таких значений переменных, при которых данное правило становится применимым.
- Если правило применимо к текущему значению рабочей памяти, то говорят, что рабочая память унифицируется с этим правилом.

«Классная доска». Примеры задач.

Программа распознавания речи:

1. Оцифрованная волна → фрагменты речи.
2. Фрагменты речи → отдельные слова.
3. Слова → предложения.
4. Предложение → представление семантики предложения.

«Классная доска». Примеры задач.

Задача слияния (объединения) показаний датчиков.

- Имеется сеть датчиков, соответствующих различным процессам.
- Процессы могут влиять друг на друга.
- Интерпретация данных каждого отдельного датчика зависит от данных, полученных другими датчиками этой единой сети.

«Классная доска»

- Что общего в предложенных задачах? Можно выделить факты и правила различных типов и проследить их взаимосвязь.
- Для решения такого рода задач применяется управляющая стратегия «классная доска» (blackboard).
- Факты (элементы рабочей памяти) имеют тип. Факты одного типа образуют т. н. модуль рабочей памяти.
- Продукционные правила имеют тип. Множество правил одного типа образует т. н. источник знаний.

«Классная доска»

- Условия правил некоторого типа PROD_T отождествляются (унифицируются) только с фактами типа FACT_T_IN, а действия правил этого типа порождают, удаляют, изменяют только факты типа FACT_T_OUT.
- Возможен вариант, когда действия влияют на некоторое определённое множество модулей рабочей памяти, а не один.

«Классная доска»

Вернёмся к задаче об интерпретации речи.

- Прослеживается чёткая иерархическая структура типов фактов и правил.



«Классная доска»

- Факты на одном уровне могут относиться к разным гипотезам. При этом гипотезы представлены фактами, находящимися на более высоких уровнях.
- Если одна из гипотез оказалась неверной (или малоперспективной), то рассматривать факты, относящиеся только к ней, нет смысла.
- Например, если цвет предмета «не красный», то бессмысленно искать его среди предметов красного цвета.

«Классная доска»

- Структура отношений между модулями рабочей памяти и подмножествами правил одного типа может быть сложнее линейной иерархии (как в примере с датчиками).
- Структура отношений определяет характер планирования заданий (алгоритм выбора правил).

«Классная доска». Преимущества подхода.

Последовательную проверку гипотез можно организовать с помощью поиска с возвратами. Тогда в чём преимущество разделения фактов и правил на типы?

- Для некоторых задач разделение на подзадачи (преобразования фактов одного типа в другой) естественно, задача в таком виде легче формализуется.
- Для каждого модуля рабочей памяти требуется сопоставление (унификация) с правилами только одного типа (а не со всеми).

«Классная доска». Преимущества подхода.

- Можно получать факты разных типов асинхронно (параллельно).
- Благодаря информации о типе фактов и правил, появляется больше возможностей для задания направления решения задачи. Направление определяется планировщиком заданий.

«Классная доска». Планировщик заданий.

- Задача: на основе данных (фактов) управлять потоком выполнения (выбором правил для применения), обеспечивая определённое направление решения задачи.
- Планировщик заданий может быть реализован набором правил отдельного типа.
- Для этого каждое правило в добавок к основному действию порождает факты определённого типа — сообщения о результатах обработки информации.
- На основе этих сообщений планировщик выбирает следующее задание.
- Такой планировщик реализован в системе понимания речи HEARSAY-II.

«Классная доска». Планировщик заданий.

- В HEARSAY-III планировщик заданий реализован как вторичная «классная доска».
- Таким образом процесс планирования разделён на подзадачи, по своей сути — наборы метаправил (о них — чуть дальше).

Метазнания

- Метазнание — знание о знаниях.
- При работе продукционной системе приходится решать некоторые нетривиальные проблемы, в частности:
 1. Формирование новых правил и сопутствующие проблемы (проверка набора правил на противоречивость и избыточность).
 2. Выбор правила из конфликтного множества (управляющая стратегия).

Метазнания

- Указанные проблемы можно решать в терминах продукционных систем. При этом решение определяется набором продукционных правил, называемых метазнаниями. В частных случаях (соответственно проблемам на предыдущем слайде) эти правила называют:
 1. Метазнание объектов окружающего мира.
 2. Метаправила (метазнания стратегий).

Метазнания

- С точки зрения работы первоначальной продукционной системы, система, использующая метазнания, — чёрный ящик.
- На входе: правила вывода, рабочая область, проблема. При необходимости можно сделать доступной информацию о ранее применённых правилах и предыдущих значениях рабочей области.
- На выходе: правила вывода и рабочая область (вообще говоря, изменённые), решение проблемы.

Метазнание объектов окружающего мира

- Формирование вопросов к эксперту для синтеза новых правил. Интерпретация ответов.
- Проверка добавляемых правил на противоречивость и избыточность.
- Разрешение противоречий среди правил (возможно, опять же, через формирование вопросов — интерпретацию ответов).

Метаправила

- Управляющая стратегия в форме правил выбора очередного правила вывода.
- Решаемые задачи:
 1. Исключение правил вывода, не подходящих к данной ситуации (состоянию рабочей памяти).
 2. Частичная классификация остальных правил, выбор наиболее перспективного пути.
- При большом числе фактов и правил — позволяет устранить комбинаторный взрыв.

Метазнания. Примеры.

- Задача упрощения символьного выражения, состоящего из:
 1. Переменных.
 2. Числовых констант.
 3. Операций умножения, деления, возведения в степень ('*', '/', '^').
- Подразумеваемое в правилах:
 1. Левую часть следует трактовать как «если выражение можно представить в виде...».
 2. X — переменная, отождествляемая с переменными выражения.
 3. M, N — переменные, отождествляемые с числовыми константами выражения.
 4. EXPR_0, EXPR_1 — переменные, отождествляемая с любым корректным подвыражением выражения.

Метазнания. Примеры.

- Правила:

1. Если $(X^M/X^N)*EXPR_1$, то $X^{(M-N)}*EXPR_1$.
2. Если $(X^0)*EXPR_1$, то $EXPR_1$ и $X \neq 0$.
3. Если $(EXPR_0^0)*EXPR_1$ и $EXPR_0$ не является очевидно нулевым, то $EXPR_1$ и $EXPR_0 \neq 0$.
4. Если $1*EXPR_1$, то $EXPR_1$.
5. Если $EXPR_0 / (1 / EXPR_1)$, то $EXPR_0 * EXPR_1$ и $EXPR_1 \neq 0$.

Метазнания. Примеры.

- Как можно улучшить работу системы?
- Если при данной унификации правило ведёт к добавлению некоторого нового ограничения, задать пользователю вопрос, можно ли считать данное ограничение удовлетворяющимся.
- Если ограничение не удовлетворяется, то считаем унифицированное данным образом правило, а также все правила, ведущие к добавлению данного ограничения, неприменимыми.
- Полученное от пользователя метазнание является знанием об окружающем мире и позволяет уточнить формулировку задачи.

Метазнания. Примеры.

- Как можно улучшить работу системы?
- Можно синтезировать новые правила, пользуясь, например, знанием о том, что, если условие имеет вид «Если $EXPR_0 * EXPR_1$ <другие условия>, <действия>», то справедливо «Если $EXPR_0 / (1 / EXPR_1)$, то <действия> $EXPR_1 \neq 0$ ».
- Это позволяет использовать правила 1-4 в случае, если происходит деление на $EXPR_1$, а не умножения.
- Знания такого рода в данном случае являются метазнаниями окружающего мира.

Метазнания. Примеры.

- Как можно улучшить работу системы?
- Пусть есть знания в духе:
 1. «Если применить правило 1 (упрощение X^M/X^N) с $M == N$, то может стать применимым правило 2 (упрощение X^0).».
 2. «Если применить правило 2, то может стать применимым правило 3 (удаление умножения на единицу).».
- Эти знания можно использовать для выбора очередности применения правила (сначала лучше выполнять правила из левых частей такого рода метаправил).
- Знания такого рода определяют выбор правила из конфликтного множества и являются метаправилами.

Метазнание объектов окружающего мира. Модели правил.

- Модель правила — краткое обобщённое описание подмножества правил или подмножества моделей правил более низкого уровня.
- Состоит из:
 1. Список всех известных примеров.
 2. Характерные черты.
 3. Список более общих моделей (родителей).
 4. Список менее общих моделей (наследников).
- Можно сказать, что модели правил — это механизм обобщения или иерархической классификации.

Метазнание объектов окружающего мира. Модели правил.

- Модели правил позволяют с некоторой степенью достоверности предугадывать свойства правил определённых типов.
- Информацию, содержащуюся в моделях правил, можно использовать при интерпретации ответов эксперта на вопросы и для проверки добавляемого в систему правила.
- Модели не статичны, они уточняются при добавлении новых правил и, в случае возникновения неоднозначностей, с помощью вопросов к эксперту о действительных свойствах добавляемого правила.

Метазнание объектов окружающего мира. Концепты.

- Понятие модели правила — из системы TEIRESIAS (программа — помощник в создании экспертных систем, ведёт диалог с экспертом на естественном языке).
- Существует похожая идея под названием «концепт» — из систем MECNO (механика) и MYCIN (диагностика заболеваний).
- Концепт содержит следующую информацию:
 1. Описание характеристик и структуры данных.
 2. Указатель на все известные примеры.
 3. Связи с другими концептами.
 4. Указатель на группу концептов того же семейства.

Недостатки продукционных систем

- Трудность формализации знаний в виде продукционных правил.
- Неудобства в работе со строго определённым синтаксическим представлением для правил (громоздкие условия).
- Неясность взаимных отношений между правилами и связанные с этим трудности программирования некоторых алгоритмов.

Преимущества продукционных систем

- Соответствие логической структуре поиска в пространстве состояний. Возможность проследить цепочку состояний и применявшиеся при поиске решения правила и метаправила.
- Возможность инкрементальной разработки с помощью добавления, удаления, изменения правил без необходимости изменения других частей продукционной системы и других правил.

Такую возможность обеспечивает отсутствие прямого (не через рабочую память) взаимодействия между правилами (синтаксическая независимость).

- Добавление новых правил не влечёт необходимость очистки рабочей памяти.

Преимущества продукционных систем

- Возможность правдоподобного моделирования решения задачи человеком.
- Доступность чтения правил и фактов как человеком (в том числе и не имеющим специальных знаний о продукционных системах), так и машиной.
- Возможность проверки базы знаний на противоречивость и избыточность.
- Возможности эвристического управления поиском.
- Эффективность в решении задач из отдельных областей. Достигается благодаря возможности влиять на общий ход поиска решения с помощью управляющей стратегии.

Среда CLIPS

- CLIPS — интегрированная с языком Си среда построения продукционных систем (C Language Integrated Production System).
- Управляющая стратегия — встроена.
- Достаточно задать правила и факты.
- Интерфейс пользователя — интерактивное консольное приложение.

Среда CLIPS

- Задание факта:
CLIPS> (assert (I think))
- Ответ среды (факт успешно записан):
==> f-1 (I think)
- Факт можно добавить в рабочую область (**assert**), удалить из неё (**retract**), изменить (**modify**), продублировать (**duplicate**).

Среда CLIPS

- Задание правила:

(defrule <имя правила>

<необязательный комментарий>

<необязательное объявление>

<предпосылка_1>

<предпосылка_2>

...

<предпосылка_n>

=>

<действие_1>

<действие_2>

...

<действие_n>

)

Среда CLIPS

- Примеры правил:

```
(defrule rule_1
  "Rene Descartes rule"
  (salience 42)
  (I think)
```

=>

```
(I am)
)
```

- **salience** — степень важности правила, целое из диапазона [-10000, 10000]. По умолчанию — 0.

Среда CLIPS

- Запуск интерпретатора:
CLIPS> (run)
- Очистка рабочей памяти:
CLIPS> (reset)

Среда CLIPS. Слоты.

- Слоты — способ параметризации фактов.
- Факт со слотами:

```
(assert  
  name  
  (slot_name_1 value_1)  
  (slot_name_2 value_2)  
  ...  
  (slot_name_n value_n)  
)
```

- Пример (англичанин живёт в красном доме):

```
CLIPS> (assert (nationality-house-color (nationality  
"Englishman") (house-color "red"))
```

Среда CLIPS. Слоты.

- Пример правила со слотами (в жёлтом доме курят Kools):

```
(defrule rule_2
```

```
  (nationality-house-color (nationality ?x) (house-color "yellow"))
```

```
=>
```

```
  (assert (nationality-smokes (nationality ?x) (smokes "Kools")))
```

```
  (printout t "Added fact: ?x smokes Kools" crlf)
```

```
)
```

- **printout** с первым параметром **t** выводит на стандартный поток вывода строку. **crlf** служит для перевода строки.

Среда CLIPS. Факты.

- Факты могут быть неупорядоченными и упорядоченными.
- Неупорядоченные факты или шаблоны не сохраняют порядок значений, но зато допускают наличие слотов. Благодаря последнему, множество одноимённых неупорядоченных фактов напоминает ассоциативный массив.
- Упорядоченные факты не допускают использование слотов, зато сохраняют порядок значений. Благодаря последнему множество одноимённых упорядоченных фактов напоминает обычный массив.

Среда CLIPS. Возможности.

- Факты: неупорядоченные и упорядоченные.
- Классы и объекты.
- Глобальные переменные.
- Представление знаний: эвристические и процедурные знания (функции, родовые функции, обработчики сообщений, модули).
- Объектно-ориентированные возможности (язык COOL — CLIPS Object-Oriented Language).

Источники

- Ж.-Л. Лорьер «Системы искусственного интеллекта». Стр. 379-387, 419-426, 435-448.
- Дж. Люгер «Искусственный интеллект: стратегии и методы решения сложных проблем». Стр. 196-215, 282-291.
- А. П. Частиков, Д. Л. Белов, Т. А. Гаврилова «Разработка экспертных систем. Среда CLIPS». Стр. 89-169.
- Ю. С. Корухова «Управление знаниями: Учебное пособие». Стр. 3-20.
- <http://vu.tuit.uz/?go=clectures/view/13/lecture/67>
(принцип «классной доски»).