

НОРМАТИВНЫЙ КОНСПЕКТ ПО ДИСЦИПЛИНЕ «ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ В ГУМАНИТАРНОЙ СФЕРЕ»

Часть 1

«ОСНОВНЫЕ ПОНЯТИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА»

Введение. *Интеллект* – это способность мыслить. *Искусственный интеллект* – это наука о создании мыслящих машин. В данных определениях остается неясным значение слова «мышление».

Мышление – это способность оперировать знаниями. Теперь можно говорить, что интеллект – это способность оперировать знаниями, а искусственный интеллект – это наука о создании машин, способных оперировать знаниями. После сказанного остается неясным значение слова «знание» и словосочетания «оперировать знаниями». Единого ответа на вопрос о значении этих формулировок в настоящее время нет. Но в разное время предложены и показали свою эффективность во многих областях человеческой деятельности несколько способов ответа на поставленный вопрос. Способы эти именуются моделями представления знаний.

Модель представления знаний – это способ ответа на вопрос, что такое знание и что означает оперировать знаниями. Наиболее известными сейчас являются пять моделей представления знаний:

- *алгоритмическая* модель представления знаний;
- *логическая* модель представления знаний;
- *продукционная* модель представления знаний (она является разновидностью логической модели);
- *фреймовая* модель представления знаний (в ряде разделов информатики и кибернетики ее называют *объектно-ориентированной* моделью);
- *синаптическая* модель представления знаний (ее еще называют *нейросетевой* моделью).

Понятие об алгоритмической модели представления знаний. Выдающийся советский математик и кибернетик Андрей Андреевич Марков в своей работе «Теория алгоритмов» так дает *неформальное* определение алгоритма: «Принято понимать под алгоритмом точное предписание, определяющее вычислительный процесс, ведущий от варьируемых исходных данных к искомому результату».

Таким образом, в алгоритмической модели представления знаний под *знанием* понимается *конкретное точное предписание*, а *оперирование знаниями* означает *применение этого точного предписания* к исходным данным с целью получения вполне конкретного искомого результата.

Для алгоритмов характерны три важнейших свойства:

- *определенность* – предписание должно быть точным, не оставляющим места произволу;
- *массовость* – алгоритм должен давать возможность исходить из варьируемых в известных пределах исходных данных;

- *результативность* – направленность алгоритма на получение некоторого искомого результата, в конце концов и получаемого при надлежащих исходных данных.

Рассмотрим для примера алгоритм нахождения наибольшего общего делителя (*НОД* (a, b)) двух натуральных чисел a и b . Придуман этот алгоритм выдающимся математиком древности Евклидом:

1. Символом a_1 обозначить большее из данных натуральных чисел:
 $a_1 = \max(a, b)$.

2. Символом a_2 обозначить меньшее из данных натуральных чисел:
 $a_2 = \min(a, b)$.

3. Символом a_3 обозначить остаток от деления a_1 на a_2 : $a_3 = a_1 \bmod a_2$.

4. Если $a_3 = 0$, то *НОД* (a, b) = a_2 , и работа алгоритма завершается. В противном случае символом a_4 обозначить остаток от деления a_2 на a_3 :
 $a_4 = a_2 \bmod a_3$.

5. Вообще, если $a_i = 0$, то *НОД* (a, b) = a_{i-1} , и работа алгоритма завершается. В противном случае символом a_{i+1} обозначить остаток от деления a_{i-1} на a_i :
 $a_{i+1} = a_{i-1} \bmod a_i$. Затем снова применить 5-й пункт алгоритма.

Пусть, например, требуется найти наибольший общий делитель натуральных чисел 42 и 64. Символом a_1 обозначается большее из данных натуральных чисел: $a_1 = 64$. Символом a_2 обозначается меньшее из данных натуральных чисел: $a_2 = 42$. Символом a_3 обозначается остаток от деления 64 на 42: $a_3 = 22$. Поскольку a_3 не равно нулю, то символом a_4 обозначается остаток от деления 42 на 22: $a_4 = 20$. Поскольку a_4 не равно нулю, то символом a_5 обозначается остаток от деления 22 на 20: $a_5 = 2$. Поскольку a_5 не равно нулю, то символом a_6 обозначается остаток от деления 20 на 2: $a_6 = 0$. Поскольку $a_6 = 0$, то *НОД* (42, 64) = $a_5 = 2$. Итак, наибольший общий делитель натуральных чисел 42 и 64 равен 2.

Алгоритм Евклида является *определенным*, так как на каждом шаге исполнителю точно ясно, что нужно делать, от исполнителя не требуется его волевых решений. Алгоритм Евклида является *массовым*, так как его входными данными являются любые два натуральных числа, а таких пар существует гораздо больше одной. Алгоритм Евклида является *результативным*, так как он направлен на получение вполне конкретного искомого результата (наибольшего общего делителя), в конце концов и получаемого при надлежащих исходных данных (если на вход поданы действительно два натуральных числа и если все предписания выполнены точно).

Часть кибернетики, изучающая построение мыслящих машин в рамках алгоритмической модели представления знаний, часто называется *традиционной информатикой*. Деятельность алгоритмически мыслящих машин в настоящее время даже не относят к машинному мышлению. Бесспорно следующее: результаты алгоритмического машинного мышления суть достижения не столько исполнителей (алгоритмически мыслящих машин), сколько носителей естественного интеллекта – людей, придумавших соответствующие алгоритмы и запрограммировавших названные машины.

Следует также отметить, что *неформальное* описание алгоритмов, использованное нами выше, не обладает математической точностью. Можно довольствоваться расплывчатым понятием, которое соответствует такому описанию. Это допустимо, пока термин «алгоритм» встречается в лишь в положительных высказываниях типа «для решения таких-то задач имеется алгоритм, и вот в чем он состоит». Однако, могут существовать задачи, относительно которых у нас складывается впечатление, что они *алгоритмически неразрешимы* (то есть алгоритма для их решения не существует). Никакие отрицательные результаты, никакие теоремы невозможности алгоритмов не могли бы быть *доказаны* с использованием неформального понятия алгоритма.

Свои варианты точного (*формального*, математического) определения понятия «алгоритм» предложили: выдающийся британский математик Алан Тьюринг (его формализация называется *машиной Тьюринга*), выдающийся советский математик Андрей Андреевич Марков (его формализация называется *нормальным алгоритмом*), выдающийся американский математик Эмиль Леон Пост (его формализация называется *машиной Поста*) и другие. Любой полноценный *алгоритмический язык* (язык программирования) является формализацией понятия «алгоритм». Примеры популярных сейчас языков программирования: *C / C++*, *Java*, *PHP*, *Python*, *Perl*, *JavaScript* и другие.

Понятие о логической модели представления знаний. Сущность логической модели представления знаний сводится к следующему:

1. Выделяются и *формализуются* (например, обозначаются буквами) *объекты* изучаемой *предметной области*. Например, если предметной областью является география, то в ней могут быть выделены такие объекты:

- объект «река», будет обозначаться символом R ;
- объект, обладающий свойством «подмывать свой восточный берег», будет обозначаться символом E ;
- объект «Волга», будет обозначаться символом V .

2. Выделяются и формализуются (например, задаются формулами) *отношения* между объектами изучаемой предметной области. В начатом выше примере: формула $X \text{ а } Y$ пусть означает, что каждый объект, обозначенный X , обладает свойством, определяемым объектом, обозначенным Y . Вместо X и Y могут подставляться R , E или V . Скажем, формула $R \text{ а } R$ интерпретируется так: «Каждая река есть река» (и это истинное утверждение). Формула же $R \text{ а } V$ означает «Каждая река есть Волга» (и это утверждение является ложным).

3. Знанием считается не любая *правильно построенная* формула, а только те из них, которые соответствуют истинным утверждениям. Фиксируются правильно построенные формулы, истинность утверждений соответствующих которым не будет далее ставиться под сомнение. Такие исходные знания называют *аксиомами*. В начатом нами примере пусть аксиомами будут:

- аксиома A_1 : $R \text{ а } E$ (что означает: «Все реки подмывают свой восточный берег»);
- аксиома A_2 : $V \text{ а } R$ (что означает: «Волга есть река»).

4. Задается одно или несколько *правил вывода*, позволяющих из уже имеющихся знаний выводить новые знания. Иными словами, знаниями считаются и аксиомы, и другие правильно построенные формулы, выведенные из аксиом по правилам вывода. Такие производные знания называют *теоремами*. При выводе новой теоремы можно использовать наряду с аксиомами и ранее уже выведенные (доказанные) теоремы. В развиваемом нами примере пусть будет принято следующее правило вывода Π_1 : если $S a M$ и одновременно $M a P$, то $S a P$. Интерпретировать это правило можно так: если каждый S есть M , и каждый M есть P , то верно и то, что каждый S есть P . При этом вместо S , M и P могут быть подставлены любые из объектов изучаемой предметной области: R , V или E . Кратко рассмотренное правило вывода с помощью формул (формально) может быть записано так:

$$\begin{array}{l} \Pi_1: M a P \\ \quad \underline{S a M} \\ \quad S a P \end{array}$$

Руководствуясь этим правилом вывода, можем доказать теорему T_1 : Волга подмывает свой восточный берег. Действительно: в силу аксиомы A_2 , каждый V есть R и (в силу аксиомы A_1) одновременно каждый R есть E . Следовательно, каждый V есть E , что означает «Волга подмывает свой восточный берег». А это и требовалось доказать. Кратко наше доказательство можно записать с помощью формул (формально) так:

$$\begin{array}{l} A_1: R a E \\ A_2: \underline{V a R} \\ \Pi_1: V a E (T_1) \end{array}$$

Следует заметить, что исходная база знаний не содержала сведений о том, что Волга подмывает свой восточный берег. Доказанная нами теорема содержит *новое* знание. Но для получения этого нового знания вовсе не обязательно понимать, что такое река, что такое берег, что такое восточный берег, что значит «подмывать» и тому подобное. Необходимо лишь хранить в памяти формулы-аксиомы, все ранее доказанные формулы-теоремы, формулы-правила вывода. И нужно еще уметь применять правила вывода к аксиомам и ранее доказанным теоремам. Иными словами, неуловимая «разумная» деятельность здесь заменяется алгоритмизируемыми действиями, формальными преобразованиями (что можно поручить и машине). Результат же оказывается ничуть не худшим, чем и в случае «разумной» деятельности.

Получается так, что в логической модели представления знаний реальная предметная область заменяется формально-символьной системой, называемой *исчислением*. Совокупность всех правильно построенных формул является *языком исчисления*. Под знаниями понимаются аксиомы исчисления и теоремы, которые

можно вывести из аксиом и ранее доказанных теорем по правилам вывода исчисления. «Оперировать знаниями» в рамках логической модели представления знаний означает «доказывать теоремы».

Возникновение логической модели представления знаний связывается с именем и трудами великого философа Аристотеля, жившего в Древней Греции в IV веке до нашей эры.

Часть кибернетики (и вообще математики), изучающая логическую модель представления знаний и возможности создания мыслящих машин на основании этой модели называется *математической логикой*. Упомянувшиеся выше пионеры кибернетики Тьюринг, Марков, Пост и многие другие им подобные (например, выдающийся советский украинский кибернетик Виктор Михайлович Глушков) были по своей основной специальности как раз исследователями в области математической логики. Математическая логика является одним из источников тех идей (наряду с физической электроникой, математической теорией управления и другими), которые привели к построению, развитию и широкому применению столь распространенных и популярных теперь вычислительных машин (компьютеров и компьютерных систем).

Понятие о продукционной модели представления знаний. Продукционная модель представления знаний является разновидностью логической модели. Она еще называется моделью представления знаний, *основанной на правилах*. Модель представления знаний и языки представления знаний, основанные на правилах (продукционные), являются наиболее распространенными в настоящее время. Большинство разработчиков экспертных систем (рассуждающих систем, консультирующих систем) используют именно их.

Продукционная модель или модель, основанная на правилах, позволяет представить знания в виде предложений типа «Если (условие), то (действие)».

Под «условием» (*антецедентом*) понимается некоторое предложение-образец, по которому осуществляется поиск в базе знаний, а под «действием» (*консеквентом*) – действия, выполняемые при успешном исходе поиска. Эти действия могут быть *промежуточными*, выступающими далее как условия, и *терминальными* или *целевыми*, завершающими работу системы.

Чаще всего вывод, основанный на сведениях, хранящихся в базе знаний, бывает *прямой* (от данных к поиску ответа на поставленный вопрос) или *обратный* (от гипотезы к исходным знаниям с целью подтверждения или опровержения гипотезы). Исходные знания – факты, хранящиеся в базе фактов, на основании которых запускается машина вывода или интерпретатор правил, перебирающий правила из продукционной базы знаний.

Продукционная модель часто применяется при разработке промышленных экспертных систем, поскольку привлекает разработчиков своей наглядностью, высокой модульностью, легкостью внесения дополнений и изменений и простотой механизма логического вывода.

Начало разработки и широкого внедрения как самой продукционной модели представления знаний, так и разрабатываемых на ее основе экспертных (консульти-

рующих, рассуждающих) интеллектуальных систем связывается с именем и трудами выдающегося американского ученого Эдварда Альберта Фейгенбаума.

Понятие о фреймовой модели представления знаний. Термин «фрейм», происходящий от английского «frame» (каркас, рамка) был предложен классиком искусственного интеллекта, выдающимся американским математиком и кибернетиком Марвином Минским для обозначения структуры знаний, возникающих при восприятии пространственных сцен.

Фрейм есть абстрактный образ для представления стереотипа объекта, понятия или ситуации. Интуитивно понятно, что под абстрактным образом понимается некоторая обобщенная и упрощенная модель или структура. Например, произнесение вслух слова «комната» порождает у слушающих образ комнаты: «жилое помещение с четырьмя стенами, полом, потолком, окнами и дверью, площадью 6-20 м²». Из этого описания ничего нельзя убрать (например, убрав окна, мы получим уже чулан, а не комнату), но в нем есть «дырки» или «слоты» – это незаполненные значения некоторых атрибутов. Например, количество окон, цвет стен, высота потолка, покрытие пола и другие.

В фреймовой модели представления знаний такой образ комнаты называется фреймом комнаты. Фреймом также называется и формализованная модель для отображения образа.

Различают фреймы-образцы или *прототипы*, хранящиеся в базе знаний, и *фреймы-экземпляры*, которые создаются для отображения реальных фактических ситуаций на основе поступающих данных. Модель фрейма является достаточно универсальной, поскольку позволяет отобразить все многообразие знаний о мире через:

- *фреймы-структуры*, использующиеся для обозначения объектов и понятий (метод обучения, средство обучения, форма организации учебной деятельности);

- *фреймы-роли* (педагог-преподаватель, обучающийся, педагог-менеджер);

- *фреймы-сценарии* (мозговой штурм, анализ конкретных ситуаций, выполнение заданий учебного проекта);

- *фреймы-ситуации* (возникновение дидактической проблемы, обнаружение факта тупиковости применяемой ветви решения, факт нахождения решения проблемы) и другие.

Традиционно структура фрейма может быть представлена как список свойств:

(ИМЯ ФРЕЙМА:

(имя 1-го слота: значение 1-го слота),

(имя 2-го слота: значение 2-го слота),

...

(имя n-го слота: значение n-го слота))

В качестве значения слота может выступать имя другого фрейма. Так образуются сети (*иерархии*) фреймов. Существует несколько способов получения слотом значений во фрейме-экземпляре: *по умолчанию* от фрейма-образца; через *наследование* свойств от фрейма, указанного как родительский в специально предназначенном для этого слоте; по формуле, указанной в слоте; путем применения присоединенной процедуры (*метода*); явно из диалога с пользователем; из базы знаний.

Важнейшим свойством фреймовой модели представления знаний является наследование свойств. Во фреймовой модели наследование происходит по АКО-связям (A-Kind-Of, род вот чего). Слот с именем АКО указывает на фрейм более высокого уровня иерархии (родительский фрейм), от которого неявно наследуются, то есть автоматически переносятся к фрейму-потомку, значения слотов, имена которых у потомка такие же, как и у фрейма-родителя. Наследование значений свойств может быть частичным: если значение слота во фрейме-потомке указано явно, то оно не замещается значением слота с тем же именем, взятым у фрейма-родителя.

Основным преимуществом фреймовой модели представления знаний является то, что она отражает концептуальную основу организации памяти человека. Важными достоинствами фреймовой модели являются ее гибкость и наглядность. Фреймовая модель представления знаний нашла широчайшее применение в современной парадигме *объектно-ориентированного* проектирования, моделирования и программирования.

Понятие о синаптической (нейросетевой) модели представления знаний. Основой работы интеллектуальных систем, основанных на синаптической модели представления знаний (нейросистем) является нейронная сеть, представляющая собой совокупность *нейронов* – элементов, связанных между собой определенным образом. Нейроны и межнейронные связи (*синапсы*) задаются программно на обычном компьютере или могут иметь «материальную» основу – особые микросхемы, применяемые в специально созданных *нейрокомпьютерах*. Функционирование нейрона в нейрокомпьютере или нейросистеме отдаленно напоминает работу биологического нейрона (живой нервной клетки).

Биологический нейрон (бионейрон) – это живая нервная клетка, имеющая длинные отростки, связанные с другими нейронами с помощью синапсов, передающих электрический импульс от одного нейрона к другому. Отростки подразделяются на дендриты, передающие сигналы к данному нейрону, и аксон, передающий сигнал от данного нейрона.

Функционирование формального (искусственного) нейрона происходит примерно так: в текущий момент времени нейрон получает по дендритам сигналы от других нейронов. Такие сигналы называются входными. Сигнал с каждого входа умножается на весовой коэффициент этого входа (весовые коэффициенты входов могут быть различными) и складывается с другими сигналами (также умноженными на весовые коэффициенты соответствующих

входов). В зависимости от полученного значения формируется выходной сигнал, передающийся по аксону другим нейронам.

Таким образом, нейронная сеть, получающая на входе некоторый сигнал, способна после прохода его по нейронам выдавать на выходе определенный ответ, который зависит от весовых коэффициентов всех нейронов. Поэтому, строго говоря, нейронная сеть моделирует не обязательно *интеллектуальную* систему. Нейронную сеть можно рассматривать как своеобразную формализацию понятия алгоритма наряду с такими «классическими» формализациями, как машина Тьюринга, машина Поста, нормальный алгоритм Маркова. Выражаясь точно, нужно отметить: не сама нейронная сеть может быть интеллектуальной, а реализуемый ею алгоритм. А уж если обнаружен интеллектуальный в некотором смысле алгоритм, то его можно выразить и без применения такой модели, как нейронная сеть. Именно это и делают, когда *идею* интеллектуального алгоритма излагают (для наглядности) с помощью нейронной сети, а *реализуют*, используя другую формализацию алгоритма, например, описывая его на распространенном алгоритмическом языке (языке программирования).

Между тем, исторически первые самообучающиеся алгоритмы были обнаружены и исследованы пионером искусственного интеллекта, выдающимся американским математиком и кибернетиком Френком Розенблаттом и его последователями при использовании именно синаптической модели представления знаний (середина XX века нашей эры). Обучение и самообучение нейронной сети есть не что иное, как «настраивание» весовых коэффициентов (синапсов) для того, чтобы определенному входному сигналу (классу входных сигналов) соответствовал определенный выходной сигнал.

Игра в имитацию (тест Тьюринга). Выдающимся британским математиком, логиком, кибернетиком, пионером искусственного интеллекта Аланом Тьюрингом в середине XX века нашей эры предложена процедура, которую сам автор называл *игрой в имитацию*, а теперь часто называют *тестом Тьюринга*. Тьюринг описал эту процедуру в статье «Может ли машина мыслить?»

Сущность процедуры такова. В одной комнате собираются *эксперты* – живые люди, относительно которых нет сомнений в том, что каждый из них является носителем интеллекта. Имеется и другая комната, в которой располагаются *испытуемые*. Все они – живые люди, кроме одного. Этот один – машина. Относительно живых испытуемых также нет сомнений относительно обладания ими интеллектом. Решается вопрос: обладает ли интеллектом машина?

Эксперты не знают имен испытуемых, но знакомы с испытуемыми по их номерам. В течение всего сеанса номер испытуемого не меняется.

Эксперты задают испытуемым вопросы, выдвигают перед ними задания. Посредники (записав, например на листе бумаги) передают задания испытуе-

мым, а затем сообщают экспертам, какой именно ответ или решение предложены каждым из испытуемых. Затем эксперты могут предлагать испытуемым новые задания.

Правильность выполнения заданий не играет роли. Задача каждого эксперта: указать тот номер, под которым «играет» испытуемый-машина. Если эксперты статистически значимым большинством голосов смогут «разоблачить» машину, отличить ее по ее ответам от живых испытуемых, тогда считается, что машина «проиграла»: ее способность мыслить (в той, конечно же, предметной области, которой касались вопросы и задания) не соответствует человеческому интеллекту, существенно отличается от него. Но если «поймать» машину не удастся – тогда придется признать, что машина оказалась способной мыслить (по крайней мере, в той предметной области, которой касались вопросы и задания).

Таким образом, Тьюринг предложил заменить ускользающий от корректного определения вопрос «Что такое мышление? Мыслит ли данное существо?» вопросом, на который ответ получить (хотя бы в принципе) возможно: «Могу ли я, опираясь на результаты интеллектуальной деятельности данного существа, отличить его от тех существ, в разумности которых и способности мыслить я не сомневаюсь?»

Следует отметить, что идея Тьюринга находится в русле так называемого бихевиоризма (поведенческой философии): индивид есть то, что нам дано в его деятельности, в его внешних проявлениях, в его результатах. Те же особенности «внутреннего мира» индивида, которые никак не проявляются снаружи, можно считать незначимыми, по крайней мере, с практической точки зрения.