

Экспертные системы — это вычислительные системы, способные в определённой предметной области принимать решения, схожие с решениями экспертов-людей. Выглядит это примерно так: система задаёт ряд вопросов, причём последующие вопросы зависят от полученных ответов. Затем система делает вывод и показывает всю цепочку рассуждений, которая к нему привела. То есть знания и опыт эксперта тиражируются, а что не менее важно — тиражируется сам ход его рассуждений.

Кроме того, можно применить несколько интересных особенностей. Например, система может сделать не один вывод, а перечислить варианты с разной степенью вероятности. Можно учитывать различные виды нечётких данных — если пользователь не знает ответы на некоторые вопросы или что-то предполагает. Вместо пользователя система может опрашивать датчики или брать информацию из разных источников. Продвинутое системы могут самообучаться, то есть выявлять закономерности во время сеансов работы и генерировать новые знания.

Таким образом, экспертные системы могут применяться для довольно широкого круга задач:

- интерпретация, т. е. описание ситуации по наблюдаемым данным, определение смысла данных;
- диагностика — заключение о нарушениях в системе, составленное на основе наблюдений;
- отладка, исправление неисправностей — составление рекомендаций и выполнение последовательности действий по устранению неисправностей в системе;
- мониторинг — непрерывное сравнение результатов наблюдений с критическими точками плана;
- прогноз — предсказание будущих событий на основе анализа имеющихся данных о прошлом и настоящем;
- проектирование, конструирование — подготовка спецификаций для создания объектов с заранее определёнными свойствами;
- планирование — нахождение плана действий для достижения заранее поставленной цели;
- обучение какой-либо дисциплине или приёмам использования чего-либо;
- управление — решение задач проектирования и планирования, а также интерпретации и диагностики с корректировкой имеющихся планов;
- поддержка принятия решений — помощь в формировании или выборе варианта действий среди множества альтернатив.

Конечно, есть и ограничения. Во-первых, использовать экспертные системы есть смысл только в довольно узких предметных областях, где трудно найти экспертов и легче использовать компьютерную систему. Во-вторых, они не подходят для тех областей, в которых эксперты руководствуются не столько своими знаниями, сколько здравым смыслом, то есть сведениями из общей области знаний, не связанной с конкретным предметом. Также к недостаткам экспертных систем можно отнести то, что они плохо соотносятся с реляционными базами данных, к которым мы все так привыкли. Тем не менее, в настоящее время существует и используется довольно много экспертных систем в сфере медицины, химии и военного дела. Вот некоторые примеры:

- [Система](#) по глобальной онтологии;
- [Акинатор](#) — система, которая отгадывает загаданного вами персонажа;
- [MYCIN](#) — выбор антимикробной терапии в условиях стационара;
- [DENDRAL](#) — химический анализ сложных молекул;
- поговаривают также, что отечественная [система «Периметр»](#), предназначенная для ответного удара в случае уничтожения командных пунктов, оснащена экспертной системой (которая и принимает решение о нанесении удара).

Попробуем создать небольшую экспертную систему. Для первого раза возьмём тренировочную предметную область. Хороший способ убедиться, что искусственный интеллект своими руками — это просто.

Не секрет, что в летний сезон в России ожидается резкий рост количества экспертов по приготовлению шашлыка. Думаю, многим знакома ситуация, когда кто-то один готовит шашлык а несколько людей стоят вокруг и дают советы и вообще высказывают своё экспертное мнение. Поведение таких шашлычных экспертов мы и попытаемся симитировать.

Немного теории

Для начала разберёмся, что нам предстоит сделать.

Итак, экспертная система состоит из нескольких основных компонентов:

- база знаний;
- механизм логического вывода (МЛВ);
- компонента объяснения.

База знаний

База знаний — это, можно сказать, сердце экспертной системы. Знание — это информация вместе со способом её интерпретации, то есть это более высокий уровень информации. Система, обладающая знаниями, может не только выдавать информацию, но и объяснять её смысл и происхождение. Описание способа интерпретации называется метаданными.

Зачастую вся сложность создания экспертной системы заключается в формировании базы знаний. Этим занимаются специально обученные люди — инженеры по знаниям. Совместно с одним или несколькими экспертами они формулируют правила, имеющиеся в предметной области, и заносят их в определённом виде в базу знаний. В промышленных системах количество правил может исчисляться тысячами. При этом связи в предметной области могут быть такими запутанными и даже противоречивыми, что незначительная модификация базы знаний, например изменение порядка следования двух правил, может вызвать кардинальные изменения в работе всей системы.

Отдельная трудность заключается в самом общении с экспертами. Эксперты могут расходиться во мнениях и устраивать принципиальные споры. Кстати, поэтому предпочтительно использовать экспертные системы для тех областей знаний, которые уже устоялись. Кроме того, эксперты могут применять какие-то правила неосознанно или даже намеренно что-то скрывать. Остаётся только пожелать инженерам по знаниям, чтобы они держались.

Получаемая при этом база знаний может быть представлена в различных видах. Это может быть, к примеру, семантическая сеть. Но более классический вариант — набор продукционных правил (продукций). Продукционное правило — это правило вида *ЕСЛИ <условие> ТО <действие>*. Формулируя на естественном языке: *если мясо жарится долго, оно приготовилось*. Где же тут действие, спросите вы. Обо всём по порядку. Посмотрим на структуру правила более пристально.

И условие, и действие содержат в себе некие факты. Факт сам по себе может быть истинным или ложным, а также мы можем не иметь данных о его истинности. Факт может иметь определённую степень уверенности, например мы не знаем, что значит «долго» в контексте приготовления шашлыка, но кажется, что жарится уже довольно давно. Состоит факт из переменной и значения. Значение берётся из множества возможных значений — домена переменной. То есть наше условие «мясо жарится долго» можно переписать в виде: *Время жарки = долго*, где «время жарки» — переменная, а «долго» — значение. Если все возможные вопросы относительно времени жарки будут иметь ответ только «долго» и «недолго» — это и будет домен этой переменной.

Действие представляет собой присвоение истинности некоторому факту. То есть вместо человеческого «оно приготовилось», в ходе работы правила факт *Мясо готово = да* помечается как истинный.

Ещё раз посмотрим на то, что получилось из простого предложения эксперта. Из начального *Если мясо жарится долго, оно приготовилось* получается:

Переменные:

1. Время жарки (домен: долго, недолго).
2. Мясо готово (домен: да, нет).

Правило:

Если факт «время жарки = долго» — истинный, то факт «мясо готово = да» пометить как истинный.

Конечно, если мясо долго жарится, ещё нельзя уверенно утверждать, что оно готово, может быть, угли холодные. Но в нашем примере мы закроем на это глаза и не будем использовать степень уверенности в фактах.

Так, изучая высказывания экспертов, инженеры по знаниям выделяют имеющиеся в предметной области сущности и формируют правила. Что же с этими правилами происходит дальше?

Механизм логического вывода

После того как база знаний сформирована, можно задавать вопросы системе. Механизм логического вывода обеспечивает поиск ответов на эти вопросы. Вопрос задаёт цель консультации, в общем случае это определение значения какой-либо переменной. В нашем примере мы хотим понять, что же делать с шашлыком, то есть целью консультации будет значение переменной *Действие с шашлыком*. Возможными вариантами будут *ждать, перевернуть* или *снимать*.

На первом шаге механизм находит в базе знаний все правила, в которых переменная-цель присутствует в качестве вывода, не важно, с каким значением. Теперь каждое правило, если они нашлись, по очереди (именно поэтому важен порядок следования правил) проверяются до первого сработавшего. То есть до того, все условия которого окажутся истинными. Для проверки правила берутся все его условия и, опять же, по очереди проверяются — объявляются целью консультации, и алгоритм запускается для них с первого шага.

Не для всех переменных существуют правила, которые их определяют. Некоторые запрашиваются из внешних источников, в простейшем случае — у пользователя. Переменные, которые определяются правилами, называются *выводимыми*, а определяемые пользователем во время консультации — *запрашиваемыми*. Тип переменной определяется ещё на этапе формирования базы знаний, и для запрашиваемых должны быть заданы способы определения, например сформулирован вопрос для пользователя. Если пользователь не может ответить или для выводимой переменной не нашлось правил, или ни одно из правил не сработало — переменная остаётся без значения, а пользователь — без установленной истины.

Существуют ещё *выводимо-запрашиваемые переменные*. Для них существуют и правила вывода, и вопрос пользователю. Сначала механизм логического вывода пытается определить такие переменные с помощью правил, а если не получается — задаёт вопрос.

Компонента объяснения

Ещё одна немаловажная часть экспертной системы. Именно объяснение хода рассуждений и обоснование выводов выгодно отличают экспертную систему от других отраслей искусственного интеллекта.

На этапе формирования базы знаний в качестве метаданных к правилам прилагаются их объяснения на естественном языке. После консультации система может показать все сработавшие правила вместе с этими пояснениями. Комментарии можно также добавлять к отдельным фактам или переменным.

В итоге мы узнаём не только что надо перевернуть шашлык, но и почему нужно это сделать.

Это очень серьёзное преимущество для систем поддержки принятия решений и для других применений экспертных систем. Некоторые промышленные системы в качестве результата дают несколько ответов с пояснениями и степенями вероятности.

Переход к практике

Механизм логического вывода и компонента объяснения не относятся к конкретной экспертной системе. Они используют правила и метаданные базы знаний, но представляют собой скорее оболочку для экспертных систем. Имея такую оболочку, можно конструировать различные системы, создавая только базы знаний. Возможно, потребуются какие-то настройки в механизме логического вывода, но в целом оболочка может использоваться в любой предметной области.

Оболочка экспертной системы

Конечно, существует множество готовых оболочек. Они предоставляют различные возможности и различные способы формирования базы знаний. Однако, я достаточно жадеен, чтобы не использовать платные решения и достаточно высокомерен, чтобы из бесплатных ни одно меня не устроило.

Ещё в студенческие годы мне довелось написать оболочку для экспертной системы. Конечно, она далеко не идеальна, ни с точки зрения кода, ни с точки зрения пользовательского интерфейса. Но, во-первых, она работает. По крайней мере для небольших задач, как наша, она вполне годится для примера. Во-вторых, как ни странно, та же Википедия называет одним из основных недостатков экспертных систем отсутствие графического пользовательского интерфейса (взаимодействие обычно идёт через терминал). Так что наличие какого бы то ни было интерфейса (который у меня есть) — уже достаточный вклад в отрасль.

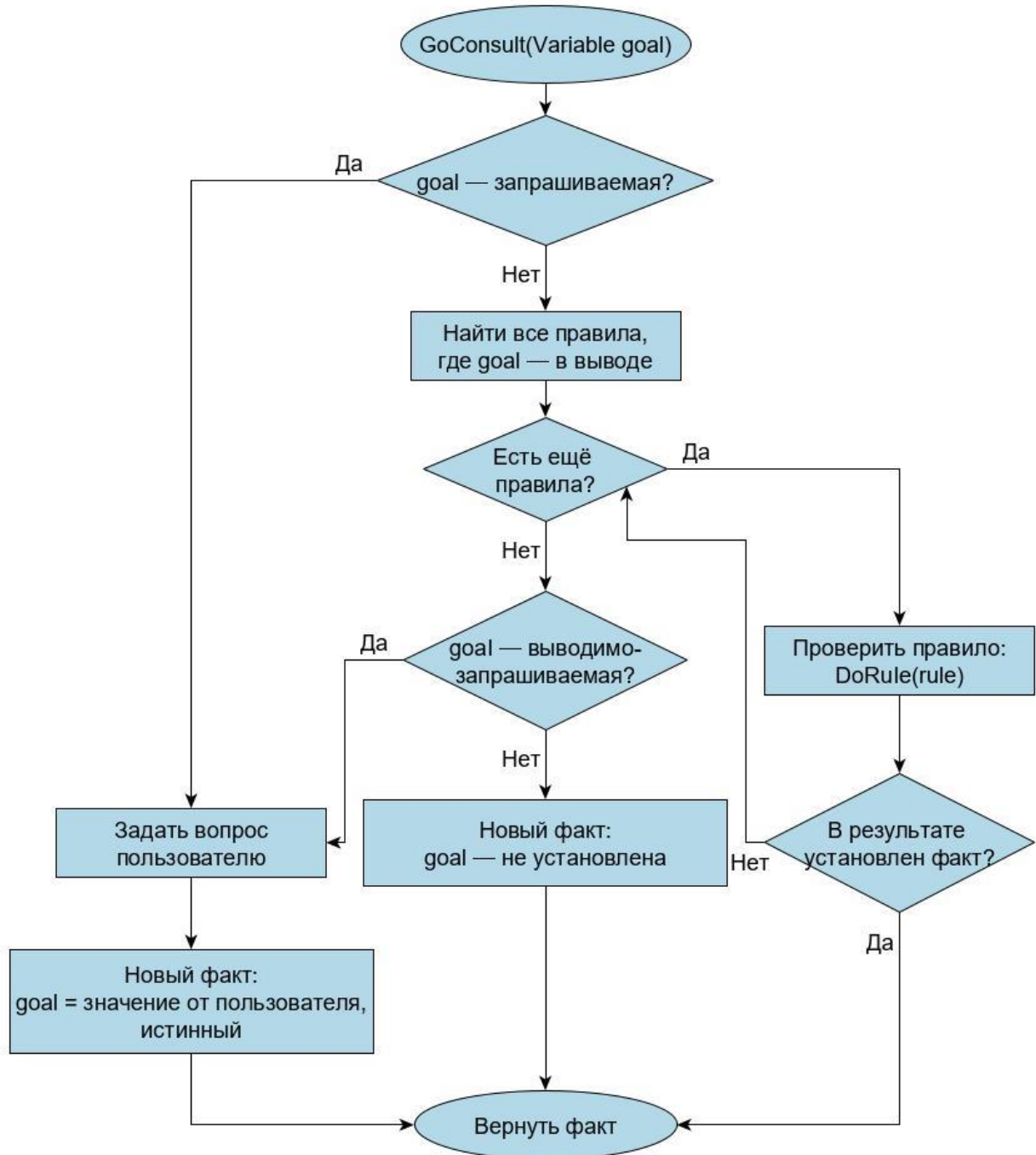
Далее я опишу схему данных и основной алгоритм работы механизма логического вывода в собственной оболочке, и дальнейшее создание экспертной системы буду показывать именно в ней. Вам же предлагаю на выбор — поискать оболочки в интернете (например можно взять [эту в Википедии](#), а вот и [ссылка на сам проект](#)), воспользоваться [моей](#).

Модель данных можно использовать самую простую, как в теории:

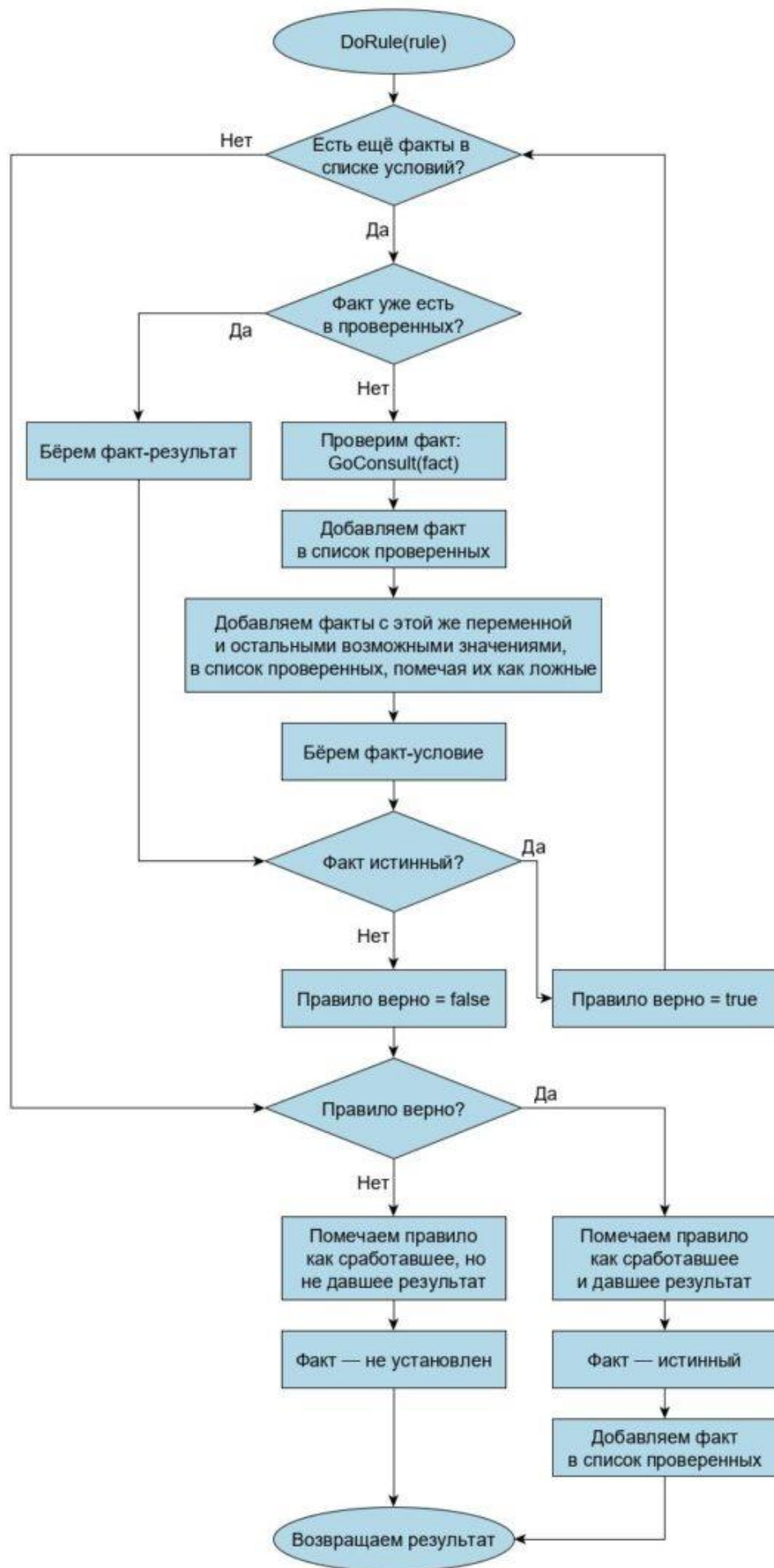
- Domain (Домен):
 - Имя.
 - Список значений.
- Variable:
 - Имя.
 - Домен.
 - Тип (выводимая, запрашиваемая, выводимо-запрашиваемая).
 - Вопрос для пользователя.
 - Объяснение.
- Fact:
 - Переменная.
 - Значение.
 - Истинность (неизвестно, истинный, ложный).
- Rule:
 - Список фактов-условий.

- Факт-вывод.
- Результат (не сработало, сработало и вывод признан истинным, сработало и не привело к получению значения переменной).

В режиме проектирования базы знаний пользователь может завести все эти данные. Затем он переходит в режим консультации. Выбирается цель консультации, и запускается процедура GoConsult. Вот схема её работы:



Как видно, эта процедура вызывает проверку правил DoRule, её тоже рассмотрим подробно:



DoRule в свою очередь вызывает GoConsult, то есть алгоритм получается косвенно-рекурсивным. А в результате мы получаем факт — значение искомой переменной, которое введено пользователем или получено из базы знаний, или отсутствие значения, если истину определить не удалось.

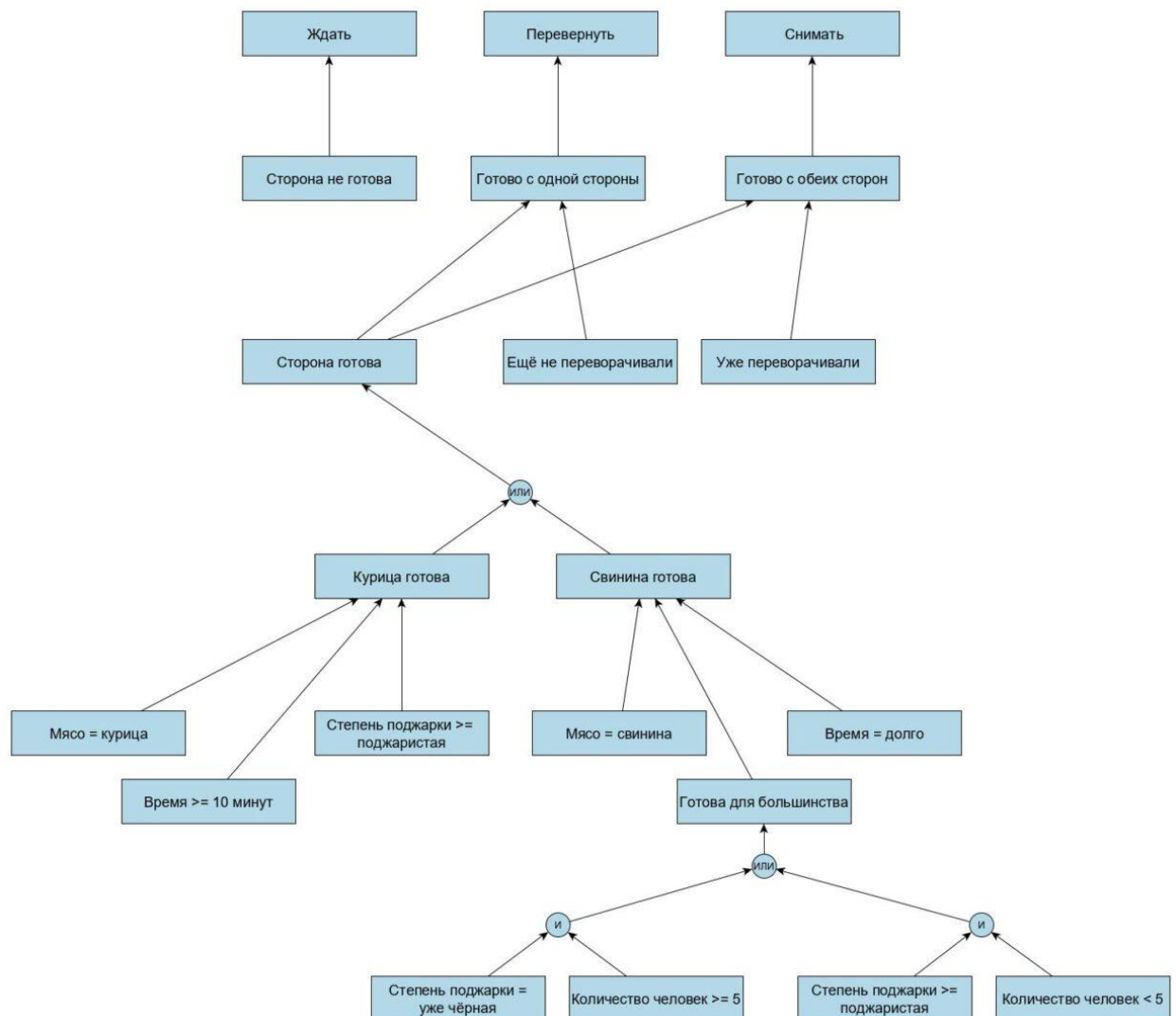
Теперь, когда у нас есть модель данных и алгоритм, можем переходить к самому интересному — проектированию базы знаний.

База знаний

Что ж, начнём записывать в систему свои знания о шашлыке. При этом нужно помнить несколько вещей:

1. Согласно алгоритму логического вывода, все условия в правиле объединены через И. Чтобы использовать ИЛИ, нужно создать несколько правил.
2. Создать нужно не только подтверждающие что-то правила, но и противоположные. Например, если мы создаём правило *ЕСЛИ Время жарки = Долго И Степень поджарки = Поджаристое, ТО Готово = Да*, мы должны создать ещё такие правила: *ЕСЛИ Время жарки = Только положили ТО Готово = Нет* и *ЕСЛИ Степень поджарки = Ещё сырое, ТО Готово = Нет*.
3. Важен порядок правил в базе знаний.

Нужно формализовать свои знания о шашлыке и представить в виде модели базы знаний, то есть выделить домены, переменные и правила. Чтобы было легче сформировать набор правил, да ещё и в нужном порядке, можно составить их графическое представление. У меня получилась довольно дикая, боюсь, понятная только мне схема:



Вершины определяют список переменных и возможных значений, а также какие переменные будут запрашиваемыми, а какие выводимыми. Дуги указывают на некоторые связи, из которых впоследствии получаются правила. Здесь я предположил, что мы будем жарить курицу или свинину, а система может подсказать три действия — ждать, перевернуть, снимать. Исходя из вашего опыта, вы можете сделать другие правила, расширить или переделать эту базу знаний. Вообще говоря, этим база знаний и хороша, что её ещё долго можно расширять, включая новые виды мяса, предполагаемые действия, обстоятельства и так далее. Пока же у меня получилось 27 правил:

Правила

```

Ждать: ЕСЛИ (Сторона готова = Нет) ТОГДА Действие = Ждать
Перевернуть: ЕСЛИ (Готово с одной стороны = Да) ТОГДА Действие = Перевернуть
Снимать: ЕСЛИ (Готово с двух сторон = Да) ТОГДА Действие = Снимать
Готова первая сторона: ЕСЛИ (Сторона готова = Да) И (Уже переворачивали = Нет) ТОГДА Готово с одной стороны = Да
Первая сторона не готова: ЕСЛИ (Уже переворачивали = Нет) И (Сторона готова = Нет) ТОГДА Готово с одной стороны = Нет
Готова вторая сторона: ЕСЛИ (Сторона готова = Да) И (Уже переворачивали = Да) ТОГДА Готово с двух сторон = Да
Вторая сторона не готова: ЕСЛИ (Уже переворачивали = Да) И (Сторона готова = Нет) ТОГДА Готово с двух сторон = Нет
Готова сторона курицы: ЕСЛИ (Курица готова = Да) ТОГДА Сторона готова = Да
Готова сторона свинины: ЕСЛИ (Свинина готова = Да) ТОГДА Сторона готова = Да
Сторона не готова: ЕСЛИ (Вид мяса = Свинина) И (Свинина готова = Нет) ТОГДА Сторона готова = Нет
Сторона не готова 2: ЕСЛИ (Вид мяса = Курица) И (Курица готова = Нет) ТОГДА Сторона готова = Нет
Курица готова 1: ЕСЛИ (Вид мяса = Курица) И (Время жарки = Ну минут 10) И (Степень поджаристости = Поджаристое) ТОГДА Курица готова = Да
Курица готова 2: ЕСЛИ (Вид мяса = Курица) И (Время жарки = Да долго уже) ТОГДА Курица готова = Да
Курица готова 3: ЕСЛИ (Вид мяса = Курица) И (Степень поджаристости = Уже черное) ТОГДА Курица готова = Да
Курица не готова 1: ЕСЛИ (Вид мяса = Курица) И (Время жарки = Только что положил) ТОГДА Курица готова = Нет
Курица не готова 2: ЕСЛИ (Вид мяса = Курица) И (Степень поджаристости = Совсем сырое еще) ТОГДА Курица готова = Нет
Курица не готова 3: ЕСЛИ (Вид мяса = Курица) И (Степень поджаристости = Начинает поджариваться) ТОГДА Курица готова = Нет
Свинина готова: ЕСЛИ (Вид мяса = Свинина) И (Время жарки = Да долго уже) И (Готово по мнению большинства = Да) ТОГДА Свинина готова = Да
Свинина не готова 1: ЕСЛИ (Вид мяса = Свинина) И (Время жарки = Только что положил) ТОГДА Свинина готова = Нет
Свинина не готова 2: ЕСЛИ (Вид мяса = Свинина) И (Время жарки = Ну минут 10) ТОГДА Свинина готова = Нет
Свинина не готова 3: ЕСЛИ (Вид мяса = Свинина) И (Готово по мнению большинства = Нет) ТОГДА Свинина готова = Нет
Готова по мнению большинства 1: ЕСЛИ (Количество человек = Меньше пяти) И (Степень поджаристости = Поджаристое) ТОГДА Готово по мнению большинства = Да
Готова по мнению большинства 2: ЕСЛИ (Количество человек = Меньше пяти) И (Степень поджаристости = Уже черное) ТОГДА Готово по мнению большинства = Да
Готова по мнению большинства 3: ЕСЛИ (Количество человек = Пять или больше) И (Степень поджаристости = Уже черное) ТОГДА Готово по мнению большинства = Да
Не готово по мнению большинства 1: ЕСЛИ (Степень поджаристости = Совсем сырое еще) ТОГДА Готово по мнению большинства = Нет
Не готово по мнению большинства 2: ЕСЛИ (Степень поджаристости = Начинает поджариваться) ТОГДА Готово по мнению большинства = Нет
Не готово по мнению большинства 3: ЕСЛИ (Количество человек = Пять или больше) И (Степень поджаристости = Поджаристое) ТОГДА Готово по мнению большинства = Нет
  
```

В промышленных системах количество правил может исчисляться тысячами. Для нашей тренировочной области пока попробуем чуть меньше. Итак, запустим консультацию и посмотрим, что нам посоветует система.

Я сказал, что жарю свинину минут 10, и система посоветовала подождать ещё, и вот почему:

Объяснение

Сработавшие правила Все правила

- Свинина не готова 2
 - ЕСЛИ
 - Вид мяса = Свинина
 - Время жарки = Ну минут 10
 - ТО
 - Свинина готова = Нет
 - ТАК КАК
 - Свинину надо долго жарить.
- Сторона не готова
 - ЕСЛИ
 - Вид мяса = Свинина
 - Свинина готова = Нет
 - ТО
 - Сторона готова = Нет
 - ТАК КАК
 - Еще ничего не приготовилось.
- Ждать
 - ЕСЛИ
 - Сторона готова = Нет
 - ТО
 - Действие = Ждать
 - ТАК КАК
 - Если нижняя сторона еще не приготовилась, ждём.

----- Установленные факты -----
 Вид мяса = Свинина
 Время жарки = Ну минут 10
 Свинина готова = Нет
 Сторона готова = Нет
 Действие = Ждать

Когда смотришь на объяснение, как-то даже легче смириться с тем, что надо ещё ждать. Можно поэкспериментировать, запуская консультацию и по-разному отвечая на вопросы. Возможно, в каких-то случаях истина установлена не будет — значит, надо дополнить базу знаний новыми правилами.

Вот и всё, искусственный интеллект — система поддержки принятия решений в узкой предметной области — создан. Конечно, он не учитывает многие факторы, нечёткости и так далее. В базе знаний нет сведений о том, как захватить мир, и в отличие от некоторых промышленных систем, наша не способна анализировать консультации и сама генерировать новые правила. Но с шашлычными экспертами, которые будут кучковаться летом вокруг мангала, она уже вполне может поспорить.