

Системы искусственного интеллекта

Михаил Юрьевич Охтилев

Оглавление

1	Системы искусственного интеллекта и основные принципы их построения	3
1.1	Этапы развития систем искусственного интеллекта	3
1.1.1	Характеристика I этапа развития СИИ	3
1.1.2	Характеристика II этапа развития СИИ	4
1.1.3	Характеристика III этапа развития СИИ	5
1.2	Новая информационная система обработки информации	5
1.2.1	Основные черты традиционной (старой) обработки данных . .	5
1.2.2	Основная идея новой технологии обработки данных	6
1.2.3	Структура ВС в новой технологии обработки информации . .	7
2	Логические основы построения систем искусственного интеллекта	9
2.1	Формальные теории (логические исчисления)	9
2.2	Исчисление высказываний	10
2.2.1	Логические функции. Определение логической функции	12
2.2.2	Разложение логических функций по переменным	13
2.2.3	Булева алгебра. Свойства булевой алгебры	14
2.2.4	Эквивалентные преобразования в булевой алгебре	15
2.2.5	Интерпретация формул исчисления высказываний	16
2.2.6	Логические следствия в исчислении высказываний	16
2.3	Исчисление предикатов или теории первого порядка	17
2.3.1	Определение исчисления предикатов	17
2.3.2	Интерпретация формул исчисления предикатов	19
2.3.3	Эквивалентные преобразования в исчислении предикатов	19
2.3.4	Нормальная форма в исчислении предикатов	20
2.3.5	Доказательство теорем в исчислении предикатов	21
2.4	Метод резолюций и его использование в системах искусственного ин- теллекта	21
2.4.1	История поиска процедур доказательства теорем	21
2.4.2	Метод резолюций для исчисления высказываний	22
3	Экспертные системы	24
3.1	Введение в экспертные системы	24
3.2	Структура экспертных систем	26
3.2.1	ЭС статического типа	26
3.2.2	ЭС динамического типа	26

3.3	Интеллектуальные информационные технологии и развитие аппарата знаний	27
3.4	Модели представления знаний (МПЗ)	27
3.5	Аппарат знаний и его влияние на интеллектуализацию информационной технологии	28
3.6	Смена парадигмы разработки интеллектуальных информационных систем	29
3.6.1	История развития информационных технологий	29
3.6.2	Смена влияния моделей и алгоритмов на развитие информационных технологий	30
А	Программный комплекс мониторинга состояний и управления сложными техническими объектами в реальном масштабе времени	32
A.1	Назначение, характеристики и структура информационных систем мониторинга состояния сложных технических объектов	32
A.2	Характеристика основных элементов информационной технологии, которую реализует программный комплекс	33
A.2.1	Назначение информационной технологии, реализуемой программным комплексом	33
A.2.2	Характеристика унифицированного типового модуля автоматизации (УТМА)	33
A.3	Режимы распределенной иерархической обработки данных	35
A.4	Основные технологических элементы информационной технологии обработки данных	36

Глава 1

Системы искусственного интеллекта и основные принципы их построения

1.1 Этапы развития систем искусственного интеллекта

Появление систем искусственного интеллекта (СИИ) и начало исследований в области СИИ началось с конца 50-х годов с появлением работ Ньюэла, Шоу.

3 этапа развития СИИ:

I этап: середина 50-х—середина 60-х;

II этап: середина 60-х—середина 70-х;

III этап: с середины 70-х и до наших дней.

1.1.1 Характеристика I этапа развития СИИ

I этап: Разработка и создание СИИ, решающих задачи на основе эвристических методов.

Эвристический метод — свойственный человеческому мышлению метод, для которого характерно использование догадок о путях решения задач с последующей их проверкой.

Противоположность «алгоритму» — «эвристика».

Эвристические методы используются когда нельзя установить жёсткие условия. Первоначально широкое распространение получили разработки, моделирующие мыслительную деятельность человека.

Область применения СИИ (I этап):

- игры (разбиение задачи на классы, моделирование на 2-3 шага вперёд);
- головоломки;
- математические задачи познавательного характера;
- и др.

Эти области применения характеризовались простотой и ясностью предметной области (проблемной среды), её относительно малой громоздкостью, возможностью прямого перебора или возможностью подбора под модель.

1.1.2 Характеристика II этапа развития СИИ

II этап: Разработка и создание СИИ, ориентированных на применение интегральных роботов.

При переходе на I этапе от искусственных предметных областей (ПрО) к реальным разработчики стали сталкиваться с большими трудностями, обусловленными необходимостью моделирования реального мира:

- описание знаний о внешнем мире;
- организация хранения этих знаний;
- эффективный поиск и доступ к знаниям;
- проверка знаний на корректность (полнота, непротиворечивость, верифицируемость и пр.).

Практическое применение СИИ привело к созданию «интегральных роботов», которые привели к необходимости рассмотрения и использования реальных ПрО.

Переход к реальным ПрО обусловлен переходом от исследований по моделированию способов мышления человека к разработке программ, способных решать «человеческие задачи», но формальным методом (математическая логика, смысловая логика). К этому периоду относится разработка метода автоматического доказательства теорем на основе метода резолюций (1969 г., Робинсон, Минский, Пайперт).

Для СИИ II этапа характерно:

- наличие формальной модели предметной области;
- наличие алгоритма распознавания ситуаций, изображений, сцен;
- алгоритм принятия решений;
- алгоритмы планирования работы интегральных роботов;
- алгоритм оценки качества выполненных/спланированных работ;
- и др.

Появление первых экспериментальных образцов роботов, решающих свои задачи в комплексе показало необходимость решения фундаментальных проблем, связанных со следующими задачами:

- представление знаний;
- зрительное восприятие (распознавание образов);
- построение сложных планов поведения в динамических средах;

- общение с роботами на естественном языке;
- и др.

Всё это привело к III этапу развития СИИ.

1.1.3 Характеристика III этапа развития СИИ

III этап: Разработка и создание СИИ, характерной чертой которых явился переход от создания автономно функционирующих систем, самостоятельно решающих задачу в реальной среде, к созданию человеко-машинных систем, совмещающих достоинства интеллекта человека и возможности ЭВМ для достижения общей цели решения задачи, поставленной перед интегральной человеко-машинной решающей системой.

Причины возникновения человеко-машинных систем:

1. К этому времени выяснилось, что даже простые задачи функционирования интегрального робота в реальных ПрО не могут быть решены методами, разработанными для роботов в экспериментальных ПрО.
2. Стало ясно, что сочетание дополняющих друг друга способностей человека и возможностей ЭВМ позволяет обойти нерешаемые задачи (острые углы) путём перекладывания на человека тех функций, которые пока недоступны для ЭВМ.

СИИ — система, в которой:

- развиваются возможности ЭВМ в направлении обеспечения совместного с пользователем решения задач;
- упрощаются процессы общения человека и ЭВМ в ходе решения задач;
- постоянно расширяется доля компьютеров в совместной с человеком деятельности по решению задач;
- значительное внимание уделяется повышению способности ЭВМ к самостоятельному решению (в автоматическом режиме) трудно решаемых задач.

1.2 Новая информационная система обработки информации

1.2.1 Основные черты традиционной (старой) обработки данных

- Наличие следующих этапов обработки информации:
 1. Специалист (эксперт) — конечный пользователь.
 2. Системный аналитик.
 3. Программист.

4. Вычислительная система.

- Большая трудоёмкость решения конкретной задачи обработки информации, связанная с имеющимися место на каждом этапе:
 - ошибками;
 - неточностями;
 - нерациональными решениями;
 - необходимостью внесения изменений в схемы решения, методы решения, постановку задачи, программу.
- Сложность сопровождения разработанного программного обеспечения (ПО).
- ПО в традиционной технологии основывается на формальной (математической) модели задач и представления данных, в то время как каждая конкретная предметная область основывается на системе, содержательных понятий, которыми оперирует пользователь при решении задач.

Это значит, что в такой схеме решения система понятий ПрО и формальной модели не совпадают.
- При формулировке задачи конечный пользователь должен перевести постановку задачи, выраженной в системе ПрО в постановку задачи формальной модели, при получении результата — наоборот.

1.2.2 Основная идея новой технологии обработки данных

Система понятий конкретной предметной области рассматривается как исходная информация для решения прикладных задач, при этом обеспечивается автоматическая интерпретация системы понятий формальной модели.

Для новой технологии характерна следующая цепочка обработки данных:

1. Специалист (эксперт) — конечный пользователь;
2. Содержательная постановка задачи;
3. Вычислительная система.

В новой технологии обработки информации вычислительной системе должна задаваться только постановка задачи в виде описания требуемого результата и условий его получения, в то время как последовательность операций, посредством которых она решается, определяется решающей задачей системой.

Вычислительная система в составе человеко-машинной системы превращается из пассивного звена в активную целенаправленную систему, целью которой является получение требуемого в постановке задачи результата или знания.

Получение этого знания достигается:

- автоматическим синтезом (генерацией) вычислительной системы;
- выполнением оптимальной в определённом смысле последовательности вычислительных, логических и поисковых операций над имеющимися знаниями.

1.2.3 Структура ВС в новой технологии обработки информации

Вычислительная система:

1. Модель языка пользователя.
2. Модель представления знаний.
3. База знаний.
4. Интеллектуальный интерфейс:
 - система общения;
 - решатель.
5. Исполнительная система.

Система программных и аппаратных средств, обеспечивающих для конечного пользователя (КП), не имеющего специальной подготовки в области вычислительной техники использования ЭВМ для решения задач, возникающих в сфере профессиональной деятельности либо без посредников-программистов либо с незначительной их помощью, называется интеллектуальным интерфейсом.

Процесс же внедрения средств интеллектуального интерфейса в вычислительную технику определяется как интеллектуализацию ЭВМ.

Под термином *интеллектуализация* понимается (подчеркивается) то, что с одной стороны имеется нетривиальность функций, выполняемых этим интеллектуальным интерфейсом и с другой стороны то, что выполнение этих функций до последнего времени является прерогативой человека.

Система общения — совокупность средств, осуществляющих трансляцию с языка пользователя (конечного пользователя) на язык представления знаний в базе знаний (БЗ) и включающую в себя средства трансляции и средства, обеспечивающие понимание.

Решатель — совокупность средств, обеспечивающих в диалоге с пользователем автоматический синтез программы решения задач, т.е. анализ условий задачи, выделение подзадач, имеющих стандартные решения, и объединение этих подзадач в единую целостную программу. Функционирование системы, имеющей в своем составе решатель, может осуществляться в 2-х режимах: в режиме компиляции и режиме интерпретации.

Исполнительная система представляет собой совокупность средств, выполняющих (исполняющих) синтезированную программу вычислений, сформированную с позиции эффективного решения задач и имеет проблемную ориентацию.

База знаний занимает центральное положение по отношению к остальным компонентам вычислительной системы. Через БЗ осуществляется интеграция всех средств вычислительной системы, участвующих в решении конкретных задач.

Базу знаний для вычислительных систем, ориентированных на использование новой технологии обработки данных целесообразно строить как 2-уровневую структуру, включающую 2 компонента:

1. концептуальную БЗ (верхний концептуальный уровень);
2. базы данных (БД) (нижний информационный уровень).

Такое деление позволяет обеспечить эффективное представление обобщенных знаний и метазнаний на верхнем концептуальном уровне и конкретной информации (данных) на нижнем уровне.

Использование баз данных в качестве отдельного компонента баз знаний позволяет использовать современные СУБД для описания взаимодействий (взаимоотношений) между объектами проблемной среды, причем на уровне конкретных фактов.

Язык представления знаний — это конкретный способ описания знаний в проблемной среде, задаваемый синтаксисом описания и правилами соотнесения языковых выражений с конкретными объектами проблемной области или интерпретацией.

Глава 2

Логические основы построения систем искусственного интеллекта

2.1 Формальные теории (логические исчисления)

Определение формальной теории (ФТ): любая формальная теория оперирует с 2 объектами:

1. языком (определенным множеством высказываний, имеющих смысл с точки зрения этой теории);
2. совокупностью теорем (подмножеством языка и состоящих из высказываний, истинностных данной теории).

Формальная система:

$$FS = \langle A, A_1, R \rangle ,$$

где A — алфавит, A_1 — аксиомы, R — правила вывода.

Формальная теория (исчисление) определяется следующим кортежем:

$$FT = \langle U, L, S, R \rangle ,$$

где U — алфавит формальной теории, L — язык формальной теории, S — аксиомы формальной теории, R — правила вывода.

Замечание (отличие формальной теории от формальной системы). Аксиомы и теоремы формальной системы (ФС) трактуются как некоторые формулы или предложения или как правильно построенные формулы (ППФ) без придания им какого-либо смысла. Это означает, что в формальных системах правильно построенные формулы сводятся к рассмотрению их структур или синтаксических свойств соответствующих предложений.

Рассмотрим более подробно элементы формальной теории:

U — алфавит формальной теории:

- символы предметных констант $\{a, b, c, \dots\}$;
- символы предметных переменных $\{x, y, z, \dots, x_1, y_1, z_1, \dots\}$;

- символы функциональных констант $\{f_1^{n1}, f_2^{n2}, f_3^{n3}, \dots\}$;
- символы предикатных констант $\{A, B, C, \dots, X, Y, Z, \dots\}$;
- символы предикатных переменных $\{A, B, C, \dots\}$;
- символы логических связок и кванторов $\{\vee, \&, \neg, \supset, \exists, \forall, \dots\}$;
- вспомогательные символы $\{ , , . ,) , (\}$.

L — язык формальной теории; определяется индивидуально, с помощью различных конструктивных процедур и, как правило, рекурсивно.

S — аксиомы, $S \subseteq L$.

Отличие формальной теории от формальных систем: каждой правильно построенной формуле (ППФ) может быть сопоставлено некоторое истинностное значение в каждой конкретной интерпретации.

В формальной теории различают следующие виды ППФ:

1. *Общезначимые* формулы, или тавтологии, — являются тождественно истинностными во всех интерпретациях.
2. *Выполнимые* формулы — принимают истину или ложь в зависимости от интерпретации.
3. *Противоречивые* — ложны во всех интерпретациях.

R — правила вывода.

Обозначают $F_1, F_2, \dots, F_n \vdash G$ или $\frac{F_1, F_2, \dots, F_n}{G}$, где $F_1, F_2, \dots, F_n, G \in L$

Выводом некоторой формулы B из $A_1, A_2, \dots, A_m$ называется такая последовательность формул $F_1, F_2, \dots, F_n$, что $F_m = B$, а каждая F_i есть:

1. Аксиома.
2. Форма посылок ($F_i \in \{A_1, A_2, \dots, A_m\}$).
3. Выводима ($F_i \vdash F_1, F_2, \dots, F_{i-1}$).

Этот факт записывается $A_1, A_2, \dots, A_m \vdash B$. В этом случае $A_1, A_2, \dots, A_m$ — посылки (гипотезы) вывода, B — теорема (вывод).

Доказательством формулы B в формальной теории называется вывод этой формулы из пустого множества формул. В этом случае посылками являются только аксиомы. В таком случае говорят, что формула B выводима (доказуема) в формальной теории.

2.2 Исчисление высказываний

Определение формальной теории исчисления высказываний: $FT = \langle U, L, S, R \rangle$,

U :

- переменные высказываний (пропозициональные символы): $A, B, C, \dots$;

- символы логических связок: $\vee, \&, \supset, \neg$;
- дополнительные символы;
- иные способы задания U нет.

L :

- высказывательная переменная является ППФ;
- если $\mathcal{A}$ и $\mathcal{B}$ — ППФ, то такими формулами являются также $\mathcal{A} \vee \mathcal{B}, \mathcal{A} \& \mathcal{B}, \neg \mathcal{B}, \dots$
- других способов задания ППФ нет.

Замечание. Формулы, состоящие из единственного символа или высказывательной переменной, называются *атомарными* (элементарными).

S :

S_I :

1. $A \supset (B \supset A)$;
2. $(A \supset B) \supset ((A \supset (B \supset C)) \supset (A \supset C))$;
3. $(A \& B) \supset A$;
4. $(A \& B) \supset B$;
5. $A \supset (B \supset (A \& B))$;
6. $A \supset (A \vee B)$;
7. $B \supset (A \vee B)$;
8. $(A \supset C) \supset ((B \supset C) \supset ((A \vee B) \supset C))$;
9. $(A \supset B) \supset ((A \supset \neg B) \supset \neg A)$;
10. $\neg \neg A \supset A$.

S_{II} :

1. $A \supset (B \supset A)$;
2. $(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$;
3. $(\neg A \supset \neg B) \supset ((\neg A \supset B) \supset A)$;
- *. $A \vee B = \neg A \supset B$;
- *. $A \& B = \neg(A \supset \neg B)$.

Замечание. Системы аксиом S_1 и S_2 равнозначны (эквивалентны, т. е. порождают одно и то же множество ППФ. Выбор той или иной системы определяется конкретными задачами. Для S_1 выводы будут короче, но она более богатая; S_2 — более обозримая, более компактная.

R :

1. *Правило подстановки.* Если $\mathcal{A}$ — некоторая выводимая формула $\mathcal{A}(A)$, то выводима и форма, получающаяся из первой заменой всех вхождений A на формулу $\mathcal{B}$: $\mathcal{A}(\mathcal{B})$. Обозначение:

$$\frac{\mathcal{A}(A)}{\mathcal{A}(\mathcal{B})} \quad \mathcal{A}(A) \vdash \mathcal{A}(\mathcal{B}).$$

2. *Modus Ponens (MP)* — правило заключения. Если $\mathcal{A}$ и $\mathcal{A} \supset \mathcal{B}$ — выводимые формулы, то выводима и формула $\mathcal{B}$. Обозначение:

$$\frac{\mathcal{A}, \mathcal{A} \supset \mathcal{B}}{\mathcal{B}} \quad \mathcal{A}, \mathcal{A} \supset \mathcal{B} \vdash \mathcal{B}.$$

3. *Теорема дедукции*. Пусть некоторая Γ — множество формул, $\mathcal{A}, \mathcal{B}$ — тоже формулы, тогда если

$$\Gamma, \mathcal{A} \vdash \mathcal{B} \Rightarrow \Gamma \vdash \mathcal{A} \supset \mathcal{B},$$

$$\frac{\mathcal{A}}{\mathcal{A} \supset \mathcal{B}}, \quad \text{где } \mathcal{B} \text{ — любая ППФ.}$$

Замечание. Система аксиом S и правила R (вместе с U и L) полностью определяют множество всех ППФ исчисления высказываний.

Пример (вывода в исчислении высказываний (ИВ)). Покажем выводимость формулы, воспользовавшись S_{II} :

1. $\frac{II.2(B)}{II.2(A \supset A)}, \frac{II.2(C)}{II.2(A)}$:
 $\vdash \left(A \supset ((A \supset A) \supset A) \right) \supset \left((A \supset (A \supset A)) \supset (A \supset A) \right).$
2. $\frac{II.1(B)}{II.1(A \supset A)}$: $\vdash A \supset ((A \supset A) \supset A).$
3. MP 1, 2: $\vdash (A \supset (A \supset A)) \supset (A \supset A).$
4. $\frac{II.1(B)}{II.1(A)}$: $\vdash A \supset (A \supset A).$
5. MP 3, 4: $\vdash (A \supset A).$

2.2.1 Логические функции. Определение логической функции

Рассмотрим некоторое множество $B_2\{0, 1\}$, а также различные операции, которые определены на этом множестве. Таким образом можно сформулировать алгебру логических функций.

Алгебра, образованная множеством B_2 вместе со всеми возможными операциями на нём называется *алгеброй логики*.

Функцией алгебры логики от n переменных называется n -арная операция на множестве B_2 .

$$\begin{cases} f^{(n)}(x_1, x_2, \dots, x_n) \in \{0, 1\}, \\ \forall x_1, x_2, \dots, x_n \in \{0, 1\}. \end{cases}$$

$$f^{(n)}: \underbrace{B_2 \times B_2 \times \dots \times B_2}_{n \text{ раз}} \rightarrow B_2 \quad \Rightarrow \quad f^{(n)}: B_2^n \rightarrow B_2.$$

Все функции $f^{(n)}$ объединяются множеством $P_2 = \{f^{(n)}\}$.

Алгебра, образованная k -элементным множеством B_k , $k = 1, 2, \dots$ вместе со всеми операциями на нём, называется *алгеброй k -значной логики*. n -арные операции на k -элементном множестве B_k , называются k -значными логическими функциями. Множество всех k -значных логических функций $\{f_k^{(n)}\} = P_k$.

Логические функции одной переменной

$n = 1$.

x	φ_0	φ_1	φ_2	φ_3
0	0	0	1	1
1	0	1	0	1

$$\varphi_0(x) = 0, \quad \varphi_1(x) = x, \quad \varphi_2(x) = \overline{\varphi_2}, \quad \varphi_3(x) = 1.$$

Логические функции 2-х переменных

$n = 2$

x_1	x_2	φ_0	φ_1	φ_2	φ_3	φ_4	φ_5	φ_6	φ_7	φ_8	φ_9	φ_{10}	φ_{11}	φ_{12}	φ_{13}	φ_{14}	φ_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

$$\begin{aligned} \varphi_0(x_1, x_2) &= 0, & \varphi_1(x_1, x_2) &= x_1 \& x_2, & \varphi_2(x_1, x_2) &= \overline{x_1 \rightarrow x_2}, \\ \varphi_3(x_1, x_2) &= x_1, & \varphi_4(x_1, x_2) &= \overline{x_1 \leftarrow x_2}, & \varphi_5(x_1, x_2) &= x_2, \\ \varphi_6(x_1, x_2) &= x_1 \oplus x_2, & \varphi_7(x_1, x_2) &= x_1 \vee x_2, & \varphi_8(x_1, x_2) &= x_1 \downarrow x_2, \\ \varphi_9(x_1, x_2) &= x_1 \sim x_2, & \varphi_{10}(x_1, x_2) &= \overline{x_2}, & \varphi_{11}(x_1, x_2) &= x_1 \leftarrow x_2, \\ \varphi_{12}(x_1, x_2) &= \overline{x_1}, & \varphi_{13}(x_1, x_2) &= x_1 \rightarrow x_2, & \varphi_{14}(x_1, x_2) &= x_1 \mid x_2, \\ \varphi_{15}(x_1, x_2) &= 1. \end{aligned}$$

2.2.2 Разложение логических функций по переменным

$$\left. \begin{aligned} x^0 &= \overline{x}, \\ x^1 &= x, \\ \alpha &\in 0, 1 = B_2. \end{aligned} \right\} \Rightarrow \begin{aligned} x^\alpha &= 1, \quad \text{если } x = \alpha, \\ x^\alpha &= 0, \quad \text{если } x \neq \alpha. \end{aligned}$$

Теорема (о разложении функций по переменным). *Всякая логическая функция $f(x_1, x_2, \dots, x_n)$ может быть представлена в виде:*

$$\bigvee_{\langle \alpha_1, \alpha_2, \dots, \alpha_m \rangle \subseteq B_2^m} x_1^{\alpha_1} x_2^{\alpha_2} \dots x_m^{\alpha_m} f(\alpha_1, \alpha_2, \dots, \alpha_m, x_{m+1}, \dots, x_n), \quad (2.1)$$

где $m \leq n$, и дизъюнкция берется по всем 2^m наборам $\langle \alpha_1, \alpha_2, \dots, \alpha_m \rangle$ значений переменных $x_1, x_2, \dots, x_n$.

Пример. Разложим логическую функцию $f(x_1, x_2, x_3, x_4)$ по переменным x_1, x_2 , $m = 2$:

$$\begin{aligned} f(x_1, x_2, x_3, x_4) &= \\ &= x_1^0 \cdot x_2^0 \cdot f(0, 0, x_3, x_4) \vee x_1^0 \cdot x_2^1 \cdot f(0, 1, x_3, x_4) \vee x_1^1 \cdot x_2^0 \cdot f(1, 0, x_3, x_4) \vee x_1^1 \cdot x_2^1 \cdot f(1, 1, x_3, x_4) = \\ &= \overline{x_1} \cdot \overline{x_2} \cdot f(0, 0, x_3, x_4) \vee \overline{x_1} \cdot x_2 \cdot f(0, 1, x_3, x_4) \vee x_1 \cdot \overline{x_2} \cdot f(1, 0, x_3, x_4) \vee x_1 \cdot x_2 \cdot f(1, 1, x_3, x_4) \end{aligned}$$

Доказательство. Подставим в обе части равенства (2.1) произвольный набор $\langle \sigma_1, \sigma_2, \dots, \sigma_n \rangle$. Т.к. $x_i^{\alpha_i} = 1$, то среди 2^m конъюнкций $x_1^{\alpha_1} x_2^{\alpha_2} \dots x_m^{\alpha_m}$ в единицу обратится только одна, причём та, в которой $x_1^{\alpha_1} = x_2^{\alpha_2} = \dots = x_m^{\alpha_m} = 1$, т.е. в которой $x_1 = \alpha_1, x_2 = \alpha_2, \dots, x_m = \alpha_m$. Это означает, что:

$$f(\sigma_1, \sigma_2, \dots, \sigma_n) = \sigma_1^{\alpha_1} \sigma_2^{\alpha_2} \dots \sigma_m^{\alpha_m} f(\sigma_1, \sigma_2, \dots, \sigma_m, x_{m+1}, \dots, x_n).$$

□

Следствие (СДНФ). *Разложение логической функции по всем переменным и формирование СДНФ:*

$$f(x_1, x_2, \dots, x_n) = \bigvee_{f(\sigma_1, \sigma_2, \dots, \sigma_n)=1} x_1^{\sigma_1} x_2^{\sigma_2}, \dots, x_n^{\sigma_n} \quad (2.2)$$

Пример. $f(x_1, x_2, x_3)$:

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

$$f(x_1, x_2, x_3) = \overline{x_1} \overline{x_2} x_3 \vee x_1 \overline{x_2} \overline{x_3} \vee x_1 \overline{x_2} x_3.$$

Следствие (СКНФ). *Разложение логической функции по всем переменным и формирование СКНФ:*

$$f(x_1, x_2, \dots, x_n) = \bigwedge_{f(\sigma_1, \sigma_2, \dots, \sigma_n)=0} x_1^{\sigma_1} x_2^{\sigma_2}, \dots, x_n^{\sigma_n} \quad (2.3)$$

2.2.3 Булева алгебра. Свойства булевой алгебры

Определение (булевой алгебры).

$$A = (P_2; \vee, \&, \neg)$$

1. Ассоциативность относительно:

- конъюнкции: $x_1 \& (x_2 \& x_3) = (x_1 \& x_2) \& x_3$;
- дизъюнкции: $x_1 \vee (x_2 \vee x_3) = (x_1 \vee x_2) \vee x_3$.

2. Коммутативность относительно:

- конъюнкции: $x_1 \& x_2 = x_2 \& x_1$;
- дизъюнкции: $x_1 \vee x_2 = x_2 \vee x_1$.

3. Дистрибутивность:

- конъюнкции относительно дизъюнкции: $x_1 \& (x_2 \vee x_3) = x_1 \& x_2 \vee x_1 \& x_3$;
- дизъюнкции относительно конъюнкции: $x_1 \vee (x_2 \& x_3) = (x_1 \vee x_2) \& (x_1 \vee x_3)$.

4. Идемпотентность:

- конъюнкции: $x \& x = x$;
- дизъюнкции: $x \vee x = x$.

5. $\neg\neg x = x$.

6. Свойство констант:

- $x \& 1 = x$;
- $x \& 0 = 0$;
- $x \vee 1 = 1$;
- $x \vee 0 = x$;
- $\neg 0 = 1$.

7. Правила де Моргана:

- $\overline{x_1 \& x_2} = \overline{x_1} \vee \overline{x_2}$;
- $\overline{x_1 \vee x_2} = \overline{x_1} \& \overline{x_2}$.

8. Закон противоречия: $x \& \bar{x} = 0$.

9. Закон исключения 3-его: $x \vee \bar{x} = 1$.

2.2.4 Эквивалентные преобразования в булевой алгебре

1. Свойство поглощения:

- $x \vee xy = x$;
- $x \& (x \vee y) = x$.

2. Свойство склеивания и расщепления: $x \& y \vee x\bar{y} = x$.

3. Обобщённое склеивание: $xz \vee y\bar{z} \vee xy = xz \vee y\bar{z}$.

4. $x \vee \bar{x}y = x \vee y$.

5. Обобщение: $x_1 \vee f(x_1, x_2, \dots, x_n) = x_1 \vee f(0, x_2, \dots, x_n)$.

2.2.5 Интерпретация формул исчисления высказываний

Как уже рассматривалось ранее, в исчислении высказываний каждой формуле должно быть приписано истинностное значение или элемент множества:

$$\{, \} = \{T, F\} = \{1, 0\}.$$

Приписывая каждому атомарному символу из формулы F истинностные значения, мы будем получать и различные истинностные значения и самой формулы F .

Таблицу, в которой указаны истинностные значения F при всевозможных истинностных значениях входящих атомов, назовём *таблицей истинности*.

Пример. $F = (A \vee B) \supset C$

A	B	C	$A \vee B$	F
0	0	0	0	1
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	0
1	1	1	1	1

Определение (интерпретации). Пусть F — некоторая формула исчисления высказываний, $A_1, A_2, \dots, A_n$ — её атомарные символы. *Интерпретацией* F является такое приписывание истинностных значений атомам $A_1, A_2, \dots, A_n$, при котором каждому A_i приписывается либо 1 либо 0.

Говорят, что F истинна при некоторой интерпретации тогда и только тогда, когда F получает значение 1 в этой интерпретации; в противном случае F ложна при этой интерпретации.

2.2.6 Логические следствия в исчислении высказываний

В математической логике следует различать понятия выводимости и логического следования.

Пример (логическое следование). Событие A — студент пришёл на экзамен. B — вытаскил билет, ответ на который он знает. C — экзамен сдан успешно.

$$A \& B \Rightarrow C, \quad C \text{ — логическое следование.}$$

Определение. Пусть даны формулы: $F_1, F_2, \dots, F_n, G$. Говорят, что G — логическое следствие $F_1, F_2, \dots, F_n$ или G следует из $F_1, F_2, \dots, F_n$ тогда и только тогда, когда для всякой интерпретации I , в которой $I: F_1 \& F_2 \& \dots \& F_n = 1$, $G = 1$. В этом случае говорят, что $F_1, F_2, \dots, F_n$ — посылки, а G — следствие, и обозначают этот факт

$$F_1, F_2, \dots, F_n \models G. \quad (2.4)$$

Теорема. Пусть даны $F_1, F_2, \dots, F_n$ и G , тогда G есть логическое следствие формул $F_1, F_2, \dots, F_n$ тогда и только тогда, когда формула $(F_1 \& F_2 \& \dots \& F_n) \supset G$ является общезначимой.

Теорема. Пусть даны $F_1, F_2, \dots, F_n$ и G . Формула G есть логическое следствие формул $F_1, F_2, \dots, F_n$ тогда и только тогда, когда $F_1 \& F_2 \& \dots \& F_n \& \bar{G}$ противоречива.

Пример. $F_1: P \supset Q$, $F_2: \bar{Q}$, $G: \bar{P}$, где P и Q — атомы. Доказать, $F_1, F_2 \models G$.

P	Q	$P \supset Q$	$\bar{Q}$	$F_1 \& F_2$	$\bar{P}$	$F_1 \& F_2 \supset G$	$F_1 \& F_2 \& \bar{G}$
0	0	1	1	1	1	1	0
0	1	1	0	0	1	1	0
1	0	0	1	0	0	1	0
1	1	1	0	0	0	1	0

2.3 Исчисление предикатов или теории первого порядка

2.3.1 Определение исчисления предикатов

Предикатом $P(x_1, x_2, \dots, x_n)$ называется функция P , ставящая в соответствие $M^m \rightarrow B$ или $P: \underbrace{M \times M \times \dots \times M}_{m \text{ раз}} \rightarrow B$, где M — произвольное множество, в

том числе и бесконечное, называемое предметной областью, $B = \{0, 1\}$ — булево множество, $x_1, x_2, \dots, x_m$ — предметные переменные.

Другими словами, m -местный предикат, определенный на множестве M — это двухзначная функция от m аргументов, принимающих значение в произвольном множестве M .

Таким образом, видно, что предикат отличается от высказывания наличием предметных переменных, которые могут принимать значения из некоторой предметной области. Другим отличием предикатов является наличие кванторов.

Пример. $\exists x P(x)$, $\forall x \bar{P}(x)$.

Определение. Переход от предиката $P(x)$

$$P(x) \rightarrow \begin{cases} \exists x P(x) \\ \forall x P(x) \end{cases}$$

называется операцией *связывания переменной*, или операцией *квантификации*, а переменная x — связанной. Если переменная в предикате не связана, то она свободная.

Определение. Выражение или формула, на которые распространяется действие квантора, называется *областью действия квантора*.

$$FT = \langle U, L, S, R \rangle$$

Алфавит исчисления предикатов включает:

- $X = \{x_i\}$ — множество предметных переменных $x_i \in M$.
- $A = \{a_j\}$ — множество предметных констант $a_j \in M$.
- $P = \{P_1^1, P_2^1, \dots, P_k^l, \dots\}$ — множество предикатных букв, где l — арность (местность) k -го предиката.
- $F = \{f_1^1, f_2^1, \dots, f_k^l, \dots\}$ — множество функциональных букв, где l — арность (местность) k -ой функции.
- знаки логических связок: $\vee, \&, \neg, \supset, \dots$
- кванторы: $\exists, \forall, \dots$
- служебные: $. , () \dots$

Язык определяется в 2 этапа:

1. Термы (элементы M), $M \rightarrow B$:
 - $x_i \in X, a_j \in A$;
 - если $f^n \in F$ — функциональная буква, $t_1, t_2, \dots, t_n$ — термы, то термом также является $f^n(t_1, t_2, \dots, t_n)$.
2. Формулы:
 - если $p^l \in P$, а $t_1, t_2, \dots, t_l$ — термы, то $P^l(t_1, t_2, \dots, t_l)$ — формула;
 - если F_1, F_2 — формулы, то формулами также являются $F_1 \vee F_2, \overline{F_1}, \overline{F_2}, \dots$
Замечание. Все переменные, входящие в таким образом построенные формулы, являются *свободными*.
 - если $F(x)$ — некоторая формула, в которой x — свободная переменная, то $\exists x F(x)$ и $\forall x F(x)$ — также формулы.
Замечание. Формулы, построенные таким образом, являются *связанными*.

Аксиомы исчисления предикатов делятся на 2 группы:

1. S_I и S_{II} — аксиомы исчисления высказываний;
2. предикатные аксиомы:
 - (a) $\forall x F(x) \supset F(y)$;
 - (b) $F(y) \supset \exists x F(x)$.

Правила вывода исчисления предикатов содержат:

1. Modus Ponens;
2. Правило обобщения (введения квантора $\forall$): $\frac{F \supset G(x)}{F \supset \forall x G(x)}$
3. Правило введения $\exists$: $\frac{G(x) \supset F}{\exists x G(x) \supset F}$

2.3.2 Интерпретация формул исчисления предикатов

Интерпретация формулы F в исчислении предикатов определяется сопоставлением:

1. Каждой предметной константе некоторого элемента из M .
2. Каждому n -местному функциональному символу отображения $f: M^n \rightarrow M$.
3. Каждому m -местному предикатному символу $P: M^m \rightarrow B$.

Пример. $F: \forall x P(x) = .$
 $M = \{1, 2\}.$

M	$\parallel$	1		2	
$P(x)$	$\parallel$	И(1)		Л(0)	

Формула F ложна в данной интерпретации.

Пример. $F: \forall x (P(x) \supset Q(f(x), a))$. Определить интерпретацию и определить истинность в данной интерпретации.

$M = \{1, 2\}, \quad a = 1, \quad a \in M.$

M	$\parallel$	1		2	
$f(n)$	$\parallel$	2		1	

M	$\parallel$	1		2	
$P(x)$	$\parallel$	Л(0)		И(1)	

(x, y)	$\parallel$	$Q(x, y)$	
(1, 1)	$\parallel$	И(1)	
(1, 2)	$\parallel$	И(1)	
(2, 1)	$\parallel$	Л(0)	
(2, 2)	$\parallel$	И(1)	

2.3.3 Эквивалентные преобразования в исчислении предикатов

Определение. Формулы называются эквивалентными, если при любых интерпретациях они принимают одно значение.

1. $\overline{\exists x P(x)} = \forall x \overline{P(x)}$;
2. $\overline{\forall x P(x)} = \exists x \overline{P(x)}$;

Дистрибутивность операции $\forall$ относительно конъюнкции:

3. $\forall x (P_1(x) \& P_2(x)) = \forall x P_1(x) \& \forall x P_2(x)$;

Дистрибутивность квантора существования относительно дизъюнкции:

4. $\exists x (P_1(x) \vee P_2(x)) = \exists x P_1(x) \vee \exists x P_2(x)$;

Односторонние:

$$5. \exists x(P_1(x) \& P_2(x)) \Rightarrow \exists xP_1(x) \& \exists xP_2(x);$$

$$6. \forall xP_1(x) \vee \forall xP_2(x) \Rightarrow \forall x(P_1(x) \vee P_2(x));$$

Коммутативность одинаковых кванторов:

$$7. \forall x\forall yP(x, y) = \forall y\forall xP(x, y);$$

$$8. \exists x\exists yP(x, y) = \exists y\exists xP(x, y);$$

Пусть Y — некоторая формула, не содержащая переменную x :

$$9. \forall x(P(x) \& Y) = \forall xP(x) \& Y;$$

$$10. \forall x(P(x) \vee Y) = \forall xP(x) \vee Y;$$

$$11. \exists x(P(x) \& Y) = \exists xP(x) \& Y;$$

$$12. \exists x(P(x) \vee Y) = \exists xP(x) \vee Y;$$

Замечание. Поскольку исчисление предикатов есть расширение исчисления высказываний, то для формул исчисления предикатов остаются верными и эквивалентные предикаты, действующие для исчисления высказываний.

2.3.4 Нормальная форма в исчислении предикатов

В исчислении предикатов так же, как и в исчислении высказываний, имеются *нормальные формы* (НФ), в частности, ПНФ — пренексная нормальная форма:

$$(Q_1x_1Q_2x_2\ldots Q_nx_n)C, \quad (2.5)$$

где $Q_1, Q_2, \ldots, Q_n \in \{\exists, \forall\}$, $(Q_1x_1, Q_2x_2, \ldots, Q_nx_n)$ — префикс, C — матрица-формула в КНФ (бескванторное логическое выражение в КНФ).

Любая формула исчисления предикатов может быть преобразована в ПНФ с помощью некоторого алгоритма. Рассмотрим работу этого алгоритма на примере:

Пример. $F: \forall x(P_1(x)) \supset \neg \forall x(P_2(y) \vee \exists yP_3(x, y))$. Задача: построить ПНФ.

1 шаг: поскольку КНФ предполагает использование логических связок $\vee$, $\&$, $\neg$, то нужно избавиться от $\supset$:

$$A \supset B = \overline{A} \vee B;$$

$$\neg \forall x(P_1(x)) \vee \neg \forall x(P_2(y) \vee \exists yP_3(x, y)).$$

2 шаг: изменить (уменьшить) область действия $\neg$:

$$\begin{aligned} \exists x(\overline{P_1(x)}) \vee \exists x(\overline{P_2(y) \vee \exists yP_3(x, y)}) &= \exists x\overline{P_1(x)} \vee \exists x(\overline{P_2(y)} \& \neg \exists yP_3(x, y)) = \\ &= \exists x\overline{P_1(x)} \vee \exists x\overline{P_2(y)} \& \forall y\overline{P_3(x, y)}. \end{aligned}$$

3 шаг: Переобозначим необходимые переменные для того, чтобы не было коллизий (или не было одинаковых):

$$\exists u\overline{P_1(u)} \vee \exists x\overline{P_2(z)} \& \forall y\overline{P_3(x, y)}.$$

4 шаг: Формирование префикса для ПНФ:

$$\begin{aligned}\exists u \overline{P_1}(u) \vee \exists x \forall y (\overline{P_2}(z) \& \overline{P_3}(x, y)) &= \exists u (\overline{P_1}(u) \vee \exists x \forall y (\overline{P_2}(z) \& \overline{P_3}(x, y))) = \\ &= \exists u \exists x (\overline{P_1}(u) \vee \forall y (\overline{P_2}(z) \& \overline{P_3}(x, y))) = \exists u \exists x \forall y (\overline{P_1}(u) \vee \overline{P_2}(z) \& \overline{P_3}(x, y)).\end{aligned}$$

5 шаг: Построение матрицы C (КНФ):

$$\exists u \exists x \forall y ((\overline{P_1}(u) \vee \overline{P_2}(z)) \& (\overline{P_1}(u) \vee \overline{P_3}(x, y))).$$

2.3.5 Доказательство теорем в исчислении предикатов

Доказательство формул исчисления предикатов путём подстановки в них констант называется *методом интерпретации*. Этот метод позволяет интерпретировать формулу как осмысленное утверждение, в связи с чем такой метод называется *семантическим*. Этот метод применим в исчислении предикатов, когда предметная область M конечна. В таком случае M содержит конечное число элементов $M = \{a_1, a_2, \dots, a_n\}$.

В соответствии с семантическим подходом (методом), формула вида $\forall x P(x)$ преобразуется в следующую ($x \in M$):

$$\begin{aligned}\forall x P(x) &= P(a_1) \& P(a_2) \& \dots \& P(a_m) \\ \exists x P(x) &= P(a_1) \vee P(a_2) \vee \dots \vee P(a_m)\end{aligned}$$

Истинность таким образом преобразованных формул может быть проверена путем конечного количества подстановок и вычислений. В случае же, когда предметная область M бесконечна ($|M| = \infty$), семантический метод неприемлем. В этом случае используют процедуры логического вывода с применением аксиом и правил вывода. Такой подход называется *формальным*.

2.4 Метод резолюций и его использование в системах искусственного интеллекта

В этом разделе будут рассмотрены процедуры (алгоритмы) поиска доказательств, которые могут быть использованы при проектировании прикладных систем искусственного интеллекта.

2.4.1 История поиска процедур доказательства теорем

Проблема: разработка общего алгоритма для проверки общезначимости формул исчисления высказываний и исчисления предикатов.

1646–1716 гг. Лейбниц, первые алгоритмы.

XVI век. Пеанó.

1920-е гг. Д. Гильберт.

1930-е гг. А. Чёрч, А. Тьюринг независимо доказали, что не существует общей решающей процедуры (алгоритма), проверяющей общезначимость формул в исчислении предикатов. Тем не менее, существуют алгоритмы поиска доказательств, которые могут подтвердить общезначимость формулы, если она общезначима.

1930 г. Эрбран Жак разработал алгоритм нахождения интерпретации, которая опровергает заданную формулу, однако, если формула общезначима, то такой интерпретации не существует, и алгоритм заканчивает работу за конечное число ходов. Этот подход лёг в основу современных методов.

1960 г. Гилмор — попытка реализации метода Эрбрана на ЭВМ, исходя из тезиса: «формула общезначима тогда и только тогда, когда её отрицание противоречно». Программа Гилмора обнаруживала противоречивость данной формулы, а значит, подтверждала её общезначимость.

1960-е гг. Если отрицание формулы противоречно, то программа обнаруживала этот факт, но была очень громоздкой. Дэвис и Патнем доказали чрезмерную громоздкость и неприменимость в практике программы Гилмора.

1965 г. Робинсон Джон предложил наиболее эффективный алгоритм поиска доказательства, который оказался самым эффективным. Сейчас применяются много оптимизированных для разных областей вариаций этого алгоритма:

- семантическая резолюция;
- лок-резолюция;
- линейная резолюция;
- алгоритм британского музея;
- и др.

2.4.2 Метод резолюций для исчисления высказываний

Теорема. Множество дизъюнктов $S = \{D_1, D_2, \dots, D_n\}$ — это эквивалентная формулировка исходной теоремы:

$$F = (F_1 \& F_2 \& \dots \& F_n \supset \overline{G}),$$

у которой $F_1, F_2, \dots, F_n$ — посылки, G — заключение, а доказательством этой теоремы является опровержение (нахождение опровержения) F .

Доказательство. Доказательство теоремы сводится к доказательству невыполнимости S . Если ввести в рассмотрение пустой дизъюнкт $\square \equiv$, то сказанное выше интерпретируется следующим образом: если среди элементов S имеется хотя бы 1 пустой дизъюнкт, то множество S будет опровергаться, причём в любой интерпретации, следовательно, S невыполнимо. $\square$

Если к невыполнимому множеству S добавить некоторое подмножество S' дизъюнктов таких, что $S': S \supset S'$, то невыполнимость S и S' будет сохранена.

Если среди логических следствий S появится пустой дизъюнкт $S \vee S'$, то это означает, что S невыполнимо, а это в свою очередь значит, что целью доказательства теоремы, сформулированной выше, является получение среди логических следствий множества S пустого дизъюнкта.

Определение (резольвенты). Для любых 2-х дизъюнктов C_1 и C_2 , если существует литера L_1 в C_1 , которая контрарна некоторой букве L_2 в C_2 , то, вычеркнув L_1 и L_2 из C_1 и C_2 соответственно, можно построить дизъюнкцию оставшихся дизъюнктов. Построенный таким образом дизъюнкт называется *резольвентой* C_1 и C_2 .

Пример. Рассмотрим $C_1 = P \vee R$, $C_2 = \bar{P} \vee Q$.

$$\left. \begin{array}{l} C_1 = P \vee R \\ C_2 = \bar{P} \vee Q \end{array} \right\} C = R \vee Q \text{ — резольвента.}$$

Важным свойством резольвенты является то, что она является следствием C_1 и C_2 .

Теорема. Пусть даны 2 дизъюнкта: C_1 и C_2 . Тогда резольвента C дизъюнктов C_1 и C_2 есть логическое следование исходных дизъюнктов, а именно:

$$(C_1 \& C_2) \supset C. \quad (2.6)$$

Определение (правило резолюций или правило вывода логических следствий или правило резолютивного вывода). Пусть S — некоторое множество дизъюнктов. Резолютивный вывод C из множества дизъюнктов есть последовательность $C_1, C_2, \dots, C_n$ дизъюнктов, каждый из которых (C_i) либо принадлежит S , либо является резольвентой дизъюнктов, предшествующих C_i .

Вывод пустого дизъюнкта $\square$ из S называется опровержением, или доказательством невыполнимости S .

Пример. $S = \{\bar{P} \vee Q, \bar{Q}, P\}$

Резольвента $\bar{P} \vee Q$ и $\bar{Q} \Rightarrow \bar{P}$

Резольвента P и $\bar{P} \Rightarrow \square$

Из $S \supset \square$ следует, что S невыполнима.

Таким образом, резольвента строится для 2-х дизъюнктов, содержащих контрарную пару. Если для формулы исчисления высказываний такие пары находятся (описываются) достаточно просто, то для формул исчисления предикатов такая задача решается сложнее. Это связано с тем, что в формулах исчисления предикатов необходимо ещё и совпадение литер контрарной пары.

Т.к. задача резолютивного вывода — поиск опровержения исходной формулы, то для этого можно использовать и константно-частные случаи соответствующих дизъюнктов.

Поиск константно-частных случаев для дизъюнктов в формулах исчисления предикатов, является предметом алгоритма унификации.

Глава 3

Экспертные системы

3.1 Введение в экспертные системы

Цель исследований по экспертным системам (ЭС) состоит в разработке программ, которые при решении задач, трудных для эксперта-человека, получают результаты, не уступающие по качеству и эффективности решениям, получаемым экспертом. Часто экспертные системы характеризуют такую предметную область, для которой применим термин «инженерия знаний», под которым понимается применение принципов и инструментария исследований из области искусственного интеллекта в решение трудных прикладных проблем, требующих знаний эксперта.

Актуальность ЭС:

- 1) технология ЭС существенно расширяет круг практически значимых задач, решаемых на компьютерах, причём решение этих задач приносит значительный экономический эффект;
- 2) технология ЭС является важнейшим средством в решении глобальных проблем традиционного программирования, а именно: длительность и высокая стоимость разработки сложных приложений;
- 3) объединение технологий ЭС с технологией традиционного программирования добавляет новые качества к программам:
 - обеспечение динамичной модификации приложений пользователем;
 - большей «прозрачности» приложений;
 - лучшей графики, интерфейса и пр.

По мнению ведущих специалистов в будущем ЭС найдут следующее применение:

- 1) ЭС будут играть ведущую роль на всех фазах проектирования, разработки, производства, сопровождения, поддержки, распределения и оказания различных услуг.
- 2) технология ЭС, получившая коммерческое распространение, обеспечит революционный прорыв в интеграции приложений из готовых интеллектуальных и информационно взаимодействующих модулей.

ЭС предназначены для решения так называемых «слабоформализуемых» или «неформализованных» задач, т.е. ЭС не отвергают и не заменяют традиционные технологии программирования.

Слабоформализуемые задачи обладают следующими особенностями:

- 1) ошибочностью, неоднозначностью, неполнотой и противоречивостью исходных данных и знаний;
- 2) ошибочностью, неоднозначностью, неполнотой и противоречивостью знаний о предметной области и методах решения задач;
- 3) большой размерностью результата вычислений, т.е. прямой перебор результатов возможных решений очень затруднён;
- 4) динамичностью изменяющихся данных и знаний.

Замечание. Следует подчеркнуть, что слабоформализуемые задачи представляют собой очень большой и важный класс задач. Более того, многие считают, что эти задачи являются наиболее массовым классом задач, требующих решения на ЭВМ.

Решения ЭС обладают свойством прозрачности, т.е. могут быть объяснены пользователю на качественном уровне (объяснительная возможность ЭС). Это качество ЭС обеспечивается их способностью рассуждать о своих знаниях и умозаключениях.

Необходимо отметить, что в настоящее время технология ЭС используется для решения задач следующих типов:

- интерпретация знаний;
- предсказание событий;
- диагностирование;
- планирование;
- конструирование;
- контроль функционирования;
- управление в различных областях:
 - финансы;
 - нефтяная и газовая промышленность;
 - транспорт;
 - авиация и космонавтика;
 - металлургия;
 - и прочих.

Причины, приведшие СИИ (ЭС) к коммерческому успеху:

- 1) интегрированность: разрабатываемые средства легко интегрируются с другими технологиями и средствами СУБД, CASE и др.;

- 2) открытость и переносимость;
- 3) использование традиционных языков программирования;
- 4) использование архитектуры «клиент-сервер», технологии промежуточного программного обеспечения.

3.2 Структура экспертных систем

3.2.1 ЭС статического типа

БД (рабочая память) предназначена для хранения исходных и промежуточных данных решаемой в текущий момент задачи. Данный термин совпадает с термином, используемым в информационно-поисковых системах и СУБД для обозначения всех данных (в первую очередь краткосрочных), хранимых в системе.

База знаний (БЗ) предназначена для хранения долгосрочных данных и знаний, описывающих рассматриваемую ПрО, а не текущих данных (БД), а также правил, описывающих целесообразные преобразования хранимых здесь данных и знаний.

Решатель, используя исходные данные из рабочей памяти и знания из базы знаний, формирует такую последовательность правил, которые, будучи применёнными к исходным данным, приводят к решению задачи.

Система приобретения знаний решает задачи по автоматизированному наполнению ЭС знаниями и предназначена для применения конечным пользователем (не программистом).

Объяснительная система объясняет, каким образом ЭС получила то или иное решение или не получила решения и какие знания система при этом использовала. Применение этой системы облегчает конечному пользователю тестирование системы и повышает доверие к полученному результату.

Диалоговая система ориентирована на организацию дружественного общения с конечным пользователем как на этапе приобретения знаний, так и на этапе получения и объяснения результатов (решений).

Описанная система является статической, поскольку используется в тех приложениях, где можно не учитывать изменения окружающего мира в процессе получения решения.

3.2.2 ЭС динамического типа

В архитектуру динамической ЭС по сравнению со статической вводятся дополнительно 2 компонента:

- 1) подсистема моделирования внешнего мира;

- 2) подсистема сопряжения с внешней средой.

Подсистема сопряжения с внешним миром осуществляет связь с внешним миром через набор датчиков и контроллеров.

Традиционные компоненты статических ЭС в динамических ЭС, прежде всего БЗ и решатель, претерпевают значительные изменения, чтобы отразить временную логику происходящих в реальном мире событий.

3.3 Интеллектуальные информационные технологии и развитие аппарата знаний

До последнего времени состояние исследований в развитии аппарата знаний соответствовало следующей схеме:

Основные направления развития аппарата знаний:

1. Извлечение знаний из различных источников (формализация и интерпретация знаний).
2. Приобретение знаний от профессионалов.
3. Представление знаний:
 - (а) Модели знаний:
 - i. Семантические сети.
 - ii. Фреймы (сети фреймов).
 - iii. Логические системы.
 - iv. Продукции.
 - (б) Системы представления знаний.
 - (с) Базы знаний.
4. Манипулирование знаниями.
5. Объяснение знаний.

3.4 Модели представления знаний (МПЗ)

Типы МПЗ:

- Декларативные:
 - продукционные;
 - редукционные;
 - предикатные.
- Процедурные:

- PLANNER;
- CONNIVER;
- ПРИЗ.
- Специальные (комбинированные):
 - семантические сети;
 - сети фреймов;
 - нечёткие МПЗ;
 - реляционные МПЗ.

3.5 Аппарат знаний и его влияние на интеллектуализацию информационной технологии

В настоящее время наблюдаются следующие основные тенденции названного влияния:

1. *Переход от классических вычислений (архитектура фон Неймана) к альтернативным способам организации вычислительного процесса.*

В течение нескольких последних десятилетий постоянно велись исследования по переходу от архитектуры фон Неймана к иным способам, связанным, в основном, с исследованиями в области искусственного интеллекта и параллельного программирования для многопроцессорных систем. Эти исследования в последнее время реализуются на моделях, строящихся децентрализованными, асинхронными, максимально параллельными, управляемыми по данным в процессе вычисления.

2. *Технология активных объектов.*

Ключевой в перестройке информационных технологий последних лет явилась реализация подхода на основе объектно-ориентированного программирования (ООП), однако этот подход определил пока лишь фундамент будущей технологии, оставляя прежним алгоритмический характер управления процессом вычислений.

Тем временем, направление развития по данным и дальнейшее развитие направлений на основе событий формируют следующие направления интеллектуализации информационных технологий:

- на основе автономных активных объектов, интегрирующих мультиагентную архитектуру;
- методы программирования в ограничениях;
- аппарат недоопределенных моделей.

3. *Приоритет модели, а не алгоритма (см. разд. 3.6.2).*

Известен прогноз, который предполагает, что через 10–15 лет алгоритм ожидает судьба ассемблера и машинных кодов и потеря сегодняшних ключевых позиций, а также места в сравнительно тонком базовом уровне информационных технологий будущего

4. *Параллелизм.*

В новых технологиях параллельность перестаёт быть проблемой, а становится естественным свойством любой программной системы.

3.6 Смена парадигмы разработки интеллектуальных информационных систем

В теории программирования существуют 2 противоположных и взаимодополняющих друг друга понятия:

1. Императивное (алгоритмическое, командное) → понятие алгоритма;
2. Декларативное (непроцедурное, основанное на моделях) → данные.

3.6.1 История развития информационных технологий

Под *моделью* можно понимать определенное множество абстрактных объектов (несколько множеств абстрактных объектов), различающихся условно приписываемым им именам в совокупности с заданной системой отношений между элементами этих множеств.

Под *алгоритмом* можно понимать точное предписание последовательности действий, необходимых для получения искомых результатов

Этапы развития информационных технологий.

Модели		Алгоритмы		
Этап 1:				
		Раздел дан- ных	Операторы обработки	Операторы управления
Этап 2:				
Данные	Модели данных	Операторы обработки	Операторы управления	
Этап 3:				
Данные	Модели данных	Модели знаний	Опера- ционная среда (ре- шатель)	

Таким образом, если на 1-ом этапе развития информационных технологий раздел данных можно рассматривать как неразвитый элемент в большом теле алгоритма, то интеллектуальный решатель на 3-ем этапе представляет собой небольшой автономный элемент алгоритма в теле моделей

3.6.2 Смена влияния моделей и алгоритмов на развитие информационных технологий

В подтверждение сказанному выше, можно отметить следующее: на начальном этапе развития вычислительной техники программирование в кодах считалось единственно возможным способом общения с ЭВМ, однако уже к настоящему времени этот тип программирования сохранил своё место только в тонком, ближайшем к аппаратной части слое операционных систем. По аналогии и модель завоёвывает своё место в практике программирования. Именно модель представляет собой объект исследования и определяет характер формального аппарата, используемого для описания задач.

С моделью работает конечный пользователь — специалист конкретной предметной области. С алгоритмом же работает профессиональный программист и разработчик вычислительных систем. Всё это определило парадоксальность современного состояния прикладных информационных технологий, когда модель можно встретить в основном только в теоретических работах как иллюстрацию к объекту исследования.

Качественное сопоставление модели и алгоритма

Модель	Алгоритм
Принципиально декларативна	Антидекларативна
Симметрична по отношению к своим параметрам, поскольку все они определяются друг через друга	Разделяет параметры на входные и выходные явным образом, определяя вторые через первые
В неявной форме определяет решение всех задач, связанных с объектом исследования (программирования)	Определяет в явной форме и задаёт решение только одной задачи, отношение которой к реальному объекту не всегда очевидно
Может быть недоопределенной	Алгоритм и недоопределенность — несовместимые понятия
В общем случае определяет всё пространство решений	Позволяет получать только отдельные точечные решения (в общем случае).

Можно представить себе такую идеальную ситуацию, когда компьютер способен взаимодействовать с конечным пользователем непосредственно (напрямую). Получая на вход формальную модель или её описание, компьютер автоматически «сжимает» её до минимального k -мерного параллелепипеда в соответствующем гиперпространстве параметров модели. При введении дополнительных ограничений или изменении параметров модели этот параллелепипед в общем случае «сжимается» или изменяет свои размеры (уточняется, дополняется), возможно исчезая совсем, в том случае, если модель или введённые ограничения несовместимы.

Такой идеальный способ организации вычислений на основе моделей в форме сжатия пространства моделей был бы внутренне параллельным, а также недетерминированным и асинхронным, и, следовательно, естественным образом переносимым на параллельные ЭВМ.

Замечание. Очевидно, что для человека, сформировавшегося как программист или как конечный пользователь на основе императивной технологии, описанный подход представляется совершенно фантастическим и невозможным в принципе. Однако, такие технологии уже существуют, в частности, такие технологии обеспечивают пользователя почти всем спектром необходимых возможностей и позволяют работать с такими моделями без посредников, допуская в рамках одной модели сочетания различных формальных аппаратов (алгебры логики, функционального анализа, теории множеств) — принцип полимодельности.

Подытожив сказанное, можно отметить, что на самом деле модель и возможность прямого взаимодействия с нею являлись с самого начала развития искусственного интеллекта ключевым ориентиром и естественным следствием этих исследований была принципиальная потребность выхода за пределы парадигмы алгоритма в самых различных направлениях:

- lisp;
- prolog;
- фреймы;
- производственные системы;
- мультиагентные системы;
- методы удовлетворения ограничений (недоопределенность вычислений).

Объективный прогноз развития информационных технологий говорит, что в ближайшие 10–20 лет алгоритм ожидает судьба ассемблера и программирования в годах, потеря сегодняшних ключевых позиций и места в сравнительно тонком базовом уровне будущей информационной технологии.

Приложение А

Программный комплекс мониторинга состояний и управления сложными техническими объектами в реальном масштабе времени

А.1 Назначение, характеристики и структура информационных систем мониторинга состояния сложных технических объектов

В настоящее время состояние дел в области проектирования и эксплуатации программных комплексов мониторинга состояния сложных технических объектов может быть охарактеризовано рядом положений:

- большие (сверхбольшие) потоки обрабатываемой информации;
- возрастание сложности объектов управления;
- увеличение количества объектов управления;
- большое разнообразие типов измерительной информации, используемой для принятия решений;
- неопределённость и нечётность задач при проведении мониторинга задач сложных технических объектов.

А.2 Характеристика основных элементов информационной технологии, которую реализует программный комплекс

А.2.1 Назначение информационной технологии, реализуемой программным комплексом

Мониторинг состояний (МС) предполагает получение в явном виде обобщённых оценок выполнения программы функционирования рассматриваемого объекта управления, либо степень его работоспособности, места и вида возникшей неисправности, оценок прогнозируемых процессов с заданной точностью, с учётом конкретных целей и условий эксплуатации на различных этапах его функционирования.

Мониторинг может проводиться при интеграции всех имеющихся видов измерительной информации и решать следующий перечень задач:

- *контроль функционирования объекта управления*, выполняющийся при его нахождении как штатных, так и внештатных ситуациях; при реализации технологии автоматизированного управления при мониторинге может быть реализован режим выдачи команд оператора с АРМ;
- *контроль работоспособности объекта управления* и, при возникновении неисправности, её диагностирование с указанием места и вида возникшей ситуации;
- *прогнозирование поведения объекта управления* и предсказание развития как штатных, так и нештатных (аварийных) ситуаций с целью их предупреждения и недопущения.

Основным элементом автоматизации процесса МС является функциональный элемент этой системы, решающий задачу по сбору, обработке и анализу всех видов измерительной информации для произвольного типа и количества пользователей.

Таким функциональным элементом является унифицированный типовой модуль автоматизации, объединяющий в себе систему поддержки интерфейса человек-машина и программное обеспечение операторских станций.

А.2.2 Характеристика унифицированного типового модуля автоматизации (УТМА)

При создании автоматизированных систем управления технологическими процессами любой сложности всегда существовала тяжело решаемая проблема: каким образом реализовать технологию формализации знаний о функционировании объектов управления (ОУ). Неоднократно предпринимались попытки реализовать такие технологии путём использования специализированных языков (блок-схемы, циклограммы функционирования и т.д.), однако все эти попытки были обречены на провал. Причиной послужило то, что очень трудно (почти невозможно) добиться от конструктора ОУ хотя бы словесного описания алгоритма манипулирования данными при МС, и дажи при получении такого описания, переложить его на язык программирования. Основная причина такого положения — принципиально разные языки,

на которых общаются программисты и технологи: программисты используют языки программирования, технологи — технологические языки описания соответствующей предметной области.

Единственный выход из этой ситуации — предоставить конечному пользователю понятное ему средство для описания соответствующей предметной области, объяснив при этом возможность самостоятельно (без участия программиста) описывать технологические процессы функционирования ОУ. Основное достоинство рассматриваемого УТМА состоит в том, что с его помощью создаются («программируются») уникальные модули автоматизации, причём самими технологами, почти без участия программистов. Вся работа технолога состоит в том, что он отлаживает непосредственно сам процесс технологического управления, записанного на понятном пользователю языке и состоящем из примитивов, которыми этот конечный пользователь оперирует. При этом технологу необходимы такие основные примитивы для описания своей предметной области, которые включают типовой набор функций, присущий всем процессам автоматизации для мониторинга состояния соответствующих ОУ:

- экранные формы отображения значений, измеряемых параметров типа стрелочных, полосковых или цифровых индикаторов, а также сигнализирующие панели различной формы и содержания, например, индикатор температуры;
- возможность создания архивов штатных и нештатных ситуаций, событий и поведения динамических процессов во времени (так называемые «тренды»), а также полное и выборочное сохранение измеряемых параметров по времени и по условию;
- упрощенный, адаптированный к предметной области язык для реализации алгоритмов обработки информации, математических и логических выражений (например, язык схем анализа, используемый на лабораторных работах;
- ядро (монитор реального времени), которое обеспечивает детерминизм поведения информационной системы или, иными словами, предсказуемое время отклика на внешнее событие. Существует понятие системы реального времени. В такой системе жёстко задано максимальное время отклика, равное Δt ;
- драйверы управления различного рода внешним оборудованием;
- средства, реализующие сетевые функции;
- средства защиты от несанкционированного доступа;
- многооконный графический интерфейс и другие очевидные функции, такие как импорт изображений, создание собственной библиотеки алгоритмов, динамических объектов и др.;
- средства ведения протокола работы пользователя при проведении сеансов работы с данными;
- средства обеспечения работы исполнительной системы на разнородных программно-аппаратных платформах;

- и др.

При обеспечении функционирования информационной системы этапность работы конечного пользователя определяется следующей схемой:

1. Формирование статического изображения рабочего стола (окна):
 - мнемосхема контролируемого объекта;
 - информационные и управленческие связи на мнемосхеме;
 - органы управления различных типов (кнопки, выключатели и др.);
 - и др.
2. Формирование динамических объектов рабочего стола:
 - стрелочные индикаторы;
 - полосковые индикаторы;
 - цифровые индикаторы;
 - тренды контролируемых процессов;
 - произвольные сигнализирующие табло;
 - и др.
3. Описание взаимодействия между имеющимися этапами на мнемосхемах:
 - динамические объекты взаимодействия;
 - входные/выходные каналы измерительной и обработанной информации;
 - и др.

А.3 Режимы распределенной иерархической обработки данных

В распределенных иерархических системах сбора и обработки данных выделяются несколько уровней:

- уровень непосредственного сбора данных. Основан на использовании датчиков, регуляторов и исполнительных механизмов, на которых программное обеспечение загружается с ПЗУ в ОЗУ, флеш-память и др.
- основной уровень. На нём собирается вся необходимая информация от источников низшего уровня и включает в свой состав не только вычислительные средства, но и человека-оператора.
- уровень оценки состояния ОУ. На этом уровне выполняются операции по модернизированию, прогнозированию, оптимизации процессов, и на него поступает информация, собранная на основном уровне. При этом предполагается использование мощных вычислительных ресурсов. Модуль автоматизации на этом уровне строится на базе экспертных, расчётно-логических или моделирующих систем обработки данных реального времени.

Типовой модуль автоматизации на любом из отмеченных уровней состоит из:

1. Базы данных измерений (БДИ) и диалогового редактора баз данных.
2. Графического редактора статических и динамических изображений на мнемосхемах.
3. Графического редактора символов, который позволяет создавать библиотеки типовых пиктограмм, используемых графическими редакторами статических и динамических изображений.
4. Средств сбора и отображения данных в предыстории по любому параметру из базы данных.
5. Генератора отчётов, который позволяет формировать отчеты по данным реального времени и по предыстории.
6. Средств отображения событий реального времени (циклограммы, мнемосхемы, тренды, свотки событий и тревог, звуковая и речевая сигнализация и др.).
7. Средств обработки значений параметров и вычислений, задаваемых пользователем по алгоритму.

Распределённые системы информационного обеспечения включают следующие компоненты:

- средства поддержки сетевой работы в рамках распределенной системы, как в составе ЛВС различных технологий, так и в составе ГВС;
- средство обмена данными и сообщениями между операторами и рабочими станциями;
- средства пароля и защиты, разграничения доступа по уровню прав оператора;
- система «горячего» резервирования и автоматического восстановления для обеспечения надежности, устойчивости и непрерывности вычислительного процесса.

А.4 Основные технологических элементы информационной технологии обработки данных

Основным технологическим элементом рассматриваемой технологии является программный комплекс, реализующий всю её функциональность. Структура программного комплекса:

- операционная среда (система автоматизированной подготовки исходных данных и знаний);
- измерительная среда (система автоматизированной обработки (анализа) измерительной информации);

- система генерации и обслуживания программного комплекса.

Структура операционной среды включает в себя следующие базовые элементы:

- монитор операционной среды;
- операционную систему;
- сетевые средства;
- средства компонентных вычислений (для платформы Windows — DM-системы, для UNIX — CORBA).

Структура функциональных подсистем:

1. Система баз данных.

2. Специализированные интерактивные подсистемы:

- концептуальное моделирование;
- поведенческое моделирование;
- GUI;
- автоматический синтез корректной метапрограммы.

3. Языковые средства операционной среды:

- подсистема описания измеряемых и вычисляемых параметров;
- подсистема описания групп параметров для их совместной обработки;
- подсистема описания моделей сегментации значений параметров;
- подсистема описания динамических моделей изменения значений параметров;
- подсистема описания диалоговых панелей отображения;
- подсистема запросов к базе данных;
- подсистема контроля функционирования объекта управления;
- подсистема проверки условий и изменения хода вычислений;
- подсистема вызова автоматизированных операций над данными;
- подсистема графического (мультимедийного) отображения;
- подсистема организации диалога с конечным пользователем.

Литература

- [1] Ефимов Е.И. *Решатели интеллектуальных задач*, М., Наука, 1992.
- [2] Кузин Л.Т. *Основы кибернетики*, 1979.
- [3] Нильсон Н. *Принципы искусственного интеллекта*, М., Радиосвязь, 1985.
- [4] Симонс Дж. *ЭВМ пятого поколения: компьютеры 90-х годов*, М., Статистика, 1985.
- [5] Эндрю А. *Искусственный интеллект*, М., Мир, 1985.
- [6] А. Тей, П. Грибомон, Ж. Луи, Д. Снийерс. *Логический подход к искусственному интеллекту: от классической логики к логическому программированию*, М., Мир, 1990.