

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Локтионова Оксана Геннадьевна

Должность: проректор по учебной работе

Дата подписания: 16.06.2020

Уникальный программный ключ:

0b817ca911e6668abb13a5d426d39e5f1c11eabb75e945d14a4851fda56d089

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра программной инженерии

УТВЕРЖДАЮ

Проректор по учебной работе

О.Г. Локтионова

20__ г.



РАЗРАБОТКА МОДЕЛЕЙ ОБРАБОТКИ ЗНАНИЙ:

методические указания к лабораторным занятиям для магистров
направления подготовки 02.04.03 Математическое обеспечение и ад-
министрирование информационных систем

Курск 20__

УДК 004

Составители: Е.А. Титенко

Рецензент

Кандидат технических наук *А. В. Киселев*

Разработка моделей обработки знаний: методические указания к лабораторным занятиям / Минобрнауки России, Юго-Зап. гос. ун-т; сост.: Е.А. Титенко. - Курск, 2022. - 20 с. - Библиогр.: с. 20.

Приводится описание моделей обработки знаний и выявления новых зависимостей в рамках продукционной парадигмы вычислений. Приведены теоретические положения цикла вывода, основные правила работы модулей, практические примеры и задания.

Методические рекомендации предназначены для студентов, обучающихся по направлению подготовки 02.04.03 Математическое обеспечение и администрирование информационных систем очной и заочной форм обучения.

Текст печатается в авторской редакции.

Подписано в печать . Формат 60x84 1/16.

Усл.печ. л. 1,16 п.л . Уч.-изд. л. 1,05 . Тираж 100 экз. Заказ. *У*

Бесплатно.

Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

Цель лабораторной работы:

Изучение и применение моделей приобретения и обработки знаний в интеллектуальных системах, приобретение практических навыков организации вычислений и строения базы знаний интеллектуальной системы.

1 Основные понятия

1.1 Модели знаний и способы пополнения знаний

Основными компонентами экспертной системы являются (рисунок 1):

- база знаний (БЗ) продукционных правил;
- рабочая память, содержащая промежуточные и конечные данные;
- цикл управления распознавание-действие (машина вывода).

Моделирование решения задачи основано на процессе *сопоставления с образцом* (pattern matching) или *поиска по шаблону*, в ходе которого текущее состояние решения сравнивается с имеющимися знаниями для определения дальнейших действий. Для обеспечения оперативности *сопоставления с образцом* (pattern matching) или *поиска по шаблону* применяется ассоциативная память, осуществляющая коллективный (параллельный) доступ к множеству сравниваемых образцов или шаблонов.

В БЗ содержится множество *продукционных правил* или просто *продукций* (productions), являющихся парами условие-действие, которые определяют элементарные шаги решения задачи. *Условная часть* (IF-part) правила является шаблоном (образцом), по которому можно определить, в какой момент необходимо использовать (активировать) данное правило для выполнения очередного этапа решения задачи. *Часть действия* (THEN-part) описывает соответствующий шаг решения. Условную часть правила также называют *антецедентом* (antecedent), а часть действия — *консеквентом* (consequent).

Рабочая память (working memory) содержит текущее описание модели мира в процессе рассуждений. В этой модели содержится набор образцов, инициализируемый начальным описанием задачи.

В управляющем цикле *распознавание-действие* осуществляется сравнение образцов из рабочей памяти с условными частями правил в БЗ. Если условие какого-либо правила соответствует образцу, то это правило помещается в *конфликтное множество* (conflict set). Продукции, содержащиеся в конфликтном множестве, называют *допустимыми*, так как они согласованы с текущим состоянием рабочей памяти.

После того, как закончит работу цикл распознавание-действие, осуществляется процесс *разрешения конфликтов* (conflict resolution), в ходе которого выбирается и активизируется (возбуждается) одна из допустимых продукций или фиксированное множество продукций, допускающих параллельное срабатывание. Наконец, в соответствии с частью действия активированного правила осуществляется модификация рабочей памяти. Весь этот процесс повторяется до тех пор, пока образцы в рабочей памяти не будут.

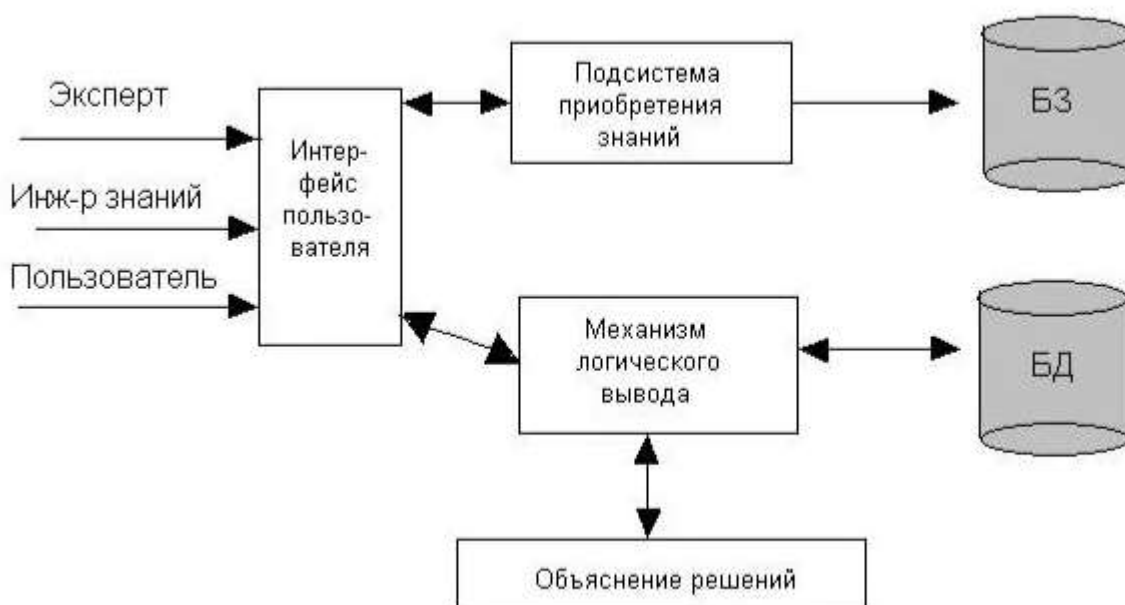


Рисунок 1 - Архитектура экспертной системы

Стратегии разрешения конфликтов отличаются в различных реализациях экспертных систем, они могут быть достаточно простыми. Например, выбирается первое из допустимых правил.

Однако многие системы допускают использование сложных эвристик (алгоритмов) для осуществления выбора из конфликтного

множества. Например, в системе OPS5 поддерживаются следующие стратегии разрешения конфликтов:

- **рефракция** (refraction) для предотвращения заикливания: после

активизации правила оно не может быть использовано снова, пока не изменится содержимое рабочей памяти.

- **новизна** (recency) позволяет сосредоточить поиск на одной линии рассуждения: предпочтение отдается правилам, в условиях которых встречаются факты, добавленные в рабочую память последними.

- **специфичность** (specificity) отдает предпочтение более конкретным правилам перед более общими: одно правило более специфично (конкретно), чем другое, если оно содержит больше фактов в условной части.

Чистая экспертная система не предусматривает выхода из тупиковых ситуаций в процессе поиска. Работа продолжается, пока не исчерпаны все возможные варианты. Гораздо более эффективным является модификация цикла управления с механизмом *возврата* (backtracking) в предыдущее состояние модели мира, позволяющего находить более точные решения задачи.

Продукционная модель позволяет имитировать различные аспекты поведения человека при решении задач. БЗ продукций соответствует навыкам решения задач в *долговременной памяти*. Подобно знаниям в долговременной памяти продукционные правила не изменяются в процессе решения задачи. Новые навыки могут быть просто добавлены в БЗ при необходимости. Рабочая память соответствует *кратковременной памяти*. В ходе решения задачи «фокус внимания» переходит от одного шага к другому, при этом после получения решения содержимое рабочей памяти не сохраняется.

1.2 Структура фактов и правил базы знаний

В качестве *условия* и *действия* в правилах может быть, например, предположение о наличии того или иного свойства, принимающее значение *истина* или *ложь*. При этом термин *действие* следует трактовать широко: это может быть директива к выполнению какой-либо операции, рекомендация, или модификация базы знаний - предположение о наличии какого-либо производного свойства.

Примером продукции может служить следующее выражение:

П1: *ЕСЛИ* клиент работает на одном месте более двух лет,
' *ТО* клиент имеет постоянную работу.

Как условие, так и действие правила могут учитывать несколько выражений, объединенных логическими связками И, ИЛИ, НЕ:

П2: *ЕСЛИ* клиент имеет постоянную работу *И*
клиенту более 18 лет *И* клиент *НЕ* имеет
финансовых обязательств,
ТО клиент может претендовать на получение кредита.

Помимо продукционных правил база знаний экспертная система должна включать и простые *факты*, поступающие в систему через интерфейс пользователя или выводимые в процессе поиска решения задачи. Факты являются простыми утверждениями типа «клиент работает на одном месте более двух лет». Факты, являясь истинными утверждениями, копируются в рабочую память для использования в цикле распознавание-действие.

Последовательное активирование правил порождает *цепочку вывода*. Цепочка вывода, полученная в результате последовательного выполнения правил П1 и П2 показана ниже (рисунок 2.). Эта цепочка показывает, как на основании правил и исходных фактов выводит заключение о возможности получения кредита. В ходе первой итерации вывода управляющий цикл, сопоставляя факт «клиент работает на одном месте более двух лет» и правило П1 добавляет в рабочую память новый факт (образец) «клиент имеет постоянную работу». Далее на основании содержимого рабочей памяти и правила П2 выводится новый факт «клиент может претендовать на получение кредита», который и является окончательным решением, так как цикл распознавание-действие останавливается и процесс вывода на этом завершается.

Монотонным выводом в продукционных системах называют вывод, при котором факты не удаляются из рабочей памяти. *Немонотонный вывод* допускает удаление фактов из рабочей памяти. При немонотонном выводе существенную роль играет порядок применения продукционных правил.

Цепочки вывода экспертной системы могут быть предъявлены пользователю и помогают понять, как было получено решение. Также визуализация этого процесса может быть полезна в процессе отладки БЗ, так как эксперт получает возможность «увидеть» неправильный ход рассуждений. Большинство реализаций продукционных систем имеют встроенные средства для интроспекции БЗ и мониторинга работы машины вывода.

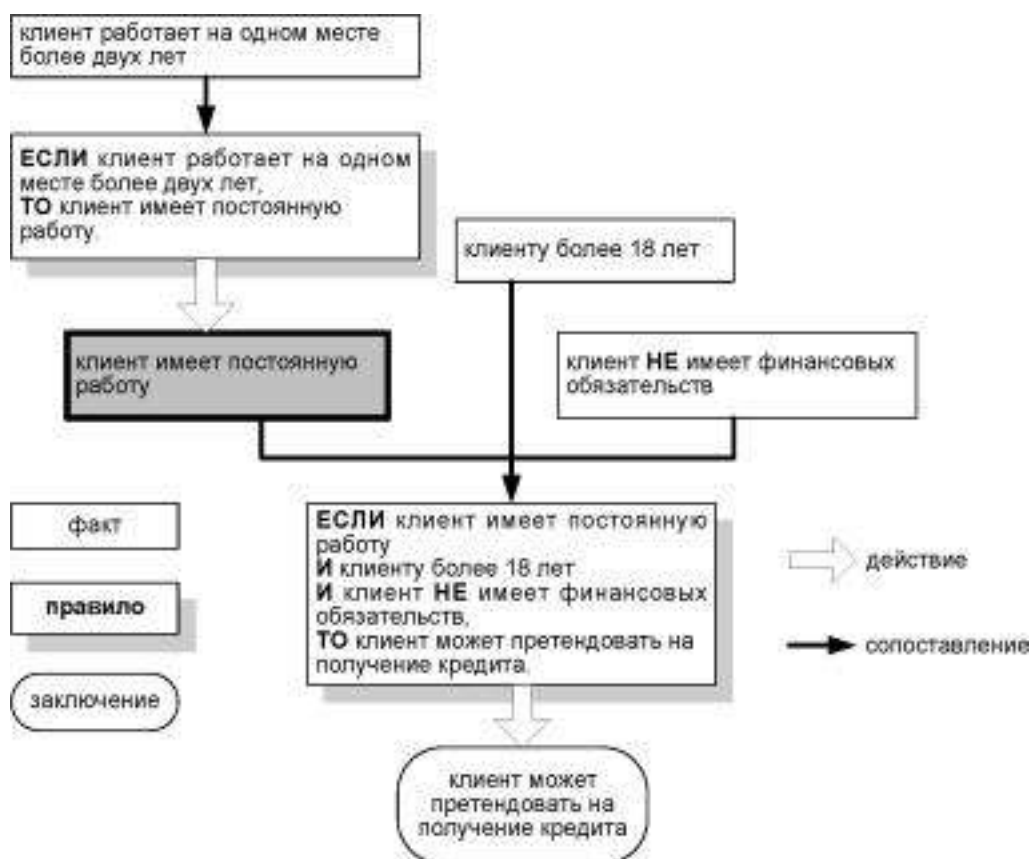


Рисунок 2 - Пример цепочки вывода в продукционной системе

Первой задачей в разработке базы знаний, является создание модели, описывающей факты (объекты, в терминах объектно-ориентированного подхода), которые участвуют в описании бизнес-правил.

В нашем примере определим два факта: заявление на кредит, решение о выдаче кредита. А так как для наименования фактов необходимо использовать только латинский алфавит, то для каждого факта придумаем название: заявление на кредит - ApplicationForCredit и решение о выдаче кредита - CreditDecision.

ApplicationForCredit будет содержать следующие поля:

- Сумма кредита (AmountOfCredit) - сумма в рублях, которую запрашивает заемщик;
- Срок кредитования (PeriodOfCredit) - срок в месяцах, на который заемщик запрашивает кредит;
- Ежемесячный доход (Salary) - заработная плата заемщика в рублях;
- Возраст (Age) - возраст заемщика в годах;
- Пол (Sex) - пол заемщика, М или Ж;
- Опыт работы (JobExperience) - совокупный опыт работы заемщика в годах;
- Последний срок работы (LastPeriodOfWork) - срок работы заемщика на последнем рабочем месте в месяцах;
- Сумма текущих обязательств (CurrentObligations) - сумма в рублях, которую заемщик выплачивает ежемесячно по другим кредитам;

CreditDecision будет содержать следующие поля:

- Ответ (Decision) - решение банка: «отказать в кредите» или «выдать кредит» (значение по умолчанию);
- Ежемесячная плата (MonthlyFee) - ежемесячная плата по кредиту в рублях;

Создание модели

Для того чтобы создать модель выделим пакет, который будет содержать все бизнес-правила, модель и тестовые сценария для базы знаний, назовём его пакет «CreditKB» нажимаем «Create New» и выбираем «New Declarative Model». Далее в открывшемся окне вводим имя модели, выбираем пакет (рисунок 3).

Создание факта

После создания модели откроется вкладка, в которой нажимаем «Add new fact type», вводим имя факта и жмем «ОК». Повторим это действие для ApplicationForCredit и CreditDecision, тогда в модели должно появиться два факта как на рис. 3.



Рисунок 3 - Список фактов в системе Model

Создание поля

Когда факт создан, мы можем приступить к созданию полей этого факта, для этого нажимаем «Add field» и в открывшемся окне вводим имя поля («Field name») и его тип («Type»). True or False (boolean) - булево значение, истина или ложно, После того как создаются поля обоих фактов, должен получиться такой же список как на рисунок 4.

Типы полей:

- Whole number (integer) - целое число,
- Text (string) - текст,
- Date (java.util.Date) - время и дата,
- Decimal number (java.util.BigDecimal) - число с плавающей точкой.

После создания структуры и ввода значений необходимо сохранить модель.

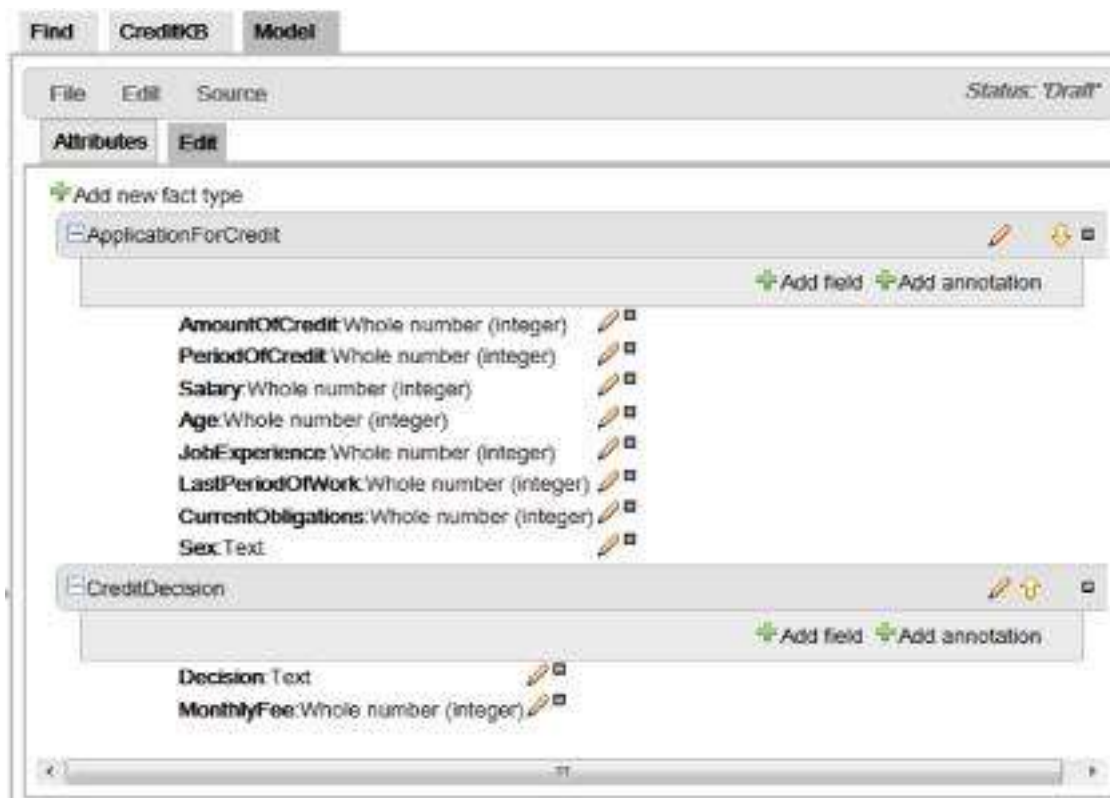


Рисунок 4 – Создание полей объекта

База знаний будет состоять всего из пяти правил (это очень мало для реальной базы знаний, но для нашего примера этого количества достаточно).

Создание правила

Первое правило, которое мы опишем, будет следующим:

ЕСЛИ ВОЗРАСТ ЗАЕМЩИКА < 18 ЛЕТ ТО ОТКАЗАТЬ В КРЕДИТЕ.

Для этого создается пустое правило: (рисунок 5).

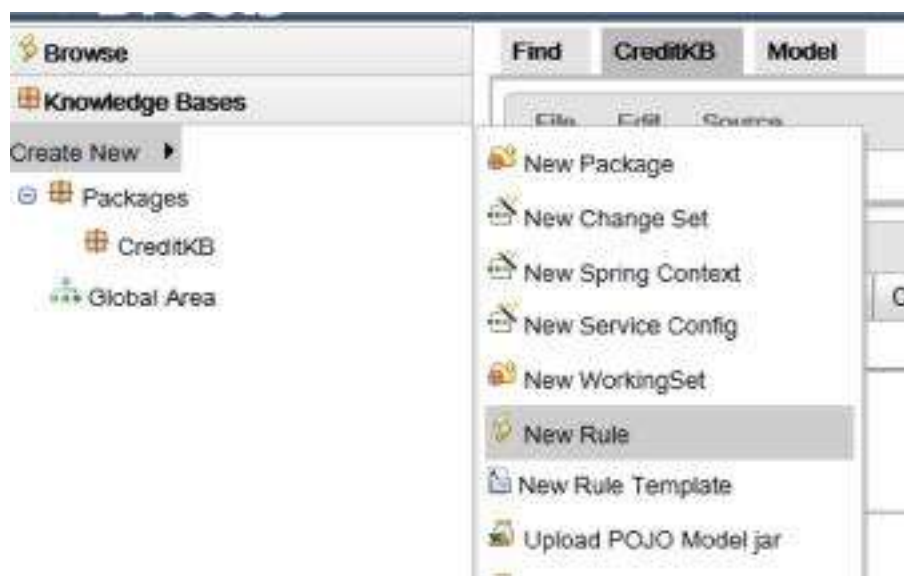


Рисунок 5 - Создание нового правила

В диалоговом окне вводится имя правила и его описание (рисунок 6).

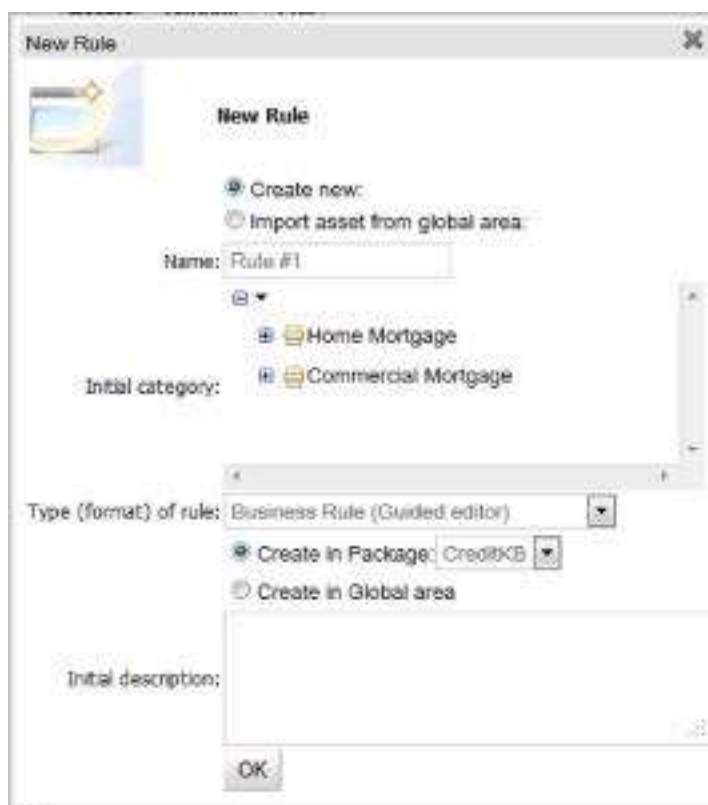


Рисунок 6 - Окно создания правила

Далее осуществляется заполнение частей правила - «условия» и «действия», в терминах экспертной системы Drools Guvnor - это When и Then части правила соответственно.

Описание части условий

Для того чтобы описать условие «возраст заемщика меньше 18 лет» необходимо выполнить следующие действия: нажать + напротив слова «WHEN» и в открывшемся окне выбрать ApplicationForCredit, после чего у вас должно появиться условие как на рисунок 7.

Далее надо добавить условие «возраст меньше 18 лет» для этого необходимо добавить ограничение на поле факта ApplicationForCredit: нажимаем на название только что добавленного условия и в появившемся окне выбираем в поле «Add a restriction on a field» поле факта Age. Далее в появившемся окне выбирается свойство «Literal Value», после чего вводится константное значение - 18, см. рис. 18. Теперь описание условия «возраст заемщика меньше 18 лет» закончено.

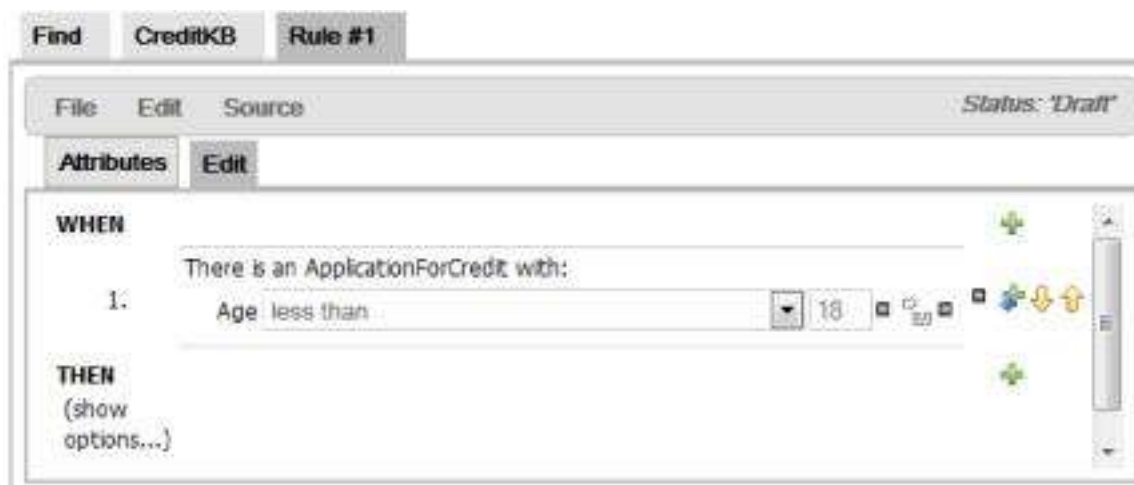


Рисунок 7 - Добавленное ограничение на поле Age

При срабатывании этого правила необходимо указать, что выдача кредита отказана, т.е. записать в поле Decision факта CreditDecision слова «отказать в кредите», а что бы это реализовать, сначала необходимо добавить в часть WHEN условие наличия факта CreditDecision, в котором будет записан текст «отказать в кредите» при срабатывании правила, и присвоить его переменной, назовем её «decision».

После всех действий правило должно выглядеть так, как показано на рисунок 8.

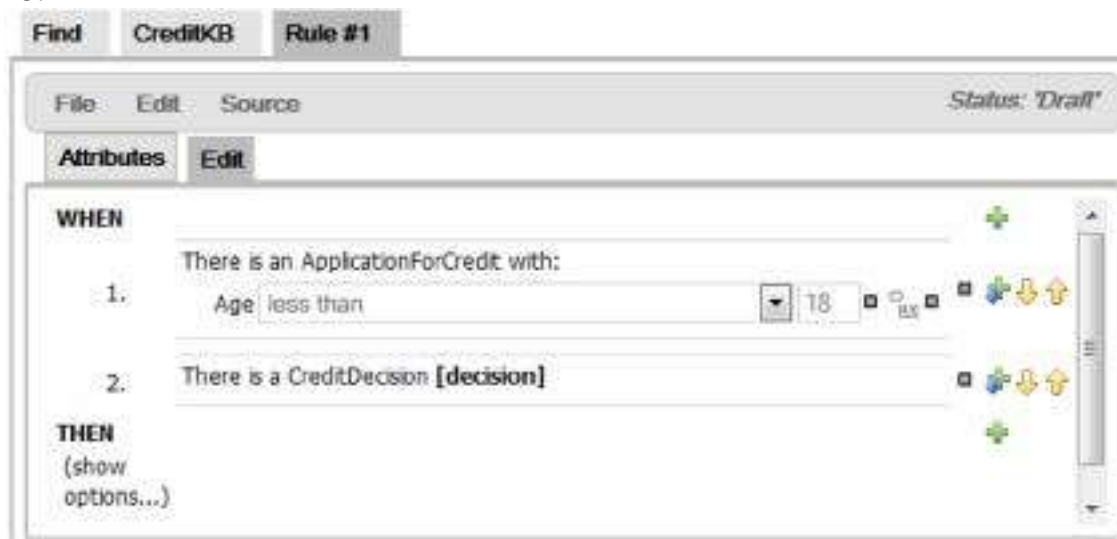


Рисунок 8 - Присвоение факта переменной

Описание части действий правила

Действие заключается в записи в поле Decision факта CreditDecision, который присвоен переменной decision, текст «отказать в кредите».

Действия: нажимаем значок плюса в части THEN и в появившемся окне выбираем «Change field values of decision», где «decision» - это имя созданной нами переменной, и нажимаем «ОК», в результате чего появляется рисунок карандаша. Далее по нажатию на появившийся рисунок карандаша,

открывается окно, в котором необходимо выбрать в списке значение «Decision», которое является именем одноименного поля факта. И снова нажимаем на рисунок карандаша, выбираем «Literal Value» и вводим текст «отказать в кредите». В результате выполненных построений правило выглядит как на рисунок 9.



Рисунок 10 – Завершенное правило

Для проверки правила выполняется его валидация с помощью команды «Validate» и «Verify».

1.3 Стратегии параллельных выводов

Далее рассматриваются следующие параллельные стратегии выводов: И-ИЛИ-параллельный продукционный вывод, равноправный вывод.

Под стратегией И-ИЛИ-параллельного продукционного вывода условимся понимать следующие способы инициализации продукций:

1. неконфликтующие продукции независимо применимы к обрабатываемому слову с минимальных позиций независимых вхождений их образцов;
2. конфликтующие продукции равноправно применимы к обрабатываемому слову по его копиям;
3. отсутствие вхождений всех образцов в обрабатываемое слово является признаком прекращения процесса вывода.

Равноправное срабатывание конфликтующих продукций и независимое срабатывание неконфликтующих продукций достигается за счет создания необходимого числа копий данных для альтернативных и независимых путей дерева вывода.

Под стратегией равноправного параллельного вывода условимся понимать следующие способы инициализации продукций

1. все продукции равноправно применимы к копиям обрабатываемого слова S с минимальных позиций вхождений их образцов;

2. результат однократного срабатывания продукции - мультислово $\{S\}$, получаемое соединением слов-результатов от срабатывания всех продукций без очередности их записи в $\{S\}$;

3. отсутствие вхождений всех образцов в слово S_i , входящее в мультислово $\{S\}$, является признаком прекращения процесса вывода по слову S_i , которое исключается из $\{S\}$ как полученное решение.

4. отсутствие вхождений всех образцов в мультислово $\{S\}$, является признаком прекращения процесса вывода с выделением $\{S\}$ как полученных решений.

Таким образом, выводы по всем альтернативным путям независимы и осуществляются самостоятельно без использования списка конфликтных слов. Такая самостоятельность, достигаемая копированием обрабатываемых слов на входы всех продукций, обуславливает независимые вхождения образцов по различным фрагментам копии S или $\{S\}$. Следовательно, процессы поиска и подстановки по каждой продукции протекают независимо во времени и распределенно по длине копии S или $\{S\}$.

Особенность стратегии равноправных выводов – симметричный вид продукции, имеющей отличительную семантическую трактовку.

$$\lambda \leftarrow 0 \rightarrow P.$$

Правая часть продукции описывает ее стандартное срабатывание, т.е. замену слова-образца на слово-подстановку с минимальной позиции вхождения O в обрабатываемое слово S . Левая часть реализует аннулирующую подстановку, необходимую для устранения возвратных перемещений по альтернативным вершинам дерева вывода и состоящую в аннуляции копии обрабатываемого слова.

На рисунке 11 приведен пример одного шага работы пятипродукционной системы со стратегии равноправных выводов с аннулирующим срабатыванием четвертой и пятой продукций и формированием мультислова $\{S\}=S_1\#S_2\#S_3$.

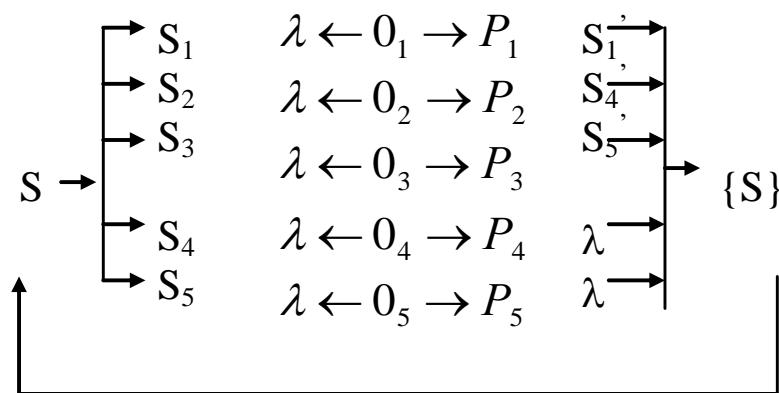


Рисунок 11- Шаг работы стратегии равноправных выводов,

где S_1, S_2, S_3, S_4, S_5 – копии обрабатываемого слова S ; $\{S\}$ – результирующее мультислово.

2 Пример реализации стратегий выводов

Пусть задано продукционное исчисление B , в котором все продукции являются активными:

$$B = \begin{cases} 1. abs \rightarrow mpk \\ 2. bsd \rightarrow pka \\ 3. ab \rightarrow dlb \\ 4. def \rightarrow bs \\ 5. ad \rightarrow lmp \end{cases}$$

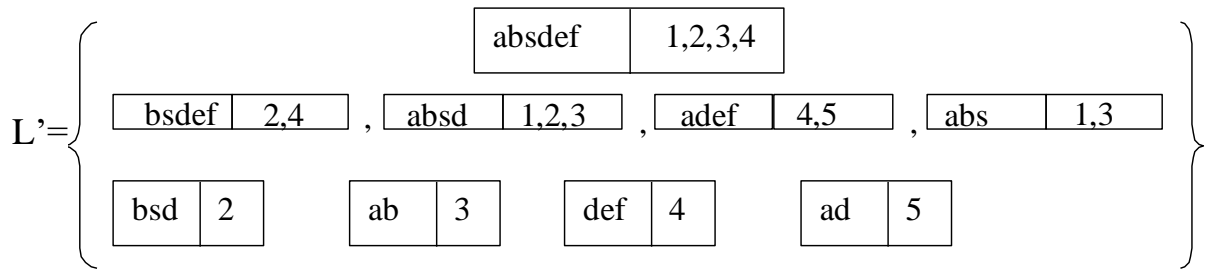
Определим список конфликтных слов и на его основе синтезируем расширенный список вхождений L'

$$K = \begin{cases} \{ab, abs\} = ab \\ \{abs, bsd\} = bs \\ \{def, ad\} = d \\ \{bsd, ab\} = b \\ \{bsd, def\} = d \end{cases} \quad K = \begin{cases} K_1^1 = abs \quad (1,3) \\ K_2^1 = absd \quad (1,2) \\ K_3^1 = adef \quad (4,5) \\ K_4^1 = absd \quad (2,3) \\ K_5^1 = bsdef \quad (2,4) \end{cases} \quad K = \begin{cases} K_1^1 = abs \quad (1,3) \\ K_2^1 = absd \quad (1,2,3) \\ K_3^1 = adef \quad (4,5) \\ K_4^1 = bsdef \quad (2,4) \end{cases}$$

Анализ конфликтных слов 1-ого уровня в соответствии с (1^a) выявляет ситуации их попарного пересечения, что обуславливает добавление конфликтных слов второго уровня.

$$K = \begin{cases} \{abs, absd\} = abs \\ \{abs, bsdef\} = bs \\ \{absd, bsdef\} = bsd \end{cases} \quad K = \begin{cases} K_1^1 = abs \quad (1,3) \\ K_2^1 = absd \quad (1,2,3) \\ K_3^1 = adef \quad (4,5) \\ K_4^1 = bsdef \quad (2,4) \\ K_1^2 = absd \quad (1,2,3) \\ K_2^2 = absdef \quad (1,2,3,4) \\ K_3^2 = absdef \quad (1,2,3,4) \end{cases} \quad K = \begin{cases} K_1^1 = abs \quad (1,3) \\ K_2^1 = absd \quad (1,2,3) \\ K_3^1 = adef \quad (4,5) \\ K_4^1 = bsdef \quad (2,4) \\ K_1^2 = absdef \quad (1,2,3,4) \end{cases}$$

Расширенный список L' будет иметь следующий вид, при этом конфликтное слово $abs(1,3)$ в нем предшествует равному по длине слову $bsd(2)$



Пусть обрабатываемое слово имеет вид $S = \text{adefdefabs}$.

Рассматривается И-ИЛИ- параллельный продукционный вывод (рис. 12).

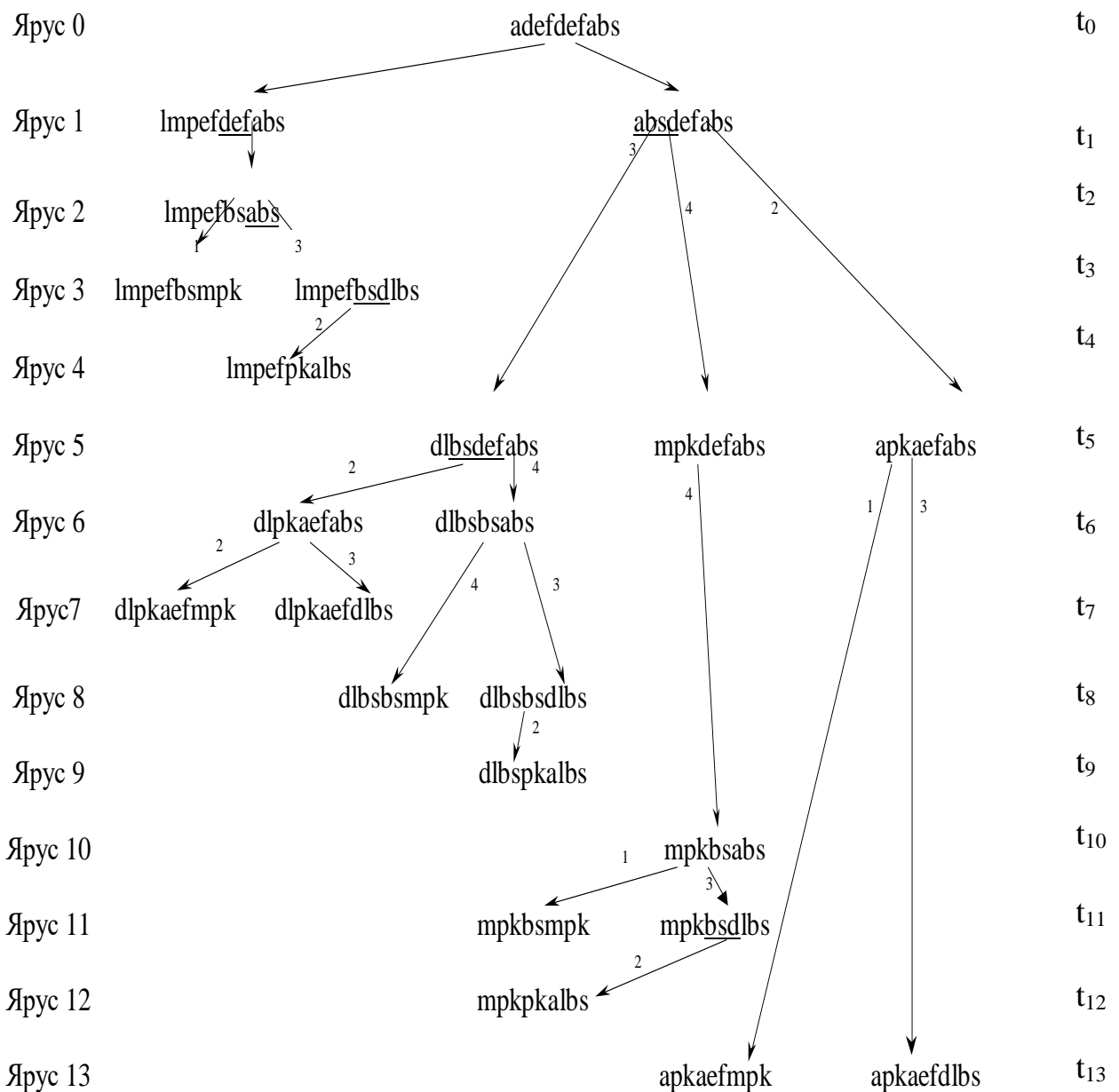


Рисунок 12 - Дерево ИЛИ-вывода над S_2

Равноправное срабатывание конфликтующих продукций и независимое срабатывание неконфликтующих продукций достигается за счет создания необходимого числа копий для альтернативных и независимых путей в дереве вывода.

3 Индивидуальное задание студента

1. Для заданной системы продукций и входного мультислова реализовать ИЛИ-, И- стратегии:

1. $fsk \rightarrow tir$
2. $sk \rightarrow mol$ $S=bsdlskfskbsd$
3. $bsd \rightarrow pm$
4. $sdl \rightarrow pf$

Взяв за основу первую продукцию, предложите модификацию системы продукций (левая часть), чтобы возникла ситуация заикливания пересечений конфликтных слов.

2. Для заданной системы продукций и входного мультислова реализовать ИЛИ-, И- параллельные стратегии:

1. $som \rightarrow fish$
2. $af \rightarrow mok$ $S=asomaftreesom$
3. $tree \rightarrow us$
4. $re \rightarrow god$

Взяв за основу вторую продукцию, предложите модификацию системы продукций (левая часть), чтобы возникла ситуация заикливания пересечений конфликтных слов.

3. Для заданной системы продукций и входного мультислова реализовать ИЛИ-, И- параллельные стратегии:

1. $knut \rightarrow um$
2. $tok \rightarrow ad$ $S=knutokxlpdpdtok$
3. $xlp \rightarrow mir$
4. $pd \rightarrow ir$

Взяв за основу третью продукцию, предложите модификацию системы продукций (левая часть), чтобы возникла ситуация заикливания пересечений конфликтных слов.

4. Для заданной системы продукций и входного мультислова реализовать ИЛИ-, И- параллельные стратегии:

1. $mal \rightarrow lim$
2. $los \rightarrow mil$ $S=losalmaldals$
3. $als \rightarrow dom$
4. $da \rightarrow net$

Взяв за основу четвертую продукцию, предложите модификацию системы продукций (левая часть), чтобы возникла ситуация зацикливания пересечений конфликтных слов.

5. Для заданной системы продукций и входного мультислова реализовать ИЛИ- И- параллельные стратегии:

1. ira→tm
2. ad→lpt S=iradadgoodol
3. good→som
4. dol→mor

Взяв за основу первую продукцию, предложите модификацию системы продукций (левая часть), чтобы возникла ситуация зацикливания пересечений конфликтных слов.

6. Для заданной системы продукций и входного мультислова реализовать ИЛИ-, И- параллельные стратегии:

1. som→more
2. best→mel S=bestolmsoms
3. stol→tn
4. ms→tok

Взяв за основу вторую продукцию, предложите модификацию системы продукций (левая часть), чтобы возникла ситуация зацикливания пересечений конфликтных слов.

7. Для заданной системы продукций и входного мультислова реализовать ИЛИ-, И- параллельные стратегии и:

1. sokrat→um
2. ats→tol S=sokraats#sokratstad
3. sok→tree
4. ad→lm

Взяв за основу третью продукцию, предложите модификацию системы продукций (левая часть), чтобы возникла ситуация зацикливания пересечений конфликтных слов.

8. Для заданной системы продукций и входного мультислова реализовать ИЛИ-, И- параллельные стратегии:

1. abq→tmp
2. qa→sor S=medqabqbedabqa
3. beda→ura
4. med→spor

Взяв за основу четвертую продукцию, предложите модификацию системы продукций (левая часть), чтобы возникла ситуация заикливания пересечений конфликтных слов.

9. Для заданной системы продукций и входного мультислова реализовать ИЛИ-, И- параллельные стратегии:

1. $klp \rightarrow ab$
2. $pol \rightarrow fsa$ $S = apolabsklpsabs$
3. $sa \rightarrow bp$
4. $bs \rightarrow sak$

Взяв за основу первую продукцию, предложите модификацию системы продукций (левая часть), чтобы возникла ситуация заикливания пересечений конфликтных слов.

10. Для заданной системы продукций и входного мультислова реализовать ИЛИ-, И- параллельные стратегии:

1. $plk \rightarrow ba$
2. $lop \rightarrow asp$ $S = splksasbfloplks$
3. $as \rightarrow pbp$
4. $sb \rightarrow as$

Взяв за основу вторую продукцию, предложите модификацию системы продукций (левая часть), чтобы возникла ситуация заикливания пересечений конфликтных слов.

4 Контрольные вопросы

1. Дайте определение интеллектуальной системы. Какие существуют виды интеллектуальных систем.
2. Опишите структуру статической экспертной системы и назначение ее блоков.
3. Что такое машина вывода интеллектуальной системы продукционного типа.
4. Опишите цикл вывода машины вывода. Назначение блока конфликтных слов и блока проверки на динамическое распараллеливание.
5. Дайте определение конфликтного слова и укажите способы его построения.
6. Опишите алгоритм построения расширенного списка конфликтных слов.
7. Что представляют собой логические условия построения списка конфликтных слов.
8. Как выглядит условие заикливания конфликтных слов.
9. Опишите стратегию И-ИЛИ-параллельных продукционных выводов. Каковы ее недостатки.

10. Опишите стратегию равноправных производственных выводов. Каковы ее недостатки.

Библиографический список

1. Сидоркина, И. Г. Системы искусственного интеллекта [Текст] : учебное пособие / И. Г. Сидоркина. - Москва : КНОРУС, 2016. - 246 с.
2. Автоматизированные информационные системы и интеллектуальные технологии [Электронный ресурс] : учебное пособие / Е. А. Титенко [и др.] ; Минобрнауки России, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования "Юго-Западный государственный университет". - Курск : ЮЗГУ, 2013. - 133 с.
3. Основы построения интеллектуальных систем [Электронный ресурс]: учебное пособие/ Г.В. Рыбина. М.: Финансы и статистика, 2010. - 432 с.
4. Потапов А.С. Технологии искусственного интеллекта [Электронный ресурс]: учебное пособие. - СПб: СПбГУ ИТМО, 2010. - 218 с.
5. Серегин, М. Ю. Интеллектуальные информационные системы [Электронный ресурс]: учебное пособие / М. Ю.Серегин, М. А Ивановский, А. В.Яковлев ; Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». - Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2012. - 205 с.