

ЛАБОРАТОРНАЯ РАБОТА 4

Продукционная модель представления знаний

1. Цель работы

Целью данной работы является приобретение студентами умений и навыков реализации пополняемой динамической базы знаний, не включаемой непосредственно в текст программы.

Необходимые для достижения поставленной цели задачи состоят в следующем:

- научиться оценивать возможности применения современных языков высокого уровня для реализации баз знаний экспертных систем (ЭС);
- изучение технических аспектов реализации продукционной модели представления знаний.

2. Задание на лабораторную работу

Реализовать продукционную систему со следующей структурой:

- база правил: область памяти, которая содержит базу знаний – совокупность знаний, представленных в форме правил вида «ЕСЛИ-ТО»;
- глобальная база данных – область памяти, содержащая факты, которые описывают вводимые данные и состояния системы;
- интерпретатор правил (механизм логического вывода) – компонент системы, который формирует заключения, используя базу правил и базу данных.

Для синтаксического представления продукций в модели использовать язык исчисления предикатов первого порядка, то есть основными формализмами представления продукций должны быть:

- а) терм, устанавливающий соответствие знаковых символов описываемому объекту;
- б) предикат для описания отношения сущностей в виде реляционной формулы, содержащей в себе термы.

Термы должны быть двух видов: терм-переменная и терм-константа.

В записи условной части должна быть предусмотрена возможность наличия логических связей «И» и «ИЛИ».

Требования к системе, реализующей продукционную модель представления знаний:

- а) наличие механизма заполнения базы правил и глобальной базы данных, а также отображение результата логического вывода (интерфейс с пользователем);
- б) механизм логического вывода выбрать в зависимости от номера варианта в *таблице 1*:
 - нечетные номера – прямой вывод;
 - четные номера – обратный вывод;
- в) представление системы продукций графом «И/ИЛИ»;
- г) предусмотреть механизм разрешения конфликта на этапе вывода;
- д) обнаружение ошибочных правил в случае, когда либо доказательство заключения закончилось неудачей, либо получено неверное заключение;
- е) протоколирование поиска на графе: описать процесс вывода в системе, используя частичный граф; в случае неуспешного вывода указать невыполнимое условие.

Таблица 1 – Варианты предметных областей для разработки экспертной системы

№ вар.	Предметная область	№ вар.	Предметная область
1	Цветковые растения	6	Выбор вида спорта для занятия с учётом состояния здоровья
2	Аквариумные рыбы	7	Оценка стоимости разработки WEB-страниц
3	Выбор профессии	8	Отбор кандидатов на вакантное место
4	Неисправности ПЭВМ	9	Конфигурация локальной вычислительной сети
5	Компьютерные вирусы	10	Выбор материала для изготовления одежды

3. Краткие теоретические сведения

3.1 Характеристики объекта и продукционные правила

При изучении какого-либо объекта выделяют одно или несколько его свойств, совокупность которых составляет сущность этого объекта в данном рассмотрении. Для описания объекта или его отдельных свойств выбираются некоторые характеристики – величины, которые могут принимать либо количественные, либо качественные значения. В первом случае характеристики называются параметрами, а во втором – признаками описываемого объекта.

Каждая **характеристика** любого объекта может принимать **значения** из некоторого списка (множества) разрешенных значений.

Таблица 2 – Примеры фактов в виде пар «характеристика-значение»

Суждение эксперта	Объект	Пара «характеристика-значение»
«У пациента температура»	Пациент	наличие_температуры=да
«Температура высокая»	Пациент	температура=высокая
«У пациента лихорадка»	Пациент	наличие_лихорадки=да
«Цена минимальная»	Лекарство	цена=минимальная

Так, характеристики в примере из *таблицы 2* принимают:

- «наличие_температуры» и «наличие_лихорадки» – одно из двух значений (да/нет);
- «цена» – одно из нескольких значений (минимальная, договорная, максимальная, постоянная и т.п.).

В свою очередь, совокупность всех характеристик некоторого объекта (или предметной области в целом) образует так называемый список разрешенных характеристик данного объекта (предметной области). Списки разрешенных характеристик и разрешенных значений этих характеристик охватывают множество всех имеющихся фактов, подлежащих хранению в базе знаний экспертной системы. Каждый из списков не является жестко фиксированным, а может изменяться в ходе перепроектирования базы знаний, например, вследствие пополнения её новыми знаниями.

Примеры описания правил с помощью продукций приведены в *таблице 3*.

Таблица 3 – Правила в виде продукций

Предпосылка (антецедент)	Заключение (консеквент)
ЕСЛИ температура = высокая	ТО наличие лихорадки = да
ЕСЛИ цена = минимальная	ТО необходимость закупки = да
ЕСЛИ класс = млекопитающие И потребление мяса = да	ТО отряд = хищник

3.2 Архитектура и особенности экспертных систем

В экспертных системах правила, по которым решаются проблемы в конкретной предметной области, хранятся в базе знаний (рис. 3). *Проблемная область ЭС* описывается посредством фактов и правил. Проблемы ставятся перед системой в виде совокупности фактов, описывающих некоторую ситуацию, и система с помощью базы знаний пытается вывести заключение из этих фактов (рис. 1).



Рис. 1. Вывод заключения в ЭС

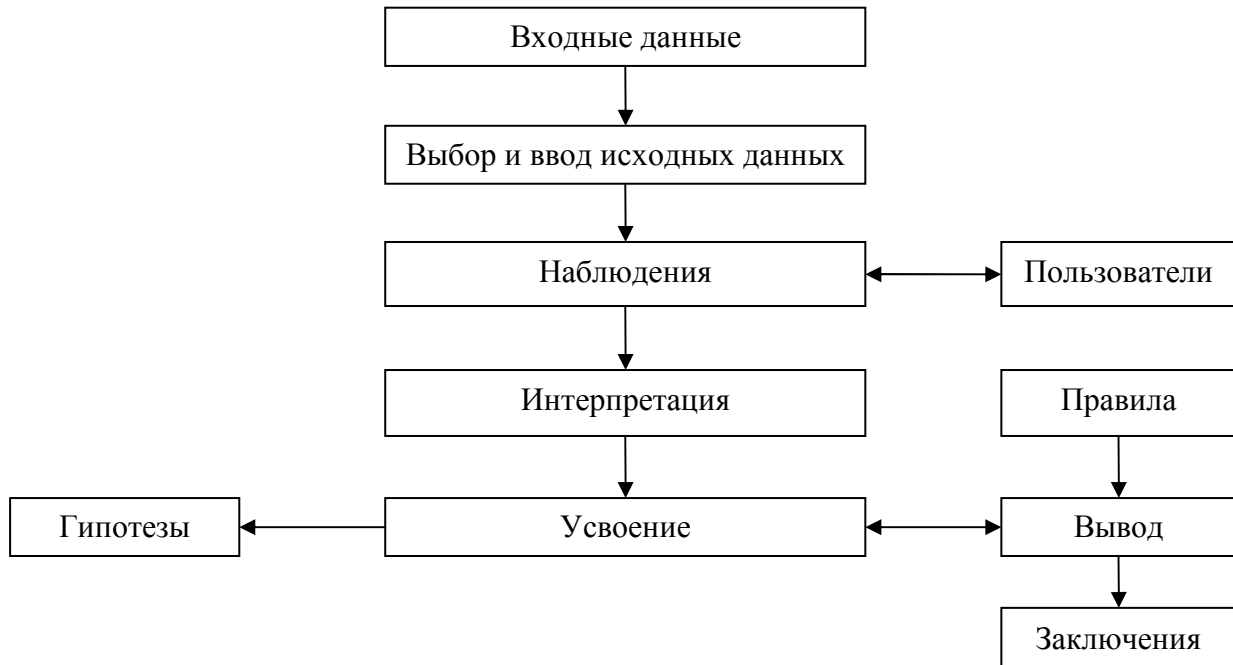


Рис. 2. Схема работы ЭС

Экспертная система может работать в двух режимах: приобретение знаний и решение задач (режим консультации).

Приобретение знаний – это передача экспертной системе потенциального опыта решения проблем (задач) от некоторого источника знаний и преобразование его к виду, доступному для использования в компьютерной программе. Важнейшая составляющая этого процесса – перенос компетентности экспертов на инженеров по знаниям.

Подсистема приобретения знаний предназначена для добавления в базу знаний новых правил и модификации имеющихся. В её задачу входит приведение правила к виду, позволяющему механизму вывода применять это правило в процессе работы.

Наиболее распространенный способ представления знаний – в виде конкретных фактов и правил, по которым из имеющихся фактов могут быть выведены новые. Факты можно представлять тройками (ХАРАКТЕРИСТИКА ОБЪЕКТ ЗНАЧЕНИЕ). Так, (ТЕМПЕРАТУРА ПАЦИЕНТ1 37.5) представляет факт «температура больного, обозначаемого ПАЦИЕНТ1, равна 37.5». В более простых случаях факт выражается не конкретным значением атрибута, а

каким-либо простым утверждением, которое может быть истинным или ложным, например: «Небо покрыто тучами». При этом факту достаточно дать некоторое имя (например, ТУЧИ) или использовать для его представления сам текст соответствующей фразы.

Синтаксис правил наглядно проиллюстрирован *таблицей 3*:

ЕСЛИ A ТО S , где A – условие; S – действие.

В качестве условия A может выступать один либо несколько фактов, соединенные логическими операциями «И» и «ИЛИ»:

$$A_{11} \& A_{12} \& \dots \& A_{mn} \vee \dots \vee A_k$$

Пример:

ЕСЛИ небо покрыто тучами И атмосферное давление падает ТО скоро пойдёт дождь.
(Правило 1).

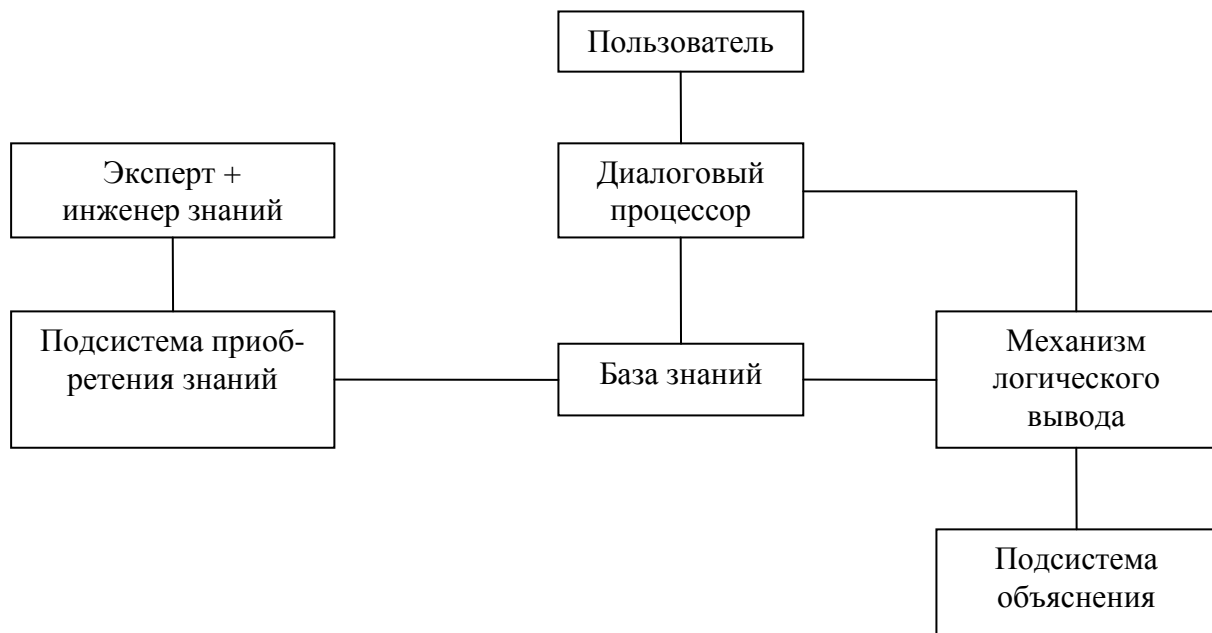


Рис. 3. Структура экспертной системы

Диалог с ЭС осуществляется через *диалоговый процессор* (рис. 3). При этом термин «пользователь» является многозначным, поскольку кроме конечного пользователя применять ЭС может и эксперт, и инженер по знаниям.

В режиме консультации диалоговый процессор обрабатывает данные о задаче пользователя, а также распределяет роли участников (пользователя и ЭС), организует их взаимодействие в процессе совместного решения задачи, преобразует данные пользователя о задаче во внутренний язык системы, а сообщения системы на её внутреннем языке – в язык пользователя. Работа ЭС в режиме консультации может быть представлена в виде схемы на рис. 2.

Действия, входящие в состав правил, могут содержать новые факты. При применении правил из базы знаний ЭС эти факты становятся известны системе, т.е. добавляются в рабочую память (рис. 4). Например, если факты «Небо покрыто тучами» и «Атмосферное давление падает» уже имеются в рабочей памяти, то после применения приведенного выше *Правила 1* в рабочую память также добавляется факт «Скоро пойдёт дождь». На основе данных из рабочей памяти, общих данных о проблемной области и правил из базы знаний механизм логического вывода формирует решение задачи. В режиме консультации экспертная система предварительно формирует исполняемую ей же последовательность действий по решению задачи. Если ответ оказывается не понятным пользователю, он может обратиться к подсистеме объяснения и узнать, как был получен данный ответ. Такой процесс продолжается до тех пор, пока не поступит информация, достаточная для окончательного заключения.

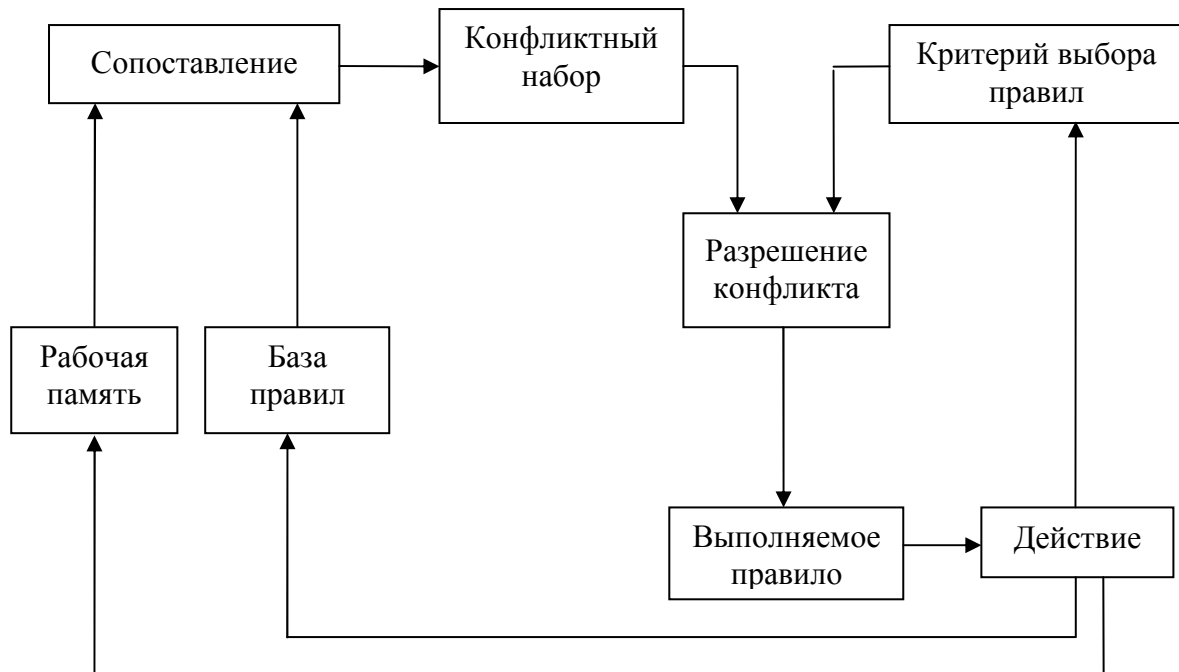


Рис. 4. Цикл работы интерпретатора правил

Если система не может вывести некоторый факт, истинность или ложность которого требуется установить, то система спрашивает о нем пользователя. Например:

ВЕРНО ЛИ, ЧТО небо покрыто тучами?

При получении положительного ответа от пользователя факт «Небо покрыто тучами» добавляется в рабочую память.

3.3 Прямой и обратный вывод

Работа экспертной системы представляет собой циклическую последовательность шагов, на каждом из которых из базы выбирается некоторое правило, которое применяется к текущему содержимому рабочей памяти. Цикл заканчивается тогда, когда выведено либо опровергнуто целевое утверждение. Данный цикл иначе называется логическим выводом. Логический вывод может происходить многими способами, из которых наиболее распространены прямой и обратный порядок вывода.

Прямой порядок вывода – от фактов, которые находятся в рабочей памяти, к заключению. Если такое заключение удастся найти, то оно добавляется в рабочую память. Прямой вывод часто называют выводом, управляемым данными.

Для иллюстрации добавим к нашему примеру базы знаний о погоде ещё одно правило:

ЕСЛИ скоро пойдёт дождь

ТО нужно взять с собой зонтик (*Правило 2*).

Предположим также, что факты «Небо покрыто тучами» и «Атмосферное давление падает» имеются в рабочей памяти ЭС, а целью системы является ответ на вопрос пользователя:

«Нужно взять с собой зонтик?»

При прямом выводе работа системы будет протекать следующим образом:

Шаг 1. Рассматривается *Правило 1*. Его условие истинно, так как оба элемента конъюнкции имеются в рабочей памяти. После применения указанного правила в рабочую память добавляется факт «Скоро пойдёт дождь».

Шаг 2. Рассматривается *Правило 2*. Его условие истинно, т.к. утверждение из условия имеется в рабочей памяти. После применения указанного правила в рабочую память добавляется факт «Нужно взять с собой зонтик». Целевое утверждение выведено.

При обратном порядке вывода заключения просматриваются до тех пор, пока не будут обнаружены в рабочей памяти или получены от пользователя факты, подтверждающие одно из них. В системах с обратным выводом вначале выдвигается некоторая гипотеза, а затем механизм вывода в процессе работы как бы возвращается назад, переходя от выдвинутой гипотезы к фактам и пытаясь найти среди них те, которые подтверждают эту гипотезу. Если она оказалась правильной, то выбирается следующая гипотеза, детализирующая первую и являющаяся по отношению к ней подцелью. Далее отыскиваются факты, подтверждающие истинность подчинённой гипотезы. Вывод такого типа называется управляемым целями. Обратный поиск применяется в тех случаях, когда цели известны и их сравнительно немного.

В рассматриваемом примере вывод целевого утверждения «Нужно взять с собой зонтик» обратной цепочкой рассуждений выполняется следующим образом:

Шаг 1. Рассматривается *Правило 1*. Оно не содержит цели в правой части. Переходим к *Правилу 2*.

Шаг 2. Рассматривается *Правило 2*. Оно содержит цель в правой части правила. Переходим к правой части правила и рассматриваем в качестве текущей цели утверждение «Скоро пойдёт дождь».

Шаг 3. Текущей цели нет в рабочей памяти. Рассмотрим *Правило 1*, которое содержит цель в правой части. Обе компоненты его условия имеются в рабочей памяти и условие истинно. Применяем *Правило 1*, в результате выводится утверждение: «Скоро пойдёт дождь», которое было нашей предыдущей целью.

Шаг 4. Применяем *Правило 2*, условием которого является данное утверждение. Получаем вывод исходного утверждения.

Заметим, что для упрощения ситуации мы предположили, что в обоих случаях факты «Небо покрыто тучами» и «Атмосферное давление падает» уже известны системе. В действительности система выясняет истинность или ложность факта, входящего в условие некоторого правила, спрашивая об этом пользователя при попытке применить правило.

3.4 Структура механизма вывода

Механизм (подсистема) вывода – программная компонента экспертных систем, реализующая процесс её рассуждений на основе базы знаний и содержимого рабочей памяти. Она выполняет две функции: во-первых, просмотр фактов из рабочей памяти и правил из базы знаний и добавление (по мере возможности) в рабочую память новых фактов и, во-вторых, определение порядка просмотра и применения правил. Соответственно, в составе механизма вывода первую функцию реализует *компонент вывода*, вторую – *управляющий компонент*. При этом механизм вывода управляет процессом консультации, сохраняет для пользователя информацию о полученных заключениях и опрашивает его, когда для срабатывания очередного правила имеющихся фактов в рабочей памяти недостаточно.

Существуют экспертные системы с монотонным и немонотонным выводом. В системе с монотонным выводом факты, хранимые в базе знаний, статичны, то есть не изменяются в процессе решения задачи. В системе с немонотонным выводом допускается изменение или удаление фактов из базы знаний. В такой системе могут быть изменены даже те данные, которые после вывода уже вызвали срабатывание каких-либо правил. Иными словами имеется возможность модифицировать значения атрибутов в составе фактов, находящихся в рабочей памяти. В свою очередь, изменение фактов приводит к необходимости удаления из базы знаний заключений, полученных с помощью упомянутых правил. Тем самым вывод выполняется повторно для того, чтобы пересмотреть те решения, которые были получены на основе подвергшихся изменению фактов.

Действие компонента вывода основаны на применении правила *modus ponens*, суть которого состоит в следующем: пусть известно, что истинно утверждение *A* и существует правило вида «Если *A*, то *B*», тогда утверждение *B* также истинно. Правила из базы знаний срабатывают при наличии фактов, удовлетворяющих их левой части: если истинна посылка, то должно быть истинно и заключение.

Управляющий компонент определяет порядок применения правил, а также устанавливает, имеются ли ещё факты, которые могут быть изменены в случае продолжения консультации. Данный компонент выполняет четыре функции:

- 1) *сопоставление* – образец правила сопоставляется с имеющимися фактами;
- 2) *выбор* – если в конкретной ситуации могут быть применены сразу несколько правил, то из них выбирается одно, наиболее подходящее по заданному критерию (разрешение конфликта);
- 3) *срабатывание* – если образец правила при сопоставлении совпал с какими-либо фактами из рабочей памяти, то правило срабатывает;
- 4) *действие* – содержимое рабочей памяти подвергается изменению путем добавления в неё заключения сработавшего правила. Если в правой части правила содержится указание на какое-либо действие, то оно выполняется (пример – системы обеспечения безопасности информации).

Интерпретатор правил работает циклически. В каждом цикле он просматривает все правила, чтобы выявить среди них те посылки, которые совпадают с известными на данный момент фактами из рабочей памяти. Интерпретатор также определяет порядок применения правил. После выбора правило срабатывает, его заключение заносится в рабочую память, после чего цикл повторяется. В одном цикле может сработать только одно правило. Если несколько правил успешно сопоставлены с фактами, то интерпретатор производит выбор по определенному критерию единственного правила, которое и срабатывает в данном цикле. Цикл работы интерпретатора схематически представлен на рис. 4.

Факты из рабочей памяти последовательно сопоставляются с посылками правил для выявления успешного сопоставления. Совокупность отобранных правил составляет так называемый конфликтный набор. Для разрешения конфликта интерпретатор имеет критерий, с помощью которого выбирается единственное правило, которое и срабатывает. Новые данные, введенные в систему сработавшим правилом, в свою очередь могут изменить критерий выбора правила. Например, для игры в шахматы это может быть переход к использованию оборонительной стратегии с последующим возвратом к наступательной игре.

Уже на ранней стадии разработки ЭС необходимо знать, какую информацию будет вводить конечный пользователь. При этом участие пользователя выражается в следующем:

- *конкретные задачи*. Пользователь, сталкиваясь с конкретными проблемами, может объяснить возникновение проблем и предложить возможные варианты их решения;
- *общение*. Интерфейс пользователя должен соответствовать его словарю и уровню подготовки;
- *установление связей*. Знакомство пользователя с причинами и последствиями, вызывающими то или иное действие в процессе функционирования системы, неопределимо в определении взаимосвязей фактов в базе знаний;
- *обратная связь*. Отличительной особенностью удобной в использовании ЭС является её способность объяснить конечному пользователю ход своих рассуждений.

3.5 Диалог с пользователем и подсистема объяснения

На примере простой ЭС и базы знаний диалог пользователя с системой можно представить себе следующим образом:

Система: Вы хотите узнать, нужно ли взять с собой зонтик?

Пользователь: Да.

Система: Верно ли, что небо покрыто тучами?

Пользователь: Да.

Система: Верно ли, что атмосферное давление падает?

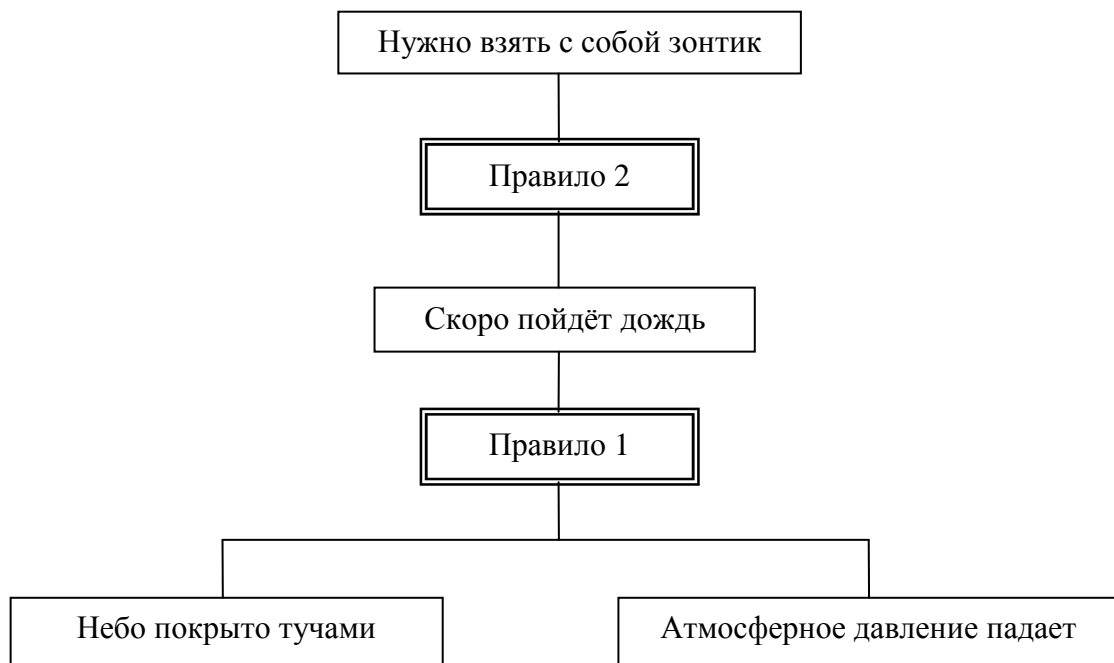
Пользователь: Да.

Система (после некоторого «размышления»): Нужно взять с собой зонтик.

Отметим, что подавляющее большинство систем, основанных на знаниях, обладают достаточно примитивным интерфейсом на естественном языке – допустимые входные сообщения пользователя ограничены набором понятий, содержащихся в базе знаний.

Как видно из этого примера, в ходе консультации инициатива диалога принадлежит системе, а сама консультация у экспертной системы выглядит примерно так же, как и консультация у эксперта-человека: задается ряд вопросов и на основании их анализа выдается экспертное заключение. Чтобы удостовериться в «разумности» и «компетентности» экспертной системы, пользователь может обратиться к её подсистеме объяснения.

В процессе консультации ЭС строит дерево вывода и хранит его в памяти в некоторой внутренней форме. Успешному применению правила соответствует добавление узла с его именем, потомками которого являются узлы, соответствующие некоторым из уже выведенных фактов, а предком – новый узел, соответствующий факту, содержащемуся в правой части правила. В рассматриваемом примере дерево вывода будет иметь следующий вид:



На данной схеме простыми рамками выделены узлы дерева, соответствующие фактам, а двойными – узлы, соответствующие названием правил. Сверху от узла-правила изображён факт, находящийся в его правой части (в терминологии деревьев – предок узла-правила). Листья дерева (т.е. узлы нижнего «яруса») соответствуют либо фактам, истинностные значения которых запрашиваются у пользователя, либо первоначально известным фактам из базы знаний, корень дерева (самый верхний узел) – целевому утверждению.

Рассмотрим теперь работу подсистемы объяснения. Для получения объяснения в процессе консультации пользователь может «перехватить инициативу» диалога в тот момент, когда система задает очередной вопрос, «переспросив» в ответ систему, почему она его задала. При этом вместо ответа на вопрос пользователь может задать системе встречный вопрос:

ПОЧЕМУ? («Почему система задала такой вопрос?»).

Система интерпретирует вопрос «Почему?» в «понятных» ей терминах дерева вывода, поднимаясь по нему на один ярус выше и находя правило, для применения которого система за-

дает этот вопрос. Пользователю выдается информация об этом правиле, о состоянии вычисления его условиям о заключении данного правила (текущей цели).

Пусть в нашем примере диалога с экспертной системой пользователь вместо ответа на вопрос системы: «Верно ли, что атмосферное давление падает?» задает ей встречный вопрос: «ПОЧЕМУ?». ЭС обращается к подсистеме объяснения, которая выдает пользователю следующее сообщение:

[1.0] Эта информация необходима, чтобы установить, что скоро пойдёт дождь.

Ранее было установлено, что:

[1.1] Небо покрыто тучами.

Следовательно, если:

[1.2] Атмосферное давление падает, то скоро пойдёт дождь.

(Правило 1).

Двойные номера в квадратных скобках имеют следующий смысл. Для облегчения работы пользователя подсистема объяснения нумерует свои ответы (номер ответа, в данном случае это 1, является первой цифрой номера), а в каждом ответе нумерует также утверждения, о которых в нём идет речь. Пользователь может использовать двойные номера для ссылок на утверждения, не выписывая их явно. Например, номер 1.1 обозначает утверждение «Небо покрыто тучами».

Допустим, что пользователь хочет проследить логику рассуждений системы дальше и узнать, почему ей необходимо установить, скоро ли пойдёт дождь. Для этого он может повторно задать ей вопрос:

ПОЧЕМУ? («Почему необходимо установить, что скоро пойдёт дождь?»).

Второй ответ системы аналогичен первому; в нем указывается, что это необходимо для применения *Правила 2* с целью установить, что нужно взять с собой зонтик:

[2.0] Эта информация необходима, чтобы установить, что нужно взять с собой зонтик.

Если:

[2.1] Скоро пойдёт дождь, то нужно взять с собой зонтик.

(Правило 2).

Утверждение [2.0] является исходным целевым утверждением системы. Таким образом, теперь ход рассуждений системы пользователю полностью понятен. Вопрос «ПОЧЕМУ?» позволяет проследить ход рассуждений системы от посылок к заключениям. Однако для более детального осмысления процесса вывода удобно было бы иметь возможность изучать его и в противоположном порядке – от заключений к посылкам. Для этого служит другой вопрос, который также «понимает» подсистема объяснений:

КАК N?

где N – номер утверждения, выданный подсистемой объяснения на одном из предыдущих шагов диалога.

Например, в ответ на вопрос:

КАК 2.0?

«Как получено утверждение 2.0?» подсистема объяснения выдает информацию в правиле, которое было применено для его получения:

[3.1] Используется правило 2, заключением которого является,
что нужно взять с собой зонтик.

Чтобы получить более подробную информацию о том, как было использовано *Правило 2*, следует повторно задать вопрос «КАК?»:

КАК 3.1? «Как было использовано правило 2?». Ответ системы:

Ранее установлено, что:

[4.1] Скоро пойдёт дождь.

Следовательно, нужно взять с собой зонтик.

Выясним теперь, как получено утверждение 4.1:

КАК 4.1?

Ответ системы:

[5.1] Используется правило 1, заключением которого является то, что «скоро пойдёт дождь». Наконец, выясним, как было использовано правило 1:

КАК 5.1?

Ответ системы:

Ранее установлено, что:

[6.1] Небо покрыто тучами.

[6.2] Атмосферное давление падает.

Следовательно, скоро пойдёт дождь.

Как именно были получены утверждения 6.1 и 6.2, пользователь помнит, поскольку он отвечал на вопросы системы относительно этих утверждений. Если все же он задаст системе вопрос КАК 6.1? или КАК 6.2?, то система напомнит ему об этом.

Описанная модель объяснения используется в системе MYCIN. Достоинством данной модели является возможность получения объяснения любого шага работы системы, недостатком – жёсткая привязка к дереву вывода. Для пользователя-непрограммиста такой способ объяснения может показаться излишне формальным и детализованным. Объяснение на «более высоком уровне» – с точки зрения здравого смысла и без углубления в детали, требует более сложной организации знаний и является перспективой экспертных систем.

4. Содержание отчёта по работе

В отчете должны быть представлены:

- конфигурация системы продукций;
- описание способа организации поиска на графе;
- описание конфликтного набора и алгоритм разрешения конфликта при логическом выводе;
- тестовый набор правил – база правил и база данных;
- протокол тестового вычисления: обход по графу.

Рекомендуемая литература

1. Михайлов Д.В. Системы искусственного интеллекта: курс лекций. Лекция 7: Представление знаний правилами и логический вывод [Электронный ресурс] / Д.В. Михайлов. Режим доступа: http://www.machinelearning.ru/wiki/images/2/2f/Lect_7_ai_mdv.pdf (дата обращения: 22.04.2016).
2. Сойер Б. Программирование экспертных систем на Паскале [Электронный ресурс] / Б. Сойер, Д.Л. Фостер. Режим доступа: <http://www.novsu.ru/file/1005762> (дата обращения: 22.04.2016).
3. Марселлус Д. Программирование экспертных систем на Турбо Прологе [Электронный ресурс] / Д. Марселлус. Режим доступа: <http://www.novsu.ru/file/1004901> (дата обращения: 22.04.2016).
4. Новикова В.А. Искусственный интеллект и экспертные системы [Электронный ресурс] / В.А. Новикова, Е.Ю. Андреева, Д.К. Туйкина. Режим доступа: http://expro.ksu.ru/materials/ii_i_es/book.html (дата обращения: 22.04.2016).
5. Чулюков В.А. Системы искусственного интеллекта. Практический курс: учебное пособие [Текст] / [В.А. Чулюков, И.Ф. Астахова, А.С. Потапов и др.; под ред. И.Ф. Астаховой]. М.: БИНОМ. Лаборатория знаний, 2008. – 292 с.