



СПБГЭТУ «ЛЭТИ» ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

2.3 Логический вывод в системе CLIPS. Стратегии разрешения конфликтов.

Базовый цикл работы МЛВ в системе CLIPS:

1. Работа МЛВ останавливается, если достигнут предел активации правил или нет текущего фокуса. В противном случае, для выполнения выбирается верхнее правило агенды того модуля, которому принадлежит текущий фокус. Если в этой агенде нет правил, текущий фокус извлекается из стека фокусов и управление переходит к следующему модулю. Если стек фокусов пуст, то выполнение останавливается, в противном случае вновь выполняется шаг 1.

2. Выполняются операторы, содержащиеся в консеквенте выбранного правила. Использование в консеквенте правила функции `return` не может переместить текущий фокус из стека фокусов. Счетчик числа правил инкрементируется для контроля предела активации правил.

3. Результатом шага 2 может быть активация или деактивация правил. Активированные правила (у которых удовлетворяется условия в антецеденте) помещаются в агенду того модуля, в котором они определены. Размещение в агенде определяется значимостью (*salience*) правила и текущей стратегией разрешения конфликтов. Деактивированные правила удаляются из агенды.





4. При использовании динамических значимостей, значения значимостей всех правил, содержащихся в агенде переоцениваются. Цикл повторяется с шага 1.

Агенда представляет собой список правил, условия которых удовлетворяются, но которые еще не были выполнены. Каждый модуль имеет собственную агенду. При извлечении правил агенда работает аналогично стеку – первым выполняется правило, находящееся на вершине агенды. Когда правило становится активным (условия в его левой части удовлетворяются), оно помещается в агенду в соответствии со следующими правилами:

1. Вновь активируемые правила помещаются над всеми правилами с более низкой *значимостью* (*salience*) и ниже всех правил с более высокой значимостью.

2. Для определения места среди правил равной значимости используется текущая стратегия разрешения конфликта.

3. Если в результате добавления или удаления факта одновременно активизируются несколько правил и шаги 1 и 2 не позволяют выполнить упорядочение, то эти правила упорядочиваются между собой произвольно.

Значимость позволяет пользователю назначать правилу приоритет, который учитывается при его помещении в агенду. Первым выбирается и срабатывает правило с максимальной значимостью. Значимость может принимать целое значение в диапазоне от -10000 до +10000. По умолчанию ее значение равно 0. Для явного назначения правилу значимости используется оператор:

```
(declare <rule- salience>),
```

который добавляется в левую часть правила и размещается перед первым условным элементом, например:

```
(defrule r1
  (declare (salience 500))
  (fire test-1)
=>
  (printout t "Rule r1 firing." crlf))
```

Значение значимости может назначаться в один из трех моментов: при определении правила, при активизации правила и в каждом цикле выполнения





(последние два случая соответствуют *динамической значимости*). По умолчанию значение значимости назначается только при определении правила. Для изменения такого поведения может использоваться команда `set-saliense-evaluation`.

Стратегии разрешения конфликтов. В CLIPS поддерживается семь стратегий разрешения конфликтов: “вглубь” (`depth`), “вширь” (`breadth`), “простоты” (`simplicity`), “сложности” (`complexity`), `LEX`, `MEA` и *случайного выбора* (`random`).

Текущая стратегия может быть установлена, используя пункт меню “Execution/Options” оконного интерфейса, или командой `set-strategy`. При этом агенда переупорядочивается на основе новой стратегии. Синтаксис команды:

```
(set-strategy <strategy>),
```

где `<strategy> ::= depth | breadth | simplicity | complexity | lex | mea | random`.

По умолчанию используется стратегия `depth`.

Стратегия “вглубь”. Вновь активируемые правила помещаются в агенду над всеми правилами такой же значимости. Например, пусть факт `f-1` активирует правила `rule-1` и `rule-2`, а факт `f-2` активирует правила `rule-3` и `rule-4`. Тогда если `f-1` устанавливается раньше, чем `f-2`, то `rule-3` и `rule-4` окажутся в агенде выше правил `rule-1` и `rule-2`. Однако положение правила `rule-1` относительно правила `rule-2` и правила `rule-3` относительно правила `rule-4` будет произвольным.

Стратегия “вширь”. Вновь активируемые правила помещаются ниже всех правил с такой же значимостью. Например, пусть факт `f-1` активирует правила `rule-1` и `rule-2`, а факт `f-2` активирует правила `rule-3` и `rule-4`. Тогда, если `f-1` устанавливается раньше, чем `f-2`, то `rule-1` и `rule-2` окажутся в агенде выше правил `rule-3` и `rule-4`. Однако, положение правила `rule-1` относительно правила `rule-2` и правила `rule-3` относительно правила `rule-4` будет произвольным.

Стратегия “простоты”. Среди правил одинаковой значимости, вновь активируемые правила помещаются над всеми правилами с равной или большей специфичностью (`specificity`). *Специфичность* правила определяется числом сравнений, которые должны быть выполнены в левой части правила.

Специфичность инкрементируется при:





- каждом сравнении с константой или предварительно связанной переменной;
- каждом вызове функции, сделанном из левой части правила в условном элементе с предикатным ограничением (:), ограничением возвращаемым значением (=) или УЭ-проверкой (test).

Булевы функции “и”, “или”, “не” не увеличивают специфичность правила, но их аргументы увеличивают. Вызовы функций, выполняемые из функций, не увеличивают специфичность. Например, следующее правило:

```
(defrule example
  (item ?x ?y ?x)
  (test (and(numberp ?x) (> ?x (+ 10 ?y)) (< ?x 100))))
=>...
```

имеет специфичность 5 (считаются операторы (item ?x ?y ?x), ?x, numberp, >, <).

Стратегия “сложности”. Среди правил одинаковой значимости, вновь активируемые правила помещаются над всеми правилами с равной или меньшей специфичностью.

Стратегия LEX. Для определения места правила в агенде среди правил одинаковой значимости в первую очередь используется *новизна* образцов, активирующих данное правило. Каждый факт и экземпляр в БД помечаются “временным тегом”, указывающим его новизну по отношению ко всем другим фактам и экземплярам в системе. Для определения местоположения правила в агенде образцы (факты или экземпляры), связанные с активацией каждого правила сортируются по убыванию новизны. Правило с более поздним образцом помещается выше правил с более ранними образцами. Чтобы определить относительный порядок размещения двух правил, отсортированные временные теги образцов, активирующих правила, сравниваются попарно, начиная с самых больших значений. Сравнение продолжается до тех пор, пока не будет обнаружено, что временной тег одной активации больше соответствующего временного тега другой активации. Правило с большим значением временного тега помещается в агенду выше другого правила. Данная стратегия поясняется примером на рис. 16.1. В данном случае раньше сработает правило Rule1, т.к. временной тег образца, связанного с его





третьим условным элементом (8) больше, чем временной тег соответствующего образца у правила Rule2 (6).

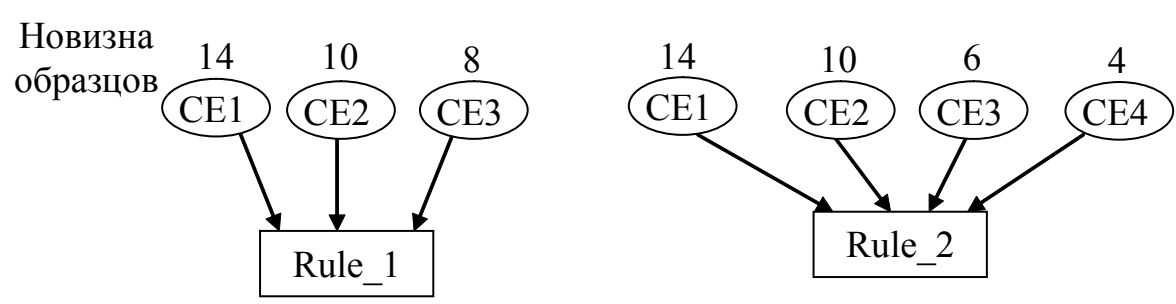


Рис. 16.1. Сравнение правил в стратегии разрешения конфликтов LEX

Если одно правило имеет больше образцов, чем другое, а все сравниваемые временные теги идентичны, то правило с большим числом временных тегов помещается выше. Если два правила имеют равную новизну, правило с более высокой специфичностью помещается выше правила с более низкой специфичностью.

Стратегия MEA. Для определения места правила в агенде среди правил равной значимости в первую очередь используется временной тег образца, связанного с первым условным элементом в правиле. Правило, у которого этот временной тег больше временных тегов, связанных с первыми условными элементами других правил, помещается в агенду выше них. Если временные теги первых образцов равны, то для определения места правила используется стратегия LEX. В примере на рис. 16.2 раньше сработает правило Rule1.



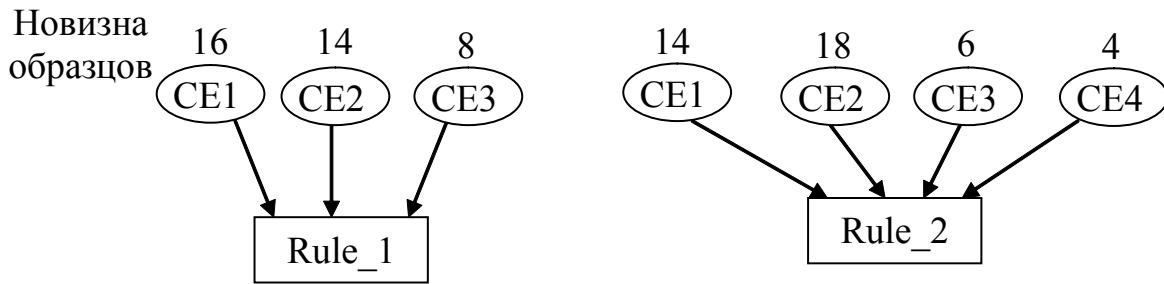


Рис. 16.2 Сравнение правил в стратегии разрешения конфликтов МЕА


Стратегия случайного выбора. Каждому правилу сопоставляется случайное число, которое используется для определения его местоположения в агенде среди правил равной значимости. Это случайное число сохраняется при изменении стратегии, а в случае возврата к случайной стратегии разрешения конфликта восстанавливается тот же порядок среди правил, которые находились в агенде, когда стратегия была изменена.

Пример построения ЭС в среде CLIPS

Ниже в качестве примера приведен фрагмент простой ЭС в среде CLIPS, определяющей наличие у студента свободного времени в зависимости от количества несделанных лабораторных работ и оставшегося до зачета времени:

```
(defrule data-input
  (initial-fact)
=>
  (printout t crlf "Введите число дней до зачета (целое
значение): ")
  (bind ?days (read))
  (if (numberp ?days)
  then (assert (days ?days))
  else (printout t "Введите число" crlf))
  (printout t crlf "Введите число несделанных
лабораторных работ (в %)")
```





```
(bind ?works (read))
(assert (works ?works)))
(defrule R1
(days ?days)
(works ?works)
(test (and (= ?days 1) (<> ?works 0)))
=>
(printout t crlf "Свободного времени нет" crlf)
(assert (freetime "no")))

(defrule R2
(days ?days)
(works ?works)
(test (and (= ?days 2) (>= ?works 10)))
=>
(printout t crlf crlf "Свободного времени нет" crlf)
(assert (freetime "no")))
```

Правило `data-input` обеспечивает ввод исходных значений числа оставшихся до зачета дней и количества неделанных работ. Оно иницируется исходным фактом (`initial-fact`) в рабочей памяти и выдает приглашение на ввод значения. Команда `bind` осуществляет занесение введенных значений в факты рабочей памяти. Кроме того, для переменной выполняется контроль типа введенного значения.