

УДК 510.62:004.652

## КЛАССИФИКАТОРЫ И СОВМЕСТИМОСТЬ УПРАВЛЕНЧЕСКИХ СПИСКОВ

Гладков Сергей Львович<sup>1</sup>,  
e-mail: gladkovs@list.ru,

<sup>1</sup>Общество с ограниченной ответственностью «Айгео», г. Красноярск, Россия

*В статье проводится исследование совместимости управленческих списков, представляющих собой расширение отношений, введенных Эдгаром Коддом. Рассмотрены признаки совместимости списков, содержащих кодовые атрибуты, значениями которых являются коды классификаторов. В качестве метода исследования семантики классификаторов предложена структурная модель, которая позволяет унифицировать цикл возникновения связи классификаторов с управленческими списками. Раскрывается особенность взаимодействия управленческих списков с классификатором. На основе представленной системы определений раскрываются свойства и классифицируемость управленческих списков. Излагается цикл формирования классификатора и его трансформация в семантический протоклассификатор. На формальном уровне рассматривается операция обобщения двух индивидуальных понятий, ставящая им в соответствие общее понятие. Приводятся леммы и доказательства по выполнению операции обобщения индивидуальных понятий. Рассмотренные в статье формальные требования к классификаторам, совместимости управленческих списков по кодовым атрибутам, а также признаки их совместимости могут лечь в основу разработки стандартов автоматизированного обмена данными между информационными системами.*

**Ключевые слова:** классификатор, протоклассификатор, управленческие списки, кодовые атрибуты, операция обобщения, индивидуальные понятия

## CLASSIFIERS AND COMPATIBILITY OF MANAGERIAL REGISTERS

Gladkov S.L.<sup>1</sup>,

e-mail: gladkovs@list.ru,

<sup>1</sup>Aygeo Limited, Krasnoyarsk, Russia

*The article investigates the compatibility of managerial registers, which represent an extension of the relations introduced by Edgar Codd. The signs of compatibility of lists containing code attributes, the values of which are classifier codes, are considered. A structural model is proposed as a method of studying the semantics of classifiers. This model allows us to unify the cycle of occurrence of the connection of classifiers with managerial registers. The specificity of the interaction of management lists with the classifier is revealed. Based on the presented system of definitions, the properties and classifiability of management lists are disclosed. The cycle of classifier formation and its transformation into a semantic protoclassifier is described. At the formal level, the operation of generalizing two individual concepts is considered, which puts them in accordance with the general concept. Lemmas and proofs on the operation of generalization of individual concepts are given. The formal requirements for classifiers, compatibility of management lists by code attributes, as well as signs of their compatibility, considered in the article, can form the basis for the development of standards for automated data exchange between information systems.*

**Keywords:** classifier, protoclassifier, managerial registers, code attributes, generalization operation, individual concepts

DOI 10.21777/2500-2112-2022-2-49-62

## Введение

В предыдущем исследовании «Формальные свойства совместимости списков» не затронута тема совместимости по атрибутам, содержащим коды классификаторов. Если для каждого из совмещаемых управленческих списков используются «хорошие» классификаторы, то соединение списков по атрибутам с их кодами решается по правилам совместимости атомарных атрибутов [1]. При наличии же противоречий в классификаторах использующие их списки могут оказаться несовместимыми, несмотря на наличие всех прочих признаков совместимости.

Вопросы качества классификаторов для совместимости управленческих списков далее будут рассматриваться в трех аспектах: совместимость классификаторов со списками, совместимость классификаторов между собой, совместимость управленческих списков, содержащих коды классификаторов.

Классификаторы и справочники в информационных системах традиционно представляются в форме таблиц, содержащих списки или иерархии списков. Из этой посылки легко допустить, что классификаторы – это некоторое подмножество управленческих списков, а значит, совместимость классификаторов с управленческими списками есть не что иное, как совместимость управленческих списков [1]. Так ли это? Нет, не так.

Управленческие списки совместимы, когда совмещаемые записи соответствуют одним и тем же понятиям. Совмещение записей управленческого списка с записями классификатора – это совмещение индивидуальных и общих понятий (сущностей и т.д.) [2, с. 59; 3] или, что то же самое, первичных и вторичных сущностей [4, с. 55]. Поэтому, прежде чем переходить к рассмотрению особенностей совместимости классификаторов с управленческими списками, необходимо сделать краткий обзор отношений между индивидуальными и общими понятиями, которые будут использоваться в дальнейшем.

### 1. Операция обобщения двух индивидуальных понятий

В работе [5] введена операция обобщения «+» двух индивидуальных понятий, ставящая им в соответствие общее понятие, что может быть представлено в виде следующего соотношения

$$C \equiv c_1 + c_2,$$

где  $c_i$  – индивидуальные понятия;

$C$  – созданное «на лету» общее понятие мощности 2;

знак «+» - операция «обобщения» или «синтеза» понятий.

Дополнительно напомним, что  $c_1$  и  $c_2$  обладают идентифицирующими характеристиками [1].

Пусть  $\{(F_j(c_1), v_{1j})\}$ ,  $\{(F_j(c_2), v_{2j})\}$  – множества пар из идентифицирующих характеристик их значений понятий  $c_1$  и  $c_2$  соответственно. Тогда все совпадающие характеристики обоих понятий, имеющие к тому же одинаковые значения, делегируются общему понятию  $C \equiv c_1 + c_2$  в качестве его идентифицирующих характеристик. В том случае, когда операнды обобщения не обладают совпадающими характеристиками, их обобщением является универсальное абстрактное понятие  $U \equiv c_1 + c_2$ , не обладающее характеристиками. То есть  $\{F_j(U)\} \equiv \emptyset$ .

Понятия  $c_1$  и  $c_2$ , не обладающие общими характеристиками, будем называть различными и обозначать  $c_1 \neq c_2$ . Если же характеристики этих понятий полностью совпадают, то  $c_1$  и  $c_2$  представляют собой одно и то же понятие  $c_1 \equiv c_2 \equiv c$ .

Общее понятие  $C$  вместе с делегированными ему идентифицирующими характеристиками образует контекст индивидуальных понятий  $c_1$  и  $c_2$ . В результате индивидуальные понятия могут представляться парами  $(C, c_1')$  и  $(C, c_2')$  или  $C * c_1'$  и  $C * c_2'$ . При таком представлении набор идентифицирующих характеристик каждого индивидуального понятия может быть ограничен только его собственными характеристиками. Войшвилло такую форму представления понятия называет описательным именем [6, с. 31].

Представление индивидуальных понятий с использованием их контекста позволяет общему понятию  $C$  сопоставить предикат  $P(C)$ , определяющий, соответствует ли набор значений  $\{v_{1j}\}(\{v_{2j}\})$  для характеристик  $\{F_j(c_1)\}(\{F_j(c_2)\})$  понятию  $c_1(c_2)$ .

С помощью операции «обобщения» на множестве понятий может быть определен частичный порядок. Если  $C \equiv c_1 + c_2$ , то  $C \geq c_1 \wedge C \geq c_2$ , другими словами,  $C$  больше или равно  $c_1$  и  $c_2$ . При этом универсальное абстрактное понятие  $U$  больше любого понятия, т.е. является верхней границей множества понятий, а любое понятие больше или равно самому себе  $C \geq C$ .

Лемма 1. Общее понятие  $C \equiv c_1 + c_2$  не совпадает с  $U$  тогда и только тогда, когда  $c_1$  и  $c_2$  обладают общими характеристиками.

Доказательство очевидно.

Лемма 2. Общее понятие  $C \equiv c_1 + c_2$  – минимальное понятие, обладающее свойством:  $C \geq c_1 \wedge C \geq c_2$ . То есть для любого  $\forall C'$  выполняются условия  $C' \geq c_1 \wedge C' \geq c_2 \Rightarrow C \leq C'$ .

Доказательство.

По определению идентификационные характеристики

$$\{(F_i(C), v_i)\} \equiv \{(F_j(c_1), v_{1j})\} \cap \{(F_j(c_2), v_{2j})\}.$$

В то же время идентификационные характеристики принадлежат пересечению характеристик  $c_1$  и  $c_2$

$$\{(F_i(C'), v_{i'})\} \subseteq \{(F_j(c_1), v_{1j})\} \cap \{(F_j(c_2), v_{2j})\}, \text{ а значит, } C \leq C'.$$

Конец доказательства.

Лемма 3.  $C \geq c$  тогда и только тогда, когда множество идентифицирующих характеристик  $C$  принадлежит множеству идентифицирующих характеристик  $c$ , т.е.  $\{(F_i(C), v_{1i})\} \subseteq \{(F_i(c), v_{1i})\}$ .

Действительно, если среди идентифицирующих характеристик  $C$  присутствует характеристика  $(F_i(C), v_{1i})$ , которая отсутствует среди идентифицирующих характеристик  $c$ , то  $C$  не является общим понятием для  $c$ . Общее понятие  $C$  задаёт отношение эквивалентности на множестве индивидуальных понятий, т.к. все индивидуальные понятия, соответствующие общему понятию, эквивалентны относительно  $C$ , т.е.

$$c \leq C \forall c_i \quad c \sim c_i \mid c_i \leq C.$$

Из определения операции «обобщения»  $C \equiv c_1 + c_2$  следует, что  $c_1 \sim c_2$ . Более того, если  $\exists c_i \mid c_i \sim c_1 \Rightarrow c_i \sim c_2$ .

## 2. Расширение операции обобщения понятий

Рассуждения предыдущего раздела никак не используют тот факт, что операция обобщения «+» определена на индивидуальных понятиях. Поэтому все рассуждения и результаты могут быть применены к общим понятиям, а также объединению индивидуальных и общих понятий.

Теперь пусть даны пара общих понятий, образованных операцией обобщения:

$$C_1 \equiv c_{11} + c_{12} \text{ и } C_2 \equiv c_{21} + c_{22}.$$

Рассмотрим  $C \equiv C_1 + C_2$ , тогда  $C \geq C_1$ , а значит, на основании следствия из леммы 2 их идентифицирующие характеристики обладают свойством:  $\{(F_i(C), v_{1i})\} \subseteq \{(F_i(C_1), v_{1i})\}$ . Другими словами, множество характеристик  $C$  принадлежит множеству характеристик  $C_1$ .

В зависимости от того, в каком соотношении друг с другом находятся общие понятия  $C_1$  и  $C_2$ ,  $C$  может оказаться в одном из следующих состояний:

если  $C_1 \neq C_2$ , то  $C = U$ ;

если  $C_1(C_2) \leq C_2(C_1)$ , то  $C = C_2(C_1)$ , в частности,  $C_1 = C_2 \Rightarrow C = C_1 = C_2$ ;

в противном случае,  $C$  – минимальная верхняя граница для обобщения  $C_1 + C_2$ .

Операция обобщения возвращает минимальное понятие, которому принадлежат оба операнда. Кроме того, эта операция идемпотентна, коммутативна и ассоциативна. Поэтому, множество понятий, на котором определена операция обобщения, образует полурешётку [7, с. 38].

### 3. Цикл возникновения классификаторов

Особенность взаимодействия управленческих списков с классификатором наглядно демонстрирует рисунок 1.



Рисунок 1 – Цикл возникновения связи классификаторов с управленческими списками

Классификатор по своему определению и назначению является списком «классов (родов, видов, множеств) элементов» [2, с. 59]. Поэтому классификатору соответствует множество группируемых элементов, которое в нашем случае имеет форму специального управленческого списка  $L$ . Этот управленческий список должен обладать свойством классифицируемости (группируемости) записей.

Классы (группы), созданные из записей исходного списка, сами могут быть представлены в форме списка  $L'$ . При этом, ключи (идентифицирующие характеристики) образованного списка будут представлять собой пересечение множеств ключевых атрибутов (идентифицирующих характеристики), входящих в класс записей исходного списка. Этот условный список в дальнейшем изложении будет именоваться *протоклассификатором*.

Связь классификатора с управленческим списком указывает принадлежность объекта, описываемого записью списка, к некоторому классу. В этом случае использование протоклассификатора неудобно тем, что он содержит ключевые атрибуты, извлечённые из исходного списка. Во-первых, список таких атрибутов уже присутствует в управленческом списке и может быть затерян среди прочих его атрибутов. Таким образом, принадлежность записи к некоему классу оказывается выраженной лишь неявно. Во-вторых, значение ключевого атрибута протоклассификатора может оказаться не атомарным, например, лингвистическим [8]. Преодолевается этот недостаток протоклассификатора кодированием, т.е. установлением соответствия между специальными кодами и наборами ключевых атрибутов каждой записи. Протоклассификатор становится классификатором путем замены в нем ключевых атрибутов и их значений соответствующими кодами. Главный недостаток такой замены в том, что теряется явная

связь между идентификатором класса и элементами этого класса, которая содержалась в ключевых атрибутах, хотя мысленно такая связь предполагается.

Последним шагом организации связи управленческого списка с классификатором является замена наборов значений ключевых атрибутов класса на соответствующий им код [9]. Так управленческие списки оказываются связанными с классификатором через специальный *атрибут класса*. Далее цикл формирования классификатора рассматривается подробнее.

#### 4. Классифицируемость управленческих списков

Основой любого общего понятия является множество индивидуальных объектов или предметов. В рамках нашего исследования индивидуальные объекты представлены в форме записей управленческих списков. Поэтому естественно проследить образование общих понятий из записей управленческого списка. Рассмотрим ряд определений.

*Определение 1.* Управленческий список  $L$ , соответствующий понятию  $C$ , является классифицируемым, если он содержит, по крайней мере, 2 записи  $l_i$  и  $l_j$  такие, что обобщение соответствующих им понятий  $c_i + c_j = C_{ij}$  меньше, чем  $C$ , т.е.

$$\exists i, j \ c_i(l_i) + c_j(l_j) = C_{ij} \wedge C_{ij} \leq C .$$

Классифицируемость управленческого списка  $L$  будет называться *тривиальной*, если все обобщения понятий, соответствующих произвольно выбранной паре записей, порождают понятие  $C$ , т.е.

$$\forall i, j \ c_i(l_i) + c_j(l_j) = C .$$

Так как  $C$  является родовым понятием для записей для списка  $L$  [1, с. 65], то его тривиальная классифицируемость означает, что среди понятий записей этого списка не удалось выявить видовых отличий (атрибутов), соответствующих общему предикату.

Тривиальную классифицируемость списка  $L$  будем называть *вырожденной*, если  $C$  является универсальным абстрактным понятием  $U$ . Вырожденная классифицируемость списка  $L$  означает, что совокупность всех понятий его записей нельзя отнести к общему роду, а лишь к любому произвольному понятию.

*Определение 2.* Управленческий список  $L$ , соответствующий понятию  $C$ , является полностью классифицируемым, если существует конечное множество общих понятий  $\{C_j \mid 1 \leq j \leq m \wedge m > 1\}$ , таких, что

$$\forall j \ C_j < C \equiv C_j \leq C \wedge C_j \neq C ,$$

$$\forall i \ \exists j \ c_i < C_j ,$$

где  $i$  – индекс записи  $l_i$  в списке  $L$ ;

$c_i$  – понятие, соответствующее этой записи.

Определение полностью классифицируемого списка содержит требование наличия разбиения множества понятий (записей) на классы  $\{C_1, C_2, \dots, C_n\}$ . Из этого следует возможность построения на отношении эквивалентности  $A$  такого, что соотношение  $cA\bar{c}$  выполняется тогда и только тогда, когда  $c + \bar{c} = C_i$ . Здесь  $C_i$  – некоторое общее понятие из разбиения, а  $c$  и  $\bar{c}$  – понятия, соответствующие записям  $l$  и  $\bar{l}$  списка  $L$  [10, с. 51].

В работе Ю.А. Шрейдера [10, с. 56] доказана любопытная теорема, которая утверждает, что каждому общему понятию из разбиения записей полностью классифицируемого списка  $L$  могут быть поставлены в соответствие наборы идентифицирующих характеристик и их значений. То есть каждому общему понятию разбиения может быть поставлено в соответствие идентифицирующее его выражение. Ниже приведена формулировка теоремы, но в нашей нотации.

*Теорема об идентифицирующем выражении.* Если  $L$  – полностью классифицируемый управленческий список, для которого  $C$  – родовое понятие, и существует отношение эквивалентности « $\sim$ » на нем, то из общего числа атрибутов списка, равного  $m$ , для каждой записи  $l \in L$  можно выделить  $n$  атрибутов и их значений:



$$F(C) \rightarrow \langle F_1(C), F_2(C), \dots, F_n(C), F_{n+1}(C), F_{n+2}(C), \dots, F_m(C) \rangle,$$

$$V \rightarrow \langle v_1, v_2, \dots, v_n, v_{n+1}, v_{n+2}, \dots, v_m \rangle,$$

и т.д.

Так что 1) разным записям соответствуют разные кортежи признаков и 2) для того, чтобы было  $l \sim \bar{l}$ , необходимо и достаточно, чтобы первые  $n$  признаков совпадали

$$F_1(C) = v_1 \wedge F_2(C) = v_2 \wedge \dots \wedge F_n(C) = v_n.$$

Из теоремы об идентифицирующем выражении вытекают 2 следствия.

Следствие 1. Пусть  $\bar{C} \equiv C_1 + C_2 + \dots$  является обобщением всех понятий разбиения, тогда:

$$P(\bar{C}) \equiv P_1(F_1(\bar{C})) \wedge P_2(F_2(\bar{C})) \wedge \dots \wedge P_j(F_j(\bar{C})) \wedge \dots \wedge P_n(F_n(\bar{C})),$$

где  $P(\bar{C})$  предикат обобщенного понятия  $\bar{C}$  следует считать  $P_j(F_j(\bar{C}))$  – предикат над одним атрибутом  $F_j(\bar{C})$  (одноместный предикат [6, с. 92]);

$\{F_j(\bar{C})\}$  – набор характеристик, идентифицирующий понятие  $\bar{C}$ .

Следствие 2. Для любого общего понятия из разбиения записей  $\{C_1, C_2, \dots\}$  списка  $L$  существует идентифицирующее его выражение:

$$\forall C_i \exists F_1(C_i) = v_{i1} \wedge F_2(C_i) = v_{i2} \wedge \dots \wedge F_n(C_i) = v_{in}.$$

Благодаря тому, что доказательство теоремы приведено в работе Ю.А. Шрейдера, здесь мы ограничимся рассмотрением метода диагностики классифицируемости (МДК) управленческих списков, позволяющего доказывать или отвергать их полную классифицируемость, а также строить набор общих понятий.

#### Метод диагностики классифицируемости

Пусть  $LR$  – множество необработанных записей списка  $L$ ;

$LA$  – множество не группируемых записей, т.е. записей, для которых не удалось построить общее понятие;

$LC$  – множество общих понятий разбиения, каждый элемент которого содержит тройку: название построенного общего понятия; идентифицирующее выражение; множество записей, соответствующие понятия которых эквивалентны относительно вновь построенного общего понятия.

Присвоить  $LR = \{l_i\}$  – массив всех записей списка  $L$ ;  $LA = \{\}$  – пустой массив;  $LC = \{\}$  – пустой массив.

Повторять, пока  $LR$  не станет пустым.

Присвоить  $l_0 = \text{первый элемент}(LR)$ ; Удалить из  $LR$  ( $l_0$ ).

Найти все  $\{l_i\} | l_i \in LR$ , для которых следующее пересечение  $\{(F_j(c(l_i)), v_j)\} \equiv \{(F_j(c(l_0)), v(l_0)_j)\} \cap \{(F_j(c(l_i)), v(l_i)_j)\}$  не пусто и содержит максимальное число характеристик ( $j$ ).

Если  $\{l_i\}$  – пусто, то Поместить в  $LA(l_0)$ .

Иначе Поместить в  $LC$  ( $\{\text{Имя общего понятия}, \{(F_j(c(l_i)), v_j)\}, l_0 \cup \{l_i\}\}$ ).

Удалить из  $LR$  ( $\{l_i\}$ ).

Конец повторения.

Если  $LA$  содержит все записи  $L$ , то  $L$  обладает тривиальной классифицируемостью.

Если  $LA$  не пуст и  $LC$  не пуст, то  $L$  обладает не полной классифицируемостью.

Если  $LA$  пуст, то  $L$  полностью классифицируем, а  $LC$  содержит общие разбиения из общих понятий. Каждое общее понятие  $LC$  снабжено идентифицирующим его выражением, а также множеством эквивалентных ему записей.

Определение 3. Полностью классифицируемый управленческий список  $L$  будем называть равномерным, если все общие понятия, порожденные МДК, содержат один и тот же набор характеристик  $\{F_j\}$ .

## 5. Семантический протоклассификатор

Разбиение записей  $\{C_1, C_2, \dots\}$  списка  $L$ , полученное с помощью МДК, создает семантическую (смысловую) основу *приведенного списка*  $L'$ , описанного в статье [1]. Напомним, что приведенный список – это список, составленный из итоговых записей исходного списка.

*Определение 4. Семантическим протоклассификатором или протоклассификатором для управленческого списка  $L$  называется любой приведенный список  $L'$ , записи которого основаны на понятиях из разбиения  $\{C_1, C_2, \dots\}$ , созданных МДК. При этом атрибуты ключей для каждой записи  $C_i$  протоклассификатора заимствуются  $\{(F_j(C_i), v_j)\} \equiv (\overline{F(C_i)}, \vec{v})$  из записей, эквивалентных относительно понятия  $C_i$ , и соответствуют максимальному набору идентифицирующих характеристик разбиения. Дополнительным атрибутом каждой записи является название понятия этой записи  $N(C_i)$ .*

Так как идентифицирующее выражение общего понятия, создаваемого МДК, строится на основе максимального числа совпадающих характеристик записей исходного списка, то каждое  $C_i$  из разбиения является минимальным общим понятием для всех эквивалентных относительно него записей  $L$ .

К записям протоклассификатора можно применить МДК, что приведет к созданию нового списка общих понятий. Повторное применение этого процесса к вновь созданным спискам должно завершиться, когда очередной список окажется тривиальным, обобщением каждой пары понятий которого является только родовое понятие  $C$ . Заметим, что управленческий список  $L$  и протоклассификатор основаны на одном и том же родовом понятии  $C$ . В результате такого итерационного процесса может быть создано несколько протоклассификаторов, отличающихся последовательным номером шага итерации или *уровнем*. При этом протоклассификатор, созданный из записей исходного списка  $L$ , будет иметь уровень 1, который в дальнейшем будет называться *нижним* уровнем.

Из множества всех протоклассификаторов разного уровня исходного списка  $L$  образуется дерево с вершиной в  $C$ , высота которого будет совпадать с максимальным уровнем списка с нетривиальной классифицируемостью. Путь от вершины этого дерева до каждого узла позволяет представить понятие  $C_{i(1)}$  в описательной форме [6, с. 31]:

$$C * C_{i(s)} * \dots * C_{i(1)},$$

где  $i$  – порядковый номер понятия на соответствующем уровне протоклассификатора;

[1-s] – диапазон уровней протоклассификатора;

$C_{i(j)}$  - *i-e* понятие на *j-м* уровне протоклассификатора.

Распределенная по уровням дерева форма понятия  $C_{i(1)}$  протоклассификатора позволяет распределить характеризующее его выражение (атрибуты и их значения) по узлам дерева:

$$(F_{1c}(C) = v_{1c} \wedge \dots) \wedge (F_{1C_{i(s)}}(C_{i(s)}) = v_{1C_{i(s)}} \wedge \dots) \wedge \dots \wedge (F_{1C_{i(1)}}(C_{i(1)}) = v_{1C_{i(1)}} \wedge \dots).$$

### 5.1 Кодирование записей протоклассификатора

Протоклассификатор бесполезен для целей совместимости управленческих списков главным образом потому, что заимствует атрибуты и значения из классифицируемого списка. Протоклассификатор следует рассматривать не более как смысловую (семантическую) «тень» классификатора. Другое дело, когда идентифицирующим характеристикам каждой записи протоклассификатора однозначным образом ставится в соответствие буквенно-цифровой код. Такой код становится знаком (меткой, тегом, токеном и т.д.) идентифицирующих характеристик, но в отличие от последних: имеет определенный фиксированный формат; уменьшает число атрибутов ключа совместимости управленческих списков, в большинстве случаев, до одного.

*Определение 5. Кодирование протоклассификатора  $L'$  – построение функции преобразования значений идентифицирующих характеристик его записей к буквенно-цифровому коду  $E(\vec{v}) = k_i \in KD$ ,*

где  $E$  – функция преобразования идентифицирующих характеристик в код,  $k_i$  – код,  $KD$  – множество кодов.

В результате кодирования протоклассификаторов могут быть созданы коды двух категорий: атомарные или структурированные.

Атомарные коды – это коды, созданные без учета зависимостей между понятиями  $C_{i(j)}$  в пути каждой записи протоклассификатора. Атомарными оказываются коды, полученные в результате *порядковой* системы кодирования. Но не только. Атомарными, например, являются коды адресных элементов и зданий ФИАС<sup>1</sup>, т.к. зависимости между различными адресными элементами нельзя извлечь непосредственно из значений кодов, как это показано в таблице 1.

Таблица 1 – Атомарные коды ФИАС зависимых адресных элементов

Код	Название
9b968c73-f4d4-4012-8da8-3dacd4d4c1bd	г Красноярск
31d7a698-9838-4c8a-a1ac-283c4074787a	ул 60 лет Октября
3a185d61-84d1-481f-a018-3014346915e8	д. 105

Синтаксически атомарные коды удобны, т.к. их использование практически сводит согласованность управленческих списков к согласованности по атомарному атрибуту. Но в смысловом (семантическом) смысле атомарные коды могут скрывать риск потери информации, когда предикат протоклассификатора является функцией более чем одной характеристики (атрибута). Лишь в случае, когда предикат протоклассификатора зависит только от одной характеристики (атрибута)  $P(C) \equiv P_1(F_1(C))$ , такой потери не происходит [11, с. 154].

Решение задачи сохранения информации может быть осуществлено преобразованием идентифицирующих характеристик протоклассификатора в серию атомарных кодов:

$$E(\vec{v}) = \{k_i, k_{i(s)}, \dots, k_{i(1)}\} \in KD \times KD_s \times \dots \times KD_1 \text{ или}$$

$$E(\vec{v}) = \vec{k} \in \overline{KD}.$$

Наличие серии кодов приводит к необходимости создания нескольких взаимосвязанных классификаторов (например, ОКВЭД<sup>2</sup>), либо к одному классификатору, имеющему иерархическую структуру (таблица 1). Оба решения не слишком упрощают решение задачи совместимости управленческих списков.

Напрашивается еще одно решение этой задачи – включать в каждую запись формируемого классификатора всю серию кодов в качестве ключевых атрибутов. Главным препятствием такого решения является возможность возникновения плавающего числа ключевых атрибутов. Например, в Боготольском районе Красноярского края две деревни Дмитриевка, значит, адрес этих деревень должен дополнительно содержать указание сельсовета. В адресах подавляющего большинства других населённых пунктов Красноярского края указывать сельсовет либо не нужно, либо не обязательно.

Для того, чтобы «упаковать» серию атомарных кодов в один структурированный код, используются *серийная, позиционная, комбинированная системы* кодирования.

**Примечание.** В дополнение к перечисленным выше системам кодирования часто приводят системы *штрихового кодирования*. Но штриховое кодирование относится не столько к разработке самого кода, сколько к его визуальному представлению, т.е. удобной для считывания различными сканерами форме, и поэтому в данной работе оно не рассматривается.

## 6. Определение классификатора

Следуя предыдущим рассуждениям, классификатором становится протоклассификатор после замены идентификационных характеристик каждой записи одним или несколькими кодами. Но такое

<sup>1</sup> ФИАС – Федеральная информационная адресная система.

<sup>2</sup> ОКВЭД – Общероссийский классификатор видов экономической деятельности.



определение не может быть удовлетворительным. Главным образом потому, что классификатор должен иметь возможность соединяться, т.е. обладать свойством совместимости с управленческими списками. Для этого классификаторы должны быть «родственными» управленческим спискам. С другой стороны, классификатор не обладает идентификационными характеристиками общих понятий, которые содержат его записи. Поэтому, например, в отличие от управленческого списка, классификатор не может быть использован как инструмент верификации (подтверждения) и фальсификации (опровержения) [1, с. 63].

Таким образом, требуется, чтобы синтаксис классификатора и управленческого списка совпадали, а по смыслу классификатор представляет собой знак (метку, тег, токен и т.д.) протоклассификатора и своего базового управленческого списка. Для того, чтобы разрешить указанное противоречие в свойствах управленческого списка и классификатора, введем дополнительное соглашение, характеризующее управленческий список.

Дополнительное соглашение<sup>3</sup>. Каждый управленческий список  $L$  обладает свойствами [1, с. 62]:

- наследование, т.е. создание нового управленческого списка на базе уже существующего;
- полиморфизм – управленческие списки могут иметь разные формы.

Дополнительное соглашение позволяет определить классификатор как подкласс (подвид), наследующий [12, с. 116] синтаксическую, но замещающий [12, с. 82] смысловую форму управленческого списка. Другими словами, речь идет о возможности переопределения прежнего определения без полной замены последнего.

*Определение 6.* Классификатором  $GL$  дальше будем считать подкласс (подвид) управленческого списка, который наследует из определения наличие названия  $N(GL)$ , заголовка, тела, а также родового понятия  $GC$ , но замещает следующие свойства:

- атрибуты классификатора – набор упорядоченных пар, разделенных на 3 группы  $(\overline{KK}, \overline{KD}) \cup (\overline{KN}, \overline{ND}) \cup (\overline{A}, \overline{D})$ , где  $(\overline{KK}, \overline{KD})$  – набор из одной или более пар, каждая из которых содержит имя ключевого атрибута и множество кодов в качестве его допустимых значений;  $(\overline{KN}, \overline{ND})$  – набор атрибутов с названиями;  $(\overline{A}, \overline{D})$  – прочие необязательные атрибуты;
- родовое понятие  $GC$  – является знаком родового понятия  $C$  как протоклассификатора, так и базового управленческого списка, т.е.  $GC = \text{sign}(C)$ ;

– имена ключевых атрибутов  $\overline{KK}$  – являются знаками имен характеристик родового понятия  $C$  как протоклассификатора, так и базового управленческого списка, т.е.  $\overline{KK} = \overline{\text{sign}(F(C))}$ . Здесь под  $\overline{\text{sign}}$  понимается набор знаков для каждого ключевого атрибута;

– домены ключевых атрибутов  $\overline{KD}$  представляют собой набор множеств кодов, допустимых в качестве значений атрибутов. Так,  $KD_j$  – кодовый домен для атрибута с именем  $KK_j$ , а также множество значений функции кодирования  $E(v(F(C))) = \vec{k} \in KD$ .

Теперь можно сделать несколько выводов об отличии классификатора от управленческого списка. Во-первых, родовое понятие классификатора и исходного списка его протоклассификатора не совпадают. Во-вторых, родовое понятие, имена и значения ключевых атрибутов не являются независимыми. Они происходят от родового понятия имен и значений характеристик протоклассификатора исходного управленческого списка.

## 7. Подстановка кодов в управленческий список

Подстановка атрибутов с кодами в исходный управленческий список вместо атрибутов и их значений, которые были экспортированы в протоклассификатор, формально завершает цикл возникно-

<sup>3</sup> Дополнительное соглашение заимствует термины «наследование» и «полиморфизм» из области объектно-ориентированного проектирования (программирования) [12].

вения классификатора, но слишком сужает область применимости классификатора. Для того, чтобы избежать неоправданного сужения значения классификатора, введем понятие подходящего для него управленческого списка.

*Определение 7. Управленческий список  $L$  называется подходящим для классификатора  $GL$ , если выполняются следующие условия:*

1. *Обладает подходящим понятием, т.е. родовым понятием  $C(L)$ , для которого родовое понятие классификатора может выступать в качестве знака или  $GC = (\lambda X.sign(X))C(L)$ .*

2. *Обладает подходящим набором атрибутов, т.е. в составе атрибутов списка можно выделить часть, соответствующую характеристикам родового понятия  $\overline{F(C(L))}$ , знаками которых является вектор ключевых атрибутов классификатора или  $\overline{KK} = (\lambda \vec{X}.sign(\vec{X}))\overline{F(C(L))}$ .*

3. *Обладает подходящим набором значений, т.е. множество значений подходящего набора атрибутов одновременно является множеством определения для функции кодирования классификатора*

$$E\left(\overline{v\left(\overline{F(C(L))}\right)}\right) = \vec{k} \in \overline{KD}.$$

Функции ( $\lambda$ -функции [13, с. 35]), примененные в условиях 1–3, в дальнейшем изложении будут называться *характеристическими функциями* классификатора. Разбор реализации этих функций оставлен на потом, пока же можно предположить, что они могут быть реализованы в форме не слишком сложных по форме предикатов.

*Определение 8. Подходящий управленческий список  $L$  называется согласованным с классификатором  $GL$ , когда:*

- 1) *подходящий набор атрибутов заменён ключевыми атрибутами классификатора;*
- 2) *подходящий набор значений заменён кодами, полученными в результате кодирования.*

## 8. Совместимость классификаторов

Итак, в основе совместимости управленческих списков по кодовым атрибутам, в особенности относящихся к различным информационным системам, лежит совместимость связанных с ними классификаторов.

Лишь на первый взгляд может показаться, что нет здесь никакой проблемы: достаточно соединить классификаторы по общим кодам напрямую или, в крайнем случае, через таблицу соответствия. Но стоит взяться за реализацию такого связывания, как возникает много проблем различного рода, которые порождены различиями в подходах в создании классификаторов<sup>4</sup>.

Причина несовместимости двух интуитивно схожих классификаторов кроется в различии их протоклассификаторов.

### *Совместимость классификаторов по целому коду*

Признак предсовместимости классификаторов. Классификатор  $GM$  предсовместим с  $GL$ , если:

- совпадают родовые понятия их протоклассификаторов  $C(GM) = C(GL)$ ;
- совпадают идентифицирующие характеристики  $\overline{F(C(GM))} = \overline{F(C(GL))}$ , в том числе домены.

*Определение 9. Предсовместимый классификатор  $GM$  строго совместим с  $GL$  тогда и только тогда, когда совпадают функции кодирования обоих классификаторов, т.е. одинаковому набору значений идентифицирующих характеристик протоклассификаторов эти функции ставят в соответствие один и тот же код. Это можно представить как*

$$E_{GM}(\vec{v}_i) = E_{GL}(\vec{v}_i) = k_i \in KD \Leftrightarrow \vec{v}(GM)_i = \vec{v}(GL)_i.$$

<sup>4</sup> Формат статьи не позволяет привести и обсудить здесь примеры таких проблем, но с ними можно ознакомиться на электронном ресурсе по ссылке: [https://regoss.ru/zabytyj-regoss-classifiers/#\\_h002](https://regoss.ru/zabytyj-regoss-classifiers/#_h002).

*Определение 10.* Предсовместимый классификатор  $GM$  совместим или функционально совместим с  $GL$  тогда и только тогда, когда можно построить функцию  $FK(KD_{GL}) \subseteq KD_{GM}$ , преобразующую коды  $GL$  в коды  $GM$ , т.е.

$$\exists FK E_{GM}(\vec{v}_i) = FK(E_{GL}(\vec{v}_i)) = k_i \in KD_{GM} \Leftrightarrow \vec{v}(GM)_i = \vec{v}(LM)_i.$$

Очевидно, что понятие совместимости шире понятия строгой совместимости классификаторов, т.к. строгая совместимость может быть превращена в функциональную при помощи тождественной функции, преобразующей коды в самих себя.

Заметим, что равенство значений кодов классификаторов является условием совместимости только для случая строгой совместимости, в остальных случаях используется функция преобразования кодов, которая обычно задаётся в форме таблицы соответствия.

#### **Совместимость классификаторов по части кода**

Как отмечалось ранее, структурированные коды – это серия кодов разного типа, упакованная в один код. Атрибут со значениями в виде структурированных кодов является специальным видом лингвистического атрибута, рассмотренного в работе [1, с. 67]. Для понимания этого факта достаточно представить коды из серии терминальными символами грамматики, а структурированный код – генерируемым предложением грамматики.

Расширение  $(\overline{KK^{ex}}, \overline{KD^{ex}})$  кодового атрибута  $(KK, KD)$  со структурированными значениями – это набор атрибутов. Домен каждого атрибута расширения состоит из кодов одного типа. Расширение структурированного кода можно представлять как извлечение каждого кода серии и размещение его в соответствующем атрибуте расширения.

Как видно, расширение кодового атрибута – это преобразование в набор из одного и более ключевого атрибута с атомарными кодами. В этом случае классификатор может совмещаться с другим классификатором не только по совпадению значений всех таких атрибутов, но и по совпадению части атрибутов расширения, в том числе, состоящей лишь из одного элемента. Например, с кодами ОКАТМО и ОКАТО функционально совместимы коды регионов России. Так коды ОКТМО, ОКАТО 04601413101, 04201813001 совместимы с кодом 24 (Красноярский край), а 41636154051, 41236554000 – с кодом 47 (Ленинградская область).

Дальнейшее обсуждение совместимости классификаторов по части кода потребует применения операции проекции полного набора атрибутов на его часть, которая подробно рассмотрена в книге Мейера [14, с. 24].

*Определение 11.* Пусть  $\vec{k}$  – вектор значений атрибутов  $\overline{KK}$  записи классификатора, а  $\vec{A} \subseteq \overline{KK}$  – подмножество атрибутов. Тогда  $\pi_A(\vec{k})$  – проекция вектора  $\vec{k}$  на подмножество атрибутов  $\vec{A}$ , а  $\pi_A(\overline{KD})$  – вектор доменов атрибутов  $\vec{A}$ .

Другими словами, операция проекции вектора  $\vec{k}$  создает новый вектор, содержащий только значения атрибутов  $\vec{A}$ .

Признак частичной предсовместимости классификаторов.

Классификатор  $GM$  предсовместим с  $GL$ , если:

– родовое понятие  $C(GM)$  протоклассификатора  $GM'$  содержится в пути порождения понятия  $C(GL)$  протоклассификатора  $GL'$ ;

– пересекаются идентифицирующие характеристики  $\overline{F(C(GM))} \cap \overline{F(C(GL))} \neq \emptyset$ , в том числе домены.

*Определение 12.* Частично предсовместимый классификатор  $GM$  частично совместим с  $GL$  тогда и только тогда, когда можно построить функцию  $FK(KD_{GL}^{ex}) \subseteq \pi_{KK_{GL}^{ex}}(KD_{GM}^{ex})$ , преобразующую коды расширения кодового атрибута  $GL$  в проекцию, приводящую расширения кодового атрибута  $GM$  к набору атрибутов расширения  $GL$ .

*Примечание.* Идея обсуждения и реализации совмещения классификаторов через расширение лингвистического атрибута может быть полезна не только для случая значений со структурированными кодами. Классификатор с атомарными кодами может быть частью дерева классификаторов. Примеры таких классификаторов приведены в разделе «5.1 Кодирование записей протоклассификатора». Расширением кода классификатора, встроенного в дерево, следует считать набор атрибутов, состоящий из кодового атрибута исходного классификатора, а также таких же атрибутов из записей его «предков». Таким образом, может быть реализована частичная совместимость с классификатором в составе дерева (иерархии) классификаторов.

#### **Совместимость списков с кодовыми атрибутами**

Выделим общий кодовый атрибут согласованных списков.

*Определение 13.* Согласованные списки  $L$  и  $M$  обладают общим кодовым атрибутом  $KK(L, M)$  по классификатору, если домены этого атрибута в каждом списке состоят из кодов:

- общего классификатора  $GL$ ;
- полностью или частично совместимых классификаторов  $GL$  и  $GM$ .

Определим признак полной (частичной) совместимости по кодовому атрибуту. Пусть даны списки  $L$ ,  $M$  и соответствующие им классификаторы  $GL$  и  $GM$ . Будем говорить, что список  $M$  совместим со списком  $L$  по кодовому атрибуту, если:

- списки  $L$  и  $M$  обладают общим кодовым атрибутом  $KK(L, M)$ ;
- классификатор  $GM$  полностью (частично) совместим с классификатором  $GL$ .

Включение кодовых атрибутов в состав управленческих списков никак не повлияет на признак совместимости списков, сформулированный в статье «Формальные свойства совместимости списков» [1, с. 62]. Это связано с тем, что общий признак здесь рассматривается как определение некоего абстрактного объекта, а для случая реализации особенностей совмещения по кодовому атрибуту предполагается использование механизма совмещения, реализующего полиморфизм управленческого списка [12, с. 82]. Коротко говоря, совместимость управленческих списков по кодовому атрибуту определяется совместимостью классификаторов, связанных с этими списками.

#### **Особенности предикатов**

В простейшем случае непосредственное соединение записей списков  $L$ ,  $M$  по общему кодовому атрибуту определяется предикатом  $KK(L) = KK(M)$ , если общий атрибут один, или  $\overline{KK(L)} = \overline{KK(M)}$ , если общий атрибут представляет собой набор (вектор) атрибутов.

Но управленческий список может содержать атрибут, значением которого является массив кодов. Например, большинство организаций в своих уставах указывают более одного вида экономической деятельности (ОКВЭД). В этом случае непосредственное соединение записей списков  $L$ ,  $M$  может быть реализовано в соответствии с одним из следующих режимов соединения:

- запретить совмещение двух списков, в которых общий атрибут каждого списка может содержать массив кодов;
- разрешить совмещение таких списков, если все записи общего атрибута одного из них могут содержать только один код;
- разрешить совмещение таких списков, в которых общий атрибут каждого списка может содержать один и тот же массив кодов с точностью до порядка их следования;
- разрешить совмещение таких списков, в которых общий атрибут каждого списка может содержать пересекающиеся массивы кодов.

### **Заключение**

Рассмотренные в статье формальные требования к классификаторам, совместимости управленческих списков по кодовым атрибутам, а также признаки их совместимости могут лечь в основу разработки стандартов автоматизированного обмена данными между информационными системами.

Взаимодействие классификаторов и управленческих списков не исчерпывается вопросами совместимости по кодовому атрибуту.

Следует обратить внимание на то, что родовые понятия списков представляют собой классы, которые могут образовывать протоклассификаторы, а, следовательно, с помощью классификаторов могут быть заданы координатные оси информационного пространства муниципального района, региона, страны. Эта тема требует дальнейшего детального исследования вопросов создания архитектуры и состава словарей информационного пространства (см., например, [15, с. 333–379]).

Отдельного анализа требует совместимость списков, записи которых могут быть соединены с вероятностью меньше 1. То есть это совместимость списков по нечеткому кодовому атрибуту [8].

Интересен также вопрос о случаях преобразования несовместимых списков к совместимому виду за счёт приведения к «общему основанию».

К общим направлениям будущих исследований управленческих списков можно отнести списки с нечеткими значениями атрибутов [8], а также разработки формальных грамматик для различных лингвистических переменных.

### Список литературы

1. *Гладков С.Л.* Формальные свойства совместимости списков // Образовательные ресурсы и технологии. – 2021. – № 3 (36). – С. 60–71. – DOI 10.21777/2500-2112-2021-3-60-71.
2. *Омельченко В.В.* Общая теория классификации. Часть 1. Основы системологии познания действительности. – М.: ИПЦ «Маска», 2008. – 436 с.
3. *Омельченко В.В.* Общая теория классификации. Часть 2. Теоретико-множественные основания / предисл. Д.А. Ловцова. – М.: Либроком, 2010. – 296 с.
4. *Аристотель.* Категории. Сочинения в четырех томах. Т. 2 / ред. З.Н. Микеладзе. – М.: Мысль, 1978. – 687 с. – (АН СССР. Ин-т философии. Филос. наследие).
5. *Гладков С.Л.* О требованиях к интеллектуальной модели данных // Образовательные ресурсы и технологии. – 2015. – № 2 (10). – С. 63–70.
6. *Войшвилло Е.К.* Понятие как форма мышления: логико-гносеологический анализ. – М.: Изд-во МГУ, 1989. – 239 с.
7. *Биркгоф Г.* Теория решеток: пер. с англ. – М.: Наука: Главная редакция физико-математической литературы, 1984. – 566 с.
8. *Заде Л.А.* Понятие лингвистической переменной и его применение к принятию приближенных решений / пер. с англ. Н.И. Ринго; под ред. Н.Н. Моисеева и С.А. Орловского. – М.: Мир, – 1976. – 165 с.
9. *Кодд Е.Ф.* Реляционная модель данных для больших совместно используемых банков данных / пер. с англ. М.Р. Когаловского // Системы управления базами данных. – 1995. – № 1. – С. 145–150.
10. *Шрейдер Ю.А.* Равенство, сходство, порядок: Популярное введение в теорию бинарных отношений. С примерами из математической лингвистики. – М.: Наука, 1971. – 256 с.
11. *Мазур М.* Качественная теория информации. – М.: Мир, 1974. – 238 с.
12. *Буч Гради.* Объектно-ориентированный анализ и проектирование с примерами приложений на C++ / пер. с англ. под ред. И. Романовского и Ф. Андреева. – 2-е изд. – М.: Бином; СПб.: Невский диалект, 1998. – 560 с.
13. *Барендрегт Х.* Лямбда-исчисление. Его синтаксис и семантика: пер. с англ. – М.: Мир, 1985. – 606 с.
14. *Мейер Д.* Теория реляционных баз данных: пер. с англ. – М.: Мир, 1987. – 608 с.
15. *Акаткин Ю.М., Ясиновская Е.Д.* Цифровая трансформация государственного управления. Датацентричность и семантическая интероперабельность / под науч. ред. и предисл. В.А. Конявского. – М.: Ленанд, 2019. – 724 с.

### References

1. *Gladkov S.L.* Formal'nye svojstva sovместимости spiskov // Obrazovatel'nye resursy i tekhnologii. – 2021. – № 3 (36). – S. 60–71. – DOI 10.21777/2500-2112-2021-3-60-71.
2. *Omel'chenko V.V.* Obshchaya teoriya klassifikacii. Chast' 1. Osnovy sistemologii poznaniya dejstvitel'nosti. – М.: IPC «Maska», 2008. – 436 s.



3. *Omel'chenko V.V.* Obshchaya teoriya klassifikacii. Chast' 2. Teoretiko-mnozhestvennye osnovaniya / predisl. D.A. Lovcova. – M.: Librokom, 2010. – 296 s.
4. *Aristotel'*. Kategorii. Sochineniya v chetyrekh tomah. T. 2 / red. 3.N. Mikeladze. – M.: Mysl', 1978. – 687 s. – (AN SSSR. In-t filosofii. Filos. nasledie).
5. *Gladkov S.L.* O trebovaniyah k intellektual'noj modeli dannyh // *Obrazovatel'nye resursy i tekhnologii*. – 2015. – № 2 (10). – S. 63–70.
6. *Vojshvillo E.K.* Ponyatie kak forma myshleniya: logiko-gnoseologicheskij analiz. – M.: Izd-vo MGU, 1989. – 239 s.
7. *Birkhof G.* Teoriya reshetok: per. s angl. – M.: Nauka: Glavnaya redakciya fiziko-matematicheskoy literatury, 1984. – 566 s.
8. *Zade L.A.* Ponyatie lingvisticheskoy peremennoj i ego primenenie k prinyatiyu priblizhennyh reshenij / per. s angl. N.I. Ringo; pod red. N.N. Moiseeva i S.A. Orlovskogo. – M.: Mir, – 1976. – 165 s.
9. *Kodd E.F.* Relyacionnaya model' dannyh dlya bol'shih sovmestno ispol'zuemyh bankov dannyh / per. s angl. M.R. Kogalovskogo // *Sistemy upravleniya bazami dannyh*. – 1995. – № 1. – S. 145–150.
10. *Shrejder Yu.A.* Ravenstvo, skhodstvo, poryadok: Populyarnoe vvedenie v teoriyu binarnyh otnoshenij. S primerami iz matematicheskoy lingvistiki. – M.: Nauka, 1971. – 256 s.
11. *Mazur M.* Kachestvennaya teoriya informacii. – M.: Mir, 1974. – 238 s.
12. *Buch Gradi.* Ob»ektno-orientirovannyj analiz i proektirovanie s primerami prilozhenij na C++ / per. s angl. pod red. I. Romanovskogo i F. Andreeva. – 2-e izd. – M.: Binom; SPb.: Nevskij dialekt, 1998. – 560 s.
13. *Barendregt H.* Lambda-ischislenie. Ego sintaksis i semantika: per. s angl. – M.: Mir, 1985. – 606 s.
14. *Mejer D.* Teoriya relyacionnyh baz dannyh: per. s angl. – M.: Mir, 1987. – 608 s.
15. *Akatkin Yu.M., Yasinovskaya E.D.* Cifrovaya transformaciya gosudarstvennogo upravleniya. Datacentrichnost' i semanticheskaya interoperabel'nost' / pod nauch. red. i predisl. V.A. Konyavskogo. – M.: Lenand, 2019. – 724 s.