Записки научных
семинаров ПОМИ
Том 499, 2021 г.

Т. М. Косовская

ИЗОМОРФИЗМ ФОРМУЛ ИСЧИСЛЕНИЯ ПРЕДИКАТОВ В ЗАДАЧАХ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

§1. Введение

Работа посвящена применению понятия изоморфизма элементарных конъюнкций формул исчисления предикатов к решению некоторых задач Искусственного Интеллекта (ИИ) и уменьшению их вычислительной сложности.

Использование языка исчисления предикатов в описании задач ИИ было предложено ещё в 70-ые годы XX века, например, в [1] и продолжает исследоваться до настоящего времени. Раздел "Математическая логика" и его подраздел "Исчисление предикатов" включены в большинство учебных программ курсов по ИИ. Однако при решении практических задач исследователи предпочитают использовать язык исчисления высказываний, который более понятен и имеет полиномиальные от длины записи исходных данных оценки по времени. Такой подход получил название логико-алгебраического подхода к решению задач искусственного интеллекта.

При исследовании сложных составных объектов, в описаниях которых учитываются не только свойства самих исследуемых объектов или их элементов, но и отношения между этими элементами, удобные короткие описания объектов, их свойств и целевых условий могут быть получены на языке исчисления предикатов. Основной проблемой такого применения является NP-трудность рассматриваемых задач [2]. Это согласуется с тем, что при моделировании исходных данных с помощью пропозициональных (булевых) переменных длина записи таких описаний экспоненциально зависит от длины записи исходных данных при моделировании задачи с помощью формул исчисления предикатов [3].

Ключевые слова: формулы исчисления предикатов, изоморфизм предикатных формул, вычислительная сложность алгоритма, задачи Искусственного Интеллекта.

Понятие изоморфизма элементарных конъюнкций атомарных предикатных формул было введено в [4] для решения некоторых задач ИИ, сформулированных на языке исчисления предикатов. Это понятие задаёт отношение эквивалентности между формулами, определяющими одно и то же отношение между элементами исследуемого объекта. Однако по сути оно использовалось и в более ранних работах автора под названием “формулы, совпадающие с точностью до имён переменных и порядка записи литералов”.

Это понятие позволяет сформулировать и решить ряд задач, таких как

– “задание метрики в пространстве элементарных конъюнкций предикатных формул”, что важно, например, при решении задачи кластеризации;

– построение “многоуровневого описания классов в задачах распознавания”, существенно снижающего вычислительную сложность задач при их многократном решении;

– построение “логических баз данных”, учитывающих связи между элементами таких баз, задаваемые изоморфными формулами;

– построение “логических онтологий”;

– формирование “предикатной сети”, которая, в отличие от классической искусственной нейронной сети, может после дообучения менять свою конфигурацию (количество слоёв и количество клеток в слое);

– формирование “нечёткой предикатной сети”, позволяющей распознавать новые, отсутствующие в обучающей выборке объекты с вычисляемой в процессе распознавания “степенью уверенности” в правильности распознавания.

Краткие описания решения этих задач (со ссылками на полное их решение) приведены в настоящей статье.

§2. Основные определения

В этом разделе будут даны основные определения и даны ссылки на алгоритмы, позволяющие сформулировать, дать точную математическую постановку и решать некоторые задачи ИИ.

Определение 1 ([4])**.** Две элементарные конъюнкции $A(x_1, \ldots, x_n)$ и $B(y_1, \ldots, y_n)$ атомарных предикатных формул называются **изоморфными**, если существуют такая элементарная конъюнкция $C(z_1, \ldots, z_n)$

и такие подстановки $\lambda_{A,C} = |^{(x_1,\dots,x_n)}_{(z_{i_1},\dots,z_{i_n})}$ и $\lambda_{B,C} = |^{(y_1,\dots,y_n)}_{(z_{j_1},\dots,z_{j_n})}$ различных переменных из формулы $C(z_1,\dots,z_n)$ вместо различных аргументов формулы $A(x_1,\dots,x_n)$ и формулы $B(y_1,\dots,y_n)$ соответственно, что результаты этих подстановок $A\lambda_{A,C} = A(z_{i_1},\dots,z_{i_n})$ и $B\lambda_{B,C} = B(z_{j_1},\dots,z_{j_n})$ совпадают с формулой $C(z_1,\dots,z_n)$ с точностью до порядка литералов.

При этом подстановки $\lambda_{A,C}$ и $\lambda_{B,C}$ называются унификаторами формул $A(x_1,\dots,x_n)$ и $B(y_1,\dots,y_n)$ с формулой $C(z_1,\dots,z_n)$.

Нетрудно видеть, что отношение изоморфизма предикатных формул является отношением эквивалентности.

Задача проверки двух элементарных конъюнкций предикатных формул на изоморфизм полиномиально эквивалентна [5] “открытой” задаче *Изоморфизм графов*, для решения которой не известен полиномиальный алгоритм и не доказана её NP-полнота. В [4] предложен приближённый полиномиальный по времени алгоритм проверки изоморфности двух элементарных конъюнкций предикатных формул. В случае отрицательного ответа этого алгоритма формулы действительно не изоморфны. По результатам численных экспериментов на 10^6 пар формул положительный ответ в 99,95% случаев получен для изоморфных формул.

Рассмотрим задачу проверки того, что из истинности элементарной конъюнкции $C(\overline{z})$[1] следует истинность $A(\overline{x})$ или некоторой её максимальной подформулы $\widetilde{A}(\overline{y})$ на наборе различных значений из $\overline{z}$, где список переменных $\overline{y}$ является подсписком списка переменных $\overline{x}$.

Пусть a и $\widetilde{a}$ – количества атомарных формул в элементарных конъюнкциях $A(\overline{x})$ и в $\widetilde{A}(\overline{y})$ соответственно, m и $\widetilde{m}$ – количества предметных переменных в $A(\overline{x})$ и $\widetilde{A}(\overline{y})$ соответственно.

Числа q и r вычисляются по формулам $q = \frac{\widetilde{a}}{a}$, $r = \frac{\widetilde{m}}{m}$ и характеризуют степень совпадения формул $A(\overline{x})$ и $\widetilde{A}(\overline{y})$. Параметр q, характеризует, насколько информативен фрагмент, содержащий лишь r-ую часть переменных. В этом случае подформула $\widetilde{A}(\overline{y})$ называется (q, r)-фрагментом формулы $A(\overline{x})$.

Определение 2 ([6])**.** Подформула $\widetilde{A}(\overline{y})$ называется **максимальной подформулой** элементарной конъюнкции $A(\overline{x})$, **следующей из** $C(\overline{z}$,

[1]Обозначение $\overline{z}$ здесь и далее используется для записи списка всех переменных формулы.

если она является её (q,r)-фрагментом с максимальным среди всех (q,r)-фрагментов значением параметра q, для которой справедливо $C(\overline{z} \Rightarrow \exists \overline{y}_{\neq} \widetilde{A}(\overline{y})$ и ни для какой подформулы формулы $A(\overline{x})$, с бо́льшим значением параметра q, это следствие не выполняется.

Задача нахождения максимального (q,r)-фрагмента $\widetilde{A}(\overline{y})$ формулы $A(\overline{x})$, следующей из $C(\overline{z})$, называется задачей проверки неполной выводимости этой формулы. Задачу проверки неполного следования будем записывать $C(\overline{z}) \Rightarrow_P \exists \overline{x}_{\neq} A(\overline{x})$.[2] В [13] приведён один из возможных алгоритмов её решения.

Определение 3 ([6]). Элементарная конъюнкция $C(z_1,\ldots,z_k)$ называется **наибольшей общей** (с точностью до имён переменных)[3] **подформулой двух элементарных конъюнкций** $A(x_1,\ldots,x_n)$ и $B(y_1,\ldots,y_m)$, если она изоморфна некоторым подформулам этих элементарных конъюнкций, но после добавления в неё хоть одного литерала становится не изоморфной ни одной подформуле либо формулы $A(x_1,\ldots,x_n)$, либо формулы $B(y_1,\ldots,y_m)$.

Две формулы

$$A(a,b,c) = p(a,b)p(b,a)q(b,a,c) \text{ и } B(a,b,d) = p(b,a)p(b,d)q(a,b,d)$$

имеют подформулы, изоморфные

$$C(u,v,w) = p(u,v)q(v,u,w),$$

которая с точностью до имён переменных является общей подформулой формул $A(a,b,c)$ и $B(a,b,d)$, так как при замене переменных u,v,w на a,b,c соответственно она превратится в $p(a,b)$ & $q(b,a,c)$ – подформулу формулы $A(a,b,c)$, а при замене переменных u,v,w на b,a,d соответственно она превратится в $p(b,a)$ & $q(a,b,d)$ – подформулу формулы $B(a,b,d)$. При добавлении в неё хоть одного литерала она становится не изоморфной ни одной подформуле либо $A(a,b,c)$, либо $B(a,b,d)$.

В худшем случае оценка числа его шагов составляет

$$O(\underbrace{(n\cdot k)\cdot((n-1)\cdot(k-1))\cdot\ldots\cdot((n-s)\cdot(k-s))}_{s=min\{n,k\}}) = O(n^s \cdot k^s),$$

---

[2]Обозначение $\exists \overline{x}_{\neq}$ здесь и далее используется для записи того, что существует набор различных значений для списка переменных $\overline{x}$.

[3]Ниже слова (с точностью до имён переменных) для краткости будут опускаться.

В лучшем случае (например, если каждый предикат содержит все переменные каждой из формул) оценка числа его шагов составляет

$$O(n^2 \cdot k^2),$$

где n и k количество литералов в этих элементарных конъюнкциях.

§3. Области применения использования понятия изоморфных формул

3.1. Метрика в пространстве элементарных конъюнкций предикатных формул. Одной из задач ИИ является разбиение множества исследуемых объектов на классы (кластеры). С этой целью обычно вводится метрика в пространстве описаний объектов.

При использовании языка исчисления предикатов исследуемый объект ω является составным и представлен как множество своих элементов $\omega = \{\omega_1, \dots, \omega_t\}$. Кроме того, на объектах из множества всех исследуемых объектов Ω задан набор предикатов $p_1, \dots, p_n$, задающий свойства элементов $\omega \in \Omega$ и отношения между ними. Описанием $S(\omega)$ объекта ω является конъюнкция всех истинных на ω литералов.

В [7] предложено вычисление метрики на основе нахождения наибольшей общей подформулы. По своей сути вычисление этой метрики похоже на вычисление мощности симметрической разности двух множеств. Однако аналогом пересечения множеств выступает наибольшая общая (с точностью до имён аргументов) подформула, алгоритм нахождения которой, как правило, экспоненциален.

Пусть $\omega^1 = \{\omega^1_1, \dots, \omega^1_{t_1}\}$ и $\omega^2 = \{\omega^2_1, \dots, \omega^2_{t_2}\}$ – два объекта из Ω с описаниями $S(\omega^1)$ и $S(\omega^2)$ соответственно. Отметим, что $S(\omega^1)$ и $S(\omega^2)$ могут иметь разное количество литералов (обозначим посредством s^1 и s^2) и разное количество аргументов.

Найдём количество литералов в наибольшей общей подформуле элементарных конъюнкций $S(\omega^1)$ и $S(\omega^2)$ и обозначим его посредством $s^{1,2}$.

В [7] доказано, что функция ρ, определяемая формулой

$$\rho(\omega^1, \omega^2) = (s^1 - s^{1,2}) + (s^2 - s^{1,2})$$

задаёт метрику в пространстве Ω.

Недостатком такой метрики является то, что, например, для достаточно непохожих объектов, описания которых содержат 5 и 6 литералов, а их наибольшая общая подформула содержит 1 литерал, расстояние между ними равно 9. В то же время для достаточно похожих объектов, описания которых содержат 105 и 106 литералов, а их наибольшая общая подформула содержит 101 литерал, расстояние между ними также равно 9.

Для решения задачи кластеризации больше подходит нормированная функция d, определяемая формулой

$$d(\omega^1, \omega^2) = \frac{\rho(\omega^1, \omega^2)}{s^1 + s^2}.$$

Функция d не задаёт метрику, т.к. для неё не выполняется неравенство треугольника. Её можно назвать степенью различия. Для приведённого в предыдущем абзаце примера степень различия будет соответственно $\frac{9}{11} \approx 0.82$ и $\frac{9}{211} \approx 0.04$.

3.2. Многоуровневое описание классов в задачах распознавания. При решении задач распознавания предполагается, что множество всех исследуемых объектов разбито на классы $\Omega = \cup_{k=1}^{K} \Omega_k$.

Распознавание сложных структурированных объектов с описаниями на языке исчисления предикатов их решение может быть сведено [2] к проверке логических следований вида

$$S(\omega) \Rightarrow \exists \overline{x}_{k\neq}\ A_k(\overline{x}_k), \tag{1}$$

где $A_k(\overline{x}_k)$ – дизъюнкции элементарных конъюнкций атомарных формул.

Строго говоря, в формуле (1) вместо квантора существования следовало бы писать слова “при каких наборах различных значений $\overline{x}$” $(?\overline{x}_{\neq})$

$$S(\omega) \Rightarrow (?\overline{x}_{k\neq})\ A_k(\overline{x}_k), \tag{1$'$}$$

но как переборный алгоритм, так и алгоритмы построения вывода в исчислении предикатов (например, в секвенциальном исчислении предикатов или доказательство методом резолюций для исчислении предикатов) при доказательстве логическог следования (1) не только отвечают на вопрос *“существует ли ... ?”*, но и предъявляют значения для переменных [8].

Проверка формулы (1) может быть разбита на последовательную проверку

$$S(\omega) \Rightarrow \exists \overline{x}_{\neq}\ A(\overline{x}), \tag{2}$$

где $A(\overline{x})$ – элементарные конъюнкции атомарных формул.

В [8] доказаны NP-полнота и оценки числа шагов алгоритмов, решающих задачу (2). Эти оценки имеют экспоненциальный от длины записи формулы $A_k(\overline{x})$ вид. Для алгоритма полного перебора в показателе оценки находится количество переменных формулы $A_k(\overline{x})$,[4] а для алгоритмов, основанных на построении вывода в исчислении предикатов, в показателе оценки находится количество атомарных формул, входящих в формулу $A_k(\overline{x})$. Там же доказана NP-полнота задачи (1) и, следовательно, NP-трудность задачи (1′).

Для каждой пары формул $A_i(\overline{x}_i)$ и $A_j(\overline{x}_j)$ выделим их наибольшие общие подформулы $Q^1_{i,j}(\overline{x}^1_{i,j})$.

При $l = 1, \dots, L-1$ для каждой пары формул $Q^l_{s_1}(\overline{x}^l_{s_1})$ и $Q^l_{s_2}(\overline{x}^l_{s_2})$ выделим их наибольшие общие подформулы $Q^{l+1}_{s_1,s_2}(\overline{x}^{l+1}_{s_1,s_2})$.

Процесс выделения наибольшей общей подформулы оборвётся, т.к. на каждом шаге выделения длина записи выделенных подформул уменьшается.

Каждую формулу вида $Q^l_s(\overline{x}^l_s)$, у которой нет общих подформул с другими выделенными подформулами, обозначим посредством $P^1_i(\overline{y}^1_i)$ $(i = 1, \dots, n_1))$. Ведём новые предикаты p^1_i, которые будем называть предикатами 1-го уровня, и переменные y^1_i – новые переменные для списков исходных переменных, которые будем называть переменными 1-го уровня. Значения этих предикатов определяются равносильностями вида ${p_i}^1(y^1_i) \Leftrightarrow P^1_i(\overline{y}^1_i)$.

При $l = 1, \dots, L-1$ в каждой формуле вида $Q^{l'}_s(\overline{x}^{l'}_s)$, из которой путём выделения наибольшей общей подформулы получена хоть одна уже переименованнвя формула $P^l_i(\overline{y}^l_i)$, заменим все вхождения $P^l_i(\overline{y}^l_i)$ на ${p_i}^1(y^1)$ и обозначим её посредством $P^{l+1}_i(\overline{y}^{l+1}_i)$ $(i = 1, \dots, n_{l+1}))$. Ведём новые предикаты p^{l+1}_i, которые будем называть предикатами $l+1$-го уровня, и переменные y^{l+1}_i – новые переменные для списков

[4]Эта оценка совпадает с оценкой длины записи при моделировании исходных данных на языке исчисления предикатов с помощью пропозициональных (булевых) переменных [3, 7].

исходных переменных и переменных 1-го, ..., l-го уровней, которые будем называть переменными $l+1$-го уровня. Значения этих предикатов определяются равносильностями вида $p_i^{l+1}(y_i^{l+1}) \Leftrightarrow P_i^{l+1}(\overline{y}_i^{l+1})$.

В результате построения предикатов различных уровней исходное множество описаний классов $\{A_k(\overline{x})\}$ может быть записано с помощью равносильной ей многоуровневой системы описаний классов вида

$$\left\{\begin{array}{cccc} & A_k^L(\overline{x}_k^L) & & \\ p_i^1(y_i^1) & \Leftrightarrow & P_i^1(\overline{y}_i^1), & i=1,\ldots,n_1 \\ & \vdots & & \\ p_i^l(y_i^l) & \Leftrightarrow & P_i^l(\overline{y}_i^l), & i=1,\ldots,n_l \\ & \vdots & & \\ p_i^L(y_i^L) & \Leftrightarrow & P_i^L(\overline{y}_i^L), & i=1,\ldots,n_L \end{array}\right. . \qquad (3)$$

Процесс распознавания с помощью многоуровневой системы описаний классов разбивается на последовательную проверку логических следований

$$S(\omega) \Rightarrow \exists \overline{y}_{i\neq}^1\ P_i^1(\overline{y}_i^1),$$

добавление в $S(\omega)$ постоянных форул $p_i^1(\omega_{i_1}^1), \ldots, p_i^1(\omega_{i_{r_i}}^1)$, где $\omega_{i_r}^1$ – найденные наборы попарно различных констант из ω, удовлетворяющие $P_i^1(\overline{y}_i^1)$, с получением описания объекта $S^1(\omega^1)$ на языке исходных предикатов и предикатов 1-го уровня. А затем проверку при $l=2,\ldots,L$

$$S^{l-1}(\omega^{l-1}) \Rightarrow \exists \overline{y}_{i\neq}^l\ P_i^l(\overline{y}_i^l),$$

формирование $S^l(\omega^l)$ и проверку

$$S^L(\omega^L) \Rightarrow \exists \overline{x}_{k\neq}^L\ A_k^L(\overline{x}_k^L).$$

Следует отметить, что все проверяемые логические следования имеют тот же вид, что и (1), причём их правые части существенно короче, чем $A_k(\overline{x}_k)$. Учитывая то, что вычислительная сложность проверки (1) экспоненциально зависит от длины записи правой части, проверка (1) с использованием многоуровневой системы описаний классов (3) имеет существенно ме́ньшую вычислительную сложность. Точные оценки уменьшения вычислительной сложности при многоуровневом описании имеются в [9].

3.3. Логические базы данных. Логическая база данных представляет собой множество объектов, обладающих некоторыми свойствами и связанных между собой заданными отношениями. Запрос имеет форму (1) или (1′). В [10] приведена NP-полная задача *Конъюнктивный булевский запрос*. Однако её практическое решение существенно отличается от практического решения задачи распознавания, задаваемой (1).

При решении задач распознавания правые части (описания классов $A_k(\overline{x}_k)$) фиксированы, а описания объектов $S(\omega)$ меняются при каждом запросе. При решении задачи *Конъюнктивный булевский запрос* база данных $S(\omega)$ фиксирована, а запросы $A(\overline{x})$ могут меняться каждый раз. Поэтому нет возможности структурировать все возможные запросы $A(\overline{x})$.

В [11] описан алгоритм, основанный на выделении в $S(\omega)$ таких множеств пар литералов, что в каждом множестве их конъюнкция изоморфна $P_i^1(\overline{y}_i^1)$. Вводится предикат 1-го уровня p_i^1 и переменная 1-го уровня y_i^1, определяемые равносильностью ${p_i}^1(y_i^1) \Leftrightarrow P_i^1(\overline{y}_i^1)$. В $S(\omega)$ добавляем постоянные атомарные формулы вида $p_i^1(d_i^{1,j})$, где $d_i^{1,j}$ – обозначение для списка констант из ω, для которых выполняется $P_i^1(y_i^1)$. Получаем 2-уровневую базу данных.

Повторяя эту процедуру, можно построить многоуровневую базу данных. Заметим, что построение 2-уровневой базы данных – NP-трудная задача с огромными исходными данными.

При выполнении запроса $S(\omega) \Rightarrow \exists \overline{x}_{\neq}\ A(\overline{x})$ в формуле $A(\overline{x})$ необходимо выделить подформулы, изоморфные $P_i^l(\overline{y}_i^l)$, и заменить их на ${p_i}^l(y_l^1)$.

3.4. Логические онтологии. Для математика онтология – это ориентированный граф, в вершине с нулевым заходом которого находится исследуемое множество объектов Ω_0. В остальных вершинах находятся его подмножества Ω_i $(i = 1, \ldots, N)$ и для каждого Ω_i имеется наибольшее общее свойство его элементов P_i, которое выполняется для всех его элементов.

Если имеется ориентированное ребро (Ω_i, Ω_j), то

– $(\Omega_j \subset \Omega_i)$,

– для всякого объекта ω верно $P_j(\omega) \Rightarrow P_i(\omega)$.

При создании логической онтологии предполагается, что элементы ω множества Ω_0 являются составными $\omega = \{\omega_1, \ldots, \omega_{t_\omega}\}$ и на их элементах заданы предикаты $p_1, \ldots, p_n$. Каждый объект ω имеет описание

$S(\omega)$ в терминах этих предикатов, являющееся конъюнкцией литералов, истинных на ω. Одна из возможных постановок задачи построения логической онтологии и алгоритм её решения приведены в [12].

Выделим в Ω_0 подмножества объектов с изоморфными описаниями. Наибольшим общим свойством $P_0(\overline{x})$ множества Ω_0 назовём дизъюнкцию неизоморфных между собой элементарных конъюнкций (с переменными в качестве аргументов), каждая из которых изоморфна описанию некоторого объекта.

Попарное выделение наибольшей общей подформулы дизъюнктивных членов из $P_0(\overline{x})$ даёт описания первого уровня онтологии. Соединим ориентированными ребрами Ω_0 и множества, удовлетворяющие полученным описаниям.

Произведём попарное выделение наибольшей общей подформулы из описаний полученного уровня. Если выделена формула, изоморфная уже присутствующей в графе, то соответствующие вершины графа отождествляются.

Предыдущее действие выполняется пока наибольшие общие подформулы имеют более одного литерала.

Построение онтологии существенно зависит от представительности множества Ω_0 и, главным образом, от выбранных предикатов, в терминах которых описываются объекты. Существующие логические онтологии часто описывают одни и те же объекты в терминах разных предикатов. Это вызывает большие сложности при объединении онтологий. Задача объединения логических онтологий может служить объектом дальнейших исследований.

3.5. Предикатные сети. Использование нейронных сетей при решении практических задач ИИ широко распространено в настоящее время. Однако, во-первых, они отражают только глобальные признаки объектов, но не их внутреннюю структуру, во-вторых, процесс их обучения не вполне формализован и зачастую уже дообученная сеть выдаёт результат, весьма далёкий от реального и, наконец, количество слоёв с сети и количества ячеек в слое строго зафиксированы, что не соответствует реальной структуре нейронов в мозге, где одни связи могут разрываться, а другие появляться во время обучения.

В [13] предложено понятие самообучающейся сети с ячейками, реализующими предикатные формулы.

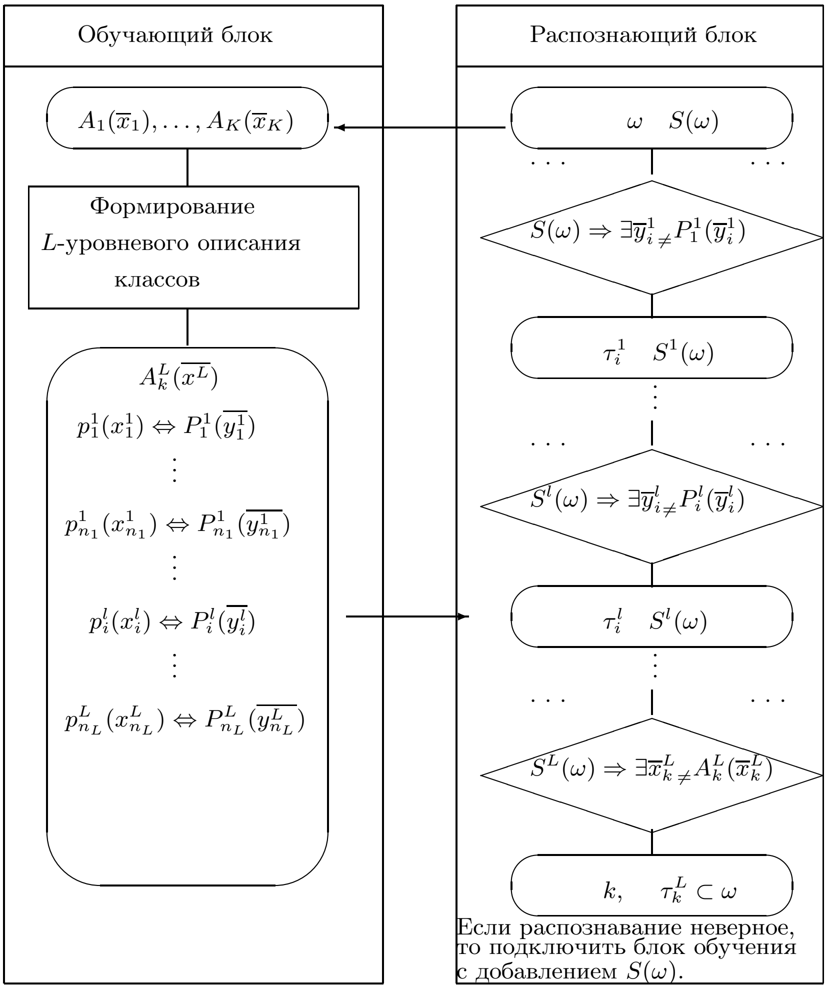

Рис. 1. Схема логико-предикатной сети

В обучающем блоке (долго работающий блок) формируется многоуровневое описание классов по обучающей выборке или по формализованному словесному описанию классов. В распознающем блоке процесс распознавания происходит в соответствии с полученным многоуровневым описанием (быстро работающий блок). Если распознать объект не удалось или распознавание наверное, то многоуровневое описание (а вместе с ним и распознающий блок) перестраивается путём выделения наибольших общих подформул описания нового объекта и уже имеющихся в многоуровневом описании формул.

3.6. Нечёткие предикатные сети. В [14] описано изменение предикатной сети, в которой в каждой ячейке распознающего блока вместо проверки $S^l(\omega) \Rightarrow \exists \overline{y}^l_{i\neq} P^l_i(\overline{y}^l_i)$ проверяется неполное следование $S^l(\omega) \Rightarrow_P \exists \overline{y}^l_{i\neq} P^l_i(\overline{y}^l_i)$ и вычисляется q – максимальная степень совпадения r-той части распознаваемого объекта с описанием одного из объектов, которые были бы распознаны верно. [5]

Следует отметить, что при этом вычислительная сложность проверки в каждой ячейке увеличивается, но так как в правых частях логических следований находятся достаточно короткие формулы, то общее время распознавания увеличится не слишком сильно. Кроме того, можно задаться ограничением на значение параметра q (например, 0,75 или 0,5) и прерывать процесс распознавания, если это значение стало меньше, чем заданное ограничение.

Этот подход основан на выборе "ближайшего соседа" среди тех объектов, на которых сеть уже обучена.

§4. Заключение

Как уже было сказано, проверка двух элементарных конъюнкций предикатных формул на изоморфизм полиномиально эквивалентна "открытой" задаче *Изоморфизм графов* и, следовательно, является GI-полной. Для её решения неизвестен (и, скорее всего, не существует) полиномиальный по времени алгоритм. Однако в большинстве из рассмотренных в работе задач она решается на предварительном этапе. После этого процесс решения основной задачи (распознавание сложного объекта, использование построенных базы данных или онтологии) существенно уменьшается.

Для уменьшения времени работы алгоритма проверки формул на изоморфизм следует внимательно рассматривать структуру проверяемых формул: частота вхождения атомарных формул с одинаковыми

[5]Здесь q и r те же, что определены после Определения 1 в разделе "Основные определения".

предикатными символами, некоторые переменные на порядок чаще встречаются в предикатах, чем другие и т.п. Это требует дополнительного исследования существующего алгоритма или разработку его модификаций для различных распределений таких частот.

При построении логической онтологии автором не рассмотрен случай, когда объекты описываются многозначными признаками, что достаточно часто встречается на практике. Например, "размер", "вес", "цвет" или "химический состав" элемента объекта, "расстояние" или "угол" между его элементами. В этом случае можно ввести бинарные предикаты с дополнительным "количественным" или "качественным" аргументом. Проблема в том, что эти дополнительные аргументы не являются элементами (или переменными для элементов) объектов. Здесь имеется возможность для дальнейших исследований.

Одной из проблем при создании логических онтологий является объединение двух существующих, в каждой из которых объекты описываются своим множеством предикатов, например, $p_1, \ldots, p_n$ и $q_1, \ldots, q_m$. Есть надежда, что выделение "часто встречающихся" изоморфных друг другу подформул в каждой из онтологий позволит установить связь между этими множествами предикатов.

Список литературы

1. Н. Нильсон, *Искусственный интеллект.* Методы поиска решений. М.: Мир, 1973.
2. Т. М. Косовская, *Некоторые задачи искусственного интеллекта, допускающие формализацию на языке исчисления предикатов, и оценки числа шагов их решения.* — Труды СПИИРАН **14** (2010), 58–75.
3. С. Рассел, П. Норвиг, *Искусственный интеллект: современный подход, 2-е изд.* Пер. с англ. М.: Издательский дом "Вильям" (2006), р. 1408.
4. Т. М. Косовская, Д. А. Петров, *Выделение наибольшей общей подформулы предикатных формул для решения ряда задач искусственного интеллекта.* — Вестник СПбГУ. Прикладная математика. Информатика. Процессы управления. **13**, вып. 3 (2017), 250–263.
5. Т. М. Косовская, Н. Н. Косовский, *Полиномиальная эквивалентность задач изоморфизм предикатных формул и изоморфизм графов.* — Вестник Санкт-Петербургского университета. Математика. Механика. Астрономия. **6(64)**. вып. 3 (2019), 430–439.
6. Т. М. Косовская, *Частичная выводимость предикатных формул как средство распознавания объектов с неполной информацией.* — Вестн. С.-Петербург.ун-та, Сер. 10, Вып. 1 (2009), 74 – 84.
7. T. Kosovskaya, *Distance between objects described by predicate formulas. International Book Series. Information Science and Computing.* Book 25.

Mathematics of Distances and Applications (Michel Deza, Michel Petitjean, Krasimir Markov (eds)), ITHEA – Publisher, Sofia, Bulgaria (2012), 153–159.
8. Т. М. Косовская, *Некоторые задачи искусственного интеллекта, допускающие формализацию на языке исчисления предикатов, и оценки числа шагов их решения.* — Труды СПИИРАН. Вып. 14 (2010), 58–75.
9. Т. М. Косовская, *Многоуровневые описания классов для уменьшения числа шагов решения задач распознавания образов, описываемых формулами исчисления предикатов.* — Вестн. С.-Петербург.ун-та. Сер. 10, Вып. 1 (2008), 64–72.
10. М. Гэри, Д. Джонсон, *Вычислительные машины и труднорешаемые задачи.* М. Мир, 1982.
11. Т. М. Косовская, *Построение многоуровневой базы для уменьшения вычислительной сложности решения задачи конъюнктивный булевский запрос.* — 11-я Российская Мультиконференция по Проблемам Управления. Материалы конференции "ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В УПРАВЛЕНИИ" (ИТУ-2018) 2 – 4 Октября 2018 г., Санкт-Петербург, Гнц Рф Ао "КОНЦЕРН «ЦНИИ "ЭЛЕКТРОПРИБОР" (2018), 33–38.
12. Т. М. Косовская, *Выделение изоморфных подформул как средство для создания логической онтологии.* — 12-я Российская Мультиконференция по Проблемам Управления. Материалы конференции "ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В УПРАВЛЕНИИ" (ИТУ-2020) 7 – 8 Октября 2020 г., Санкт-Петербург, Гнц Рф Ао "КОНЦЕРН "ЦНИИ "ЭЛЕКТРОПРИБОР" (2020), ??–??.
13. Т. М. Косовская, *Самообучающаяся сеть с ячейками, реализующими предикатные формулы.* — Труды СПИИРАН No. 6 (43) (2015), 94–113.
14. T. Kosovskaya, *Fuzzy Recognition by Logic-Predicate Network* — Advances in Science, Technology and Engineering Systems J. **5(4)** (2020), 686–699;
15. Т. М. Косовская, *Подход к решению задачи построения многоуровневого описания классов на языке исчисления предикатов.* — Труды СПИИРАН No. 3 (34) (2014), 204–217.

Kosovskaya T. M. Isomorphism of predicate formulas in artificil intelligence problems.

The notion of isomorphism of elementary conjunctions of atomic predicate formulas was introduced by the author for solving some AI problems formulated in the predicate calculus language. Two elementary conjunctions are isomorphic if they coincide up to the names of their arguments and the order of literals. This notion specifies such an equivalence relation between formulas that they determine the same relation between the elements of the investigated object. This notion allows you to formulate and solve a number of problems, such as

– "setting a metric in the space of elementary conjunctions of predicate formulas", which is important, for example, when solving the problem of clustering;

– construction of a “multilevel description of classes in a recognition problem”, which significantly reduces its computational complexity while its multiple solutions;

– construction of “logical databases”, taking into account the relationship between elements of such bases;

– construction of “logic ontologies”;

– formation a “predicate network”, in which, in contrast to the classical artificial neural network, its configuration (the number of layers and the number of cells in a layer) can change after additional training;

– formation of a “fuzzy predicate networ” that allows to recognize new, absent in the training set, objects and to calculate the“degree of confidenc” in the correctness of recognition in the process of its implementation. Brief descriptions of solutions to these problems (with links to complete solutions) are provided in this article.

С.-Петербургский государственный университет
E-mail: kosovtm@gmail.com

Поступило 9 ноября 2020 г.