УДК 519.711.3

# МЕТОДЫ ИССЛЕДОВАНИЯ И ПРЕОБРАЗОВАНИЯ ИЕРАРХИЙ ТИПОВ НА ОСНОВЕ ЛОГИЧЕСКИХ СТРУКТУР

**С. Д. Махортов**

*Воронежский государственный университет*

Статья посвящена построению математической структуры, адекватно моделирующей архитектуру иерархии типов в объектно-ориентированных системах. В качестве такой структуры используется решетка с заданным на ней бинарным отношением специального класса. Даны определения и приведены теоремы, связанные с логикой ограниченных обобщений, которая свойственна операциям на решетке типов. На основе данной теории предложена модель иерархии типов. Эта модель может быть применена для исследования и автоматической оптимизации такой иерархии. Рассмотрено одно из направлений оптимизации — устранение дублирования кода, что представляет собой важнейшую задачу рефакторинга в объектно-ориентированных системах.

## ВВЕДЕНИЕ

Математическая структура — это множество, на элементах которого аксиоматически заданы некоторые отношения и операторы. Решетки, являющиеся частным видом структур, представляют собой эффективное средство формального представления знаний. В частности, решетка множеств (булеан) может быть образована наборами фактов базы знаний, решетка типов формализует иерархию наследования в объектно-ориентированных системах [1—2].

В работах [3—4] автором был предложен математический аппарат, позволяющий рассматривать задачи формализации логического вывода с точки зрения теории решеток и отношений. Введен специальный класс бинарных отношений на решетках и доказан ряд связанных с ним свойств. Рассматриваемые отношения названы логическими, поскольку обладают всеми свойствами, характерными для отношений монотонного продукционно-логического вывода. Для данного класса отношений изучены следующие основные вопросы: существование, структура, эквивалентные преобразования, редукция, каноническая форма. В [5] задача логического вывода сведена к решению специального класса уравнений на решетках.

Указанная теория может быть применена при решении таких практических задач как автоматическая оптимизация логических программ, эффективный обратный вывод в продукционных системах и ряд других. Теоретической базой этих применений является решетка множеств, а также справедливое для нее основное свойство операций продукционно-логического вывода: если $A \to B$ и $A \to C$, то $A \to B \bigcup C$. Его смысл состоит в возможности свободного перехода от частей к целому. Такую логику можно назвать логикой обобщений.

Другие возможные применения теории логических структур базируются на понятии решетки типов. Семантика частичного порядка и основных операций на такой решетке существенно отличается от случая решетки множеств. J. Sova доказал [6], что решетки типов не изоморфны решеткам множеств. В связи с этим и вышеупомянутая логика обобщений не может быть непосредственно перенесена на данный случай. Таким образом, логические структуры на решетках типов и возможности их применения требуют отдельного рассмотрения.

Именно этому аспекту посвящена настоящая работа. Основной ее результат — построение логической структуры, учитывающей особенности архитектуры решетки типов. Даны определения и приведены теоремы, связанные с логикой ограниченных обобщений, которая свойственна операциям на решетке типов. На основе данной теории в заключительном разделе предложена модель иерархии классов. Эта модель может быть применена для исследования и автоматической оптимизации такой иерархии. Рассмотрено одно из направлений оптимизации — устранение дублирования кода, что представляет собой важнейшую задачу рефакторинга в объектно-ориентированных системах [7].

## 1. ОСНОВНЫЕ ПОНЯТИЯ

Вначале введем обозначения и напомним некоторые определения, связанные с решетками. Необходимые для чтения данной статьи сведения о решетках содержатся в [8].

Решеткой называется полуупорядоченное множество $\mathbb{F}$, в котором наряду с отношением частичного порядка $\leq$ («не больше», «содержится») введены также два двуместных оператора $\wedge$ («пересечение») и $\vee$ («объединение»). Эти операторы таковы, что при любых $a, b \in \mathbb{F}$ справедливо

— $a \wedge b \leq a$, $a \wedge b \leq b$;

— если $c \in \mathbb{F}$ и $c \leq a$, $c \leq b$, то $c \leq a \wedge b$;

— $a \leq a \vee b, b \leq a \vee b$;

— если $c \in \mathbb{F}$ и $a \leq c$, $b \leq c$, то $a \vee b \leq c$.

Решетка $\mathbb{F}$ называется ограниченной, если она содержит верхнюю и нижнюю грани — такие два элемента $O, I$, что $O \leq a \leq I$ для любого $a \in \mathbb{F}$. Решетка $\mathbb{F}$ называется полной, если любое ее подмножество имеет в $\mathbb{F}$ точные верхнюю и нижнюю грани.

Как обычно, для частично упорядоченных множеств (в том числе и решеток) будем различать понятия наименьшего элемента (он меньше всех) и минимального элемента (для него нет меньшего элемента).

## 2. ЛОГИЧЕСКИЕ ОТНОШЕНИЯ НА РЕШЕТКАХ

В этом разделе мы рассматриваем бинарные отношения на некоторой полной решетке $\mathbb{F}$. Заметим, что отношение частичного порядка $\leq$ на $\mathbb{F}$ по определению является рефлексивным и транзитивным. Сформулируем ряд базовых определений.

**Определение 2.1**. Бинарное отношение $R$ на $\mathbb{F}$ назовем дистрибутивным по отношению к операции $\vee$ (в данной работе — просто дистрибутивным), если $(b_1, a), (b_2, a) \in R$ справедливо в том и только том случае, если $(b_1 \vee b_2, a) \in R$. Здесь $a, b_1, b_2 \in \mathbb{F}$ — любые.

**Определение 2.2.** Отношение $R$ на $\mathbb{F}$ называется логическим, если оно содержит $\leq$, транзитивно и дистрибутивно.

Из определения 2.2 следует, что отношение $\leq$ само является логическим.

**Определение 2.3**. Логическим замыканием некоторого отношения $R$, заданного на $\mathbb{F}$, называется наименьшее логическое отношение, содержащее $R$.

Два отношения $R_1$ и $R_2$, определенные на одной решетке $\mathbb{F}$, называются логически эквивалентными (или, в контексте данной работы, просто эквивалентными), если их логические замыкания совпадают. Логической редукцией данного отношения $R$ на $\mathbb{F}$ называется эквивалентное ему минимальное отношение $R_0 \subseteq R$.

В работах [3—4] проведено подробное исследование свойств логических отношений, вводимых определениями 2.1—2.3. В частности, доказаны теоремы о существовании логического замыкания и логической редукции для данного произвольного отношения $R$. Однако, как уже говорилось во введении, логика обобщений, основанная на понятии дистрибутивности в смысле определения 2.1, неприменима в некоторых важных практических задачах. Поэтому далее мы введем понятие нестрогой дистрибутивности, которое позволит расширить область применения логических структур.

**Определение 2.4**. Отношение $R$ на решетке $\mathbb{F}$ назовем нестрого дистрибутивным по отношению к операции $\vee$ (в данной работе — просто нестрого дистрибутивным), если $(b_1, a), (b_2, a) \in R$ имеет место в том и только том случае, когда $(b_1 \vee b_2, a) \in R$. Это условие выполнено для любых $a, b_1, b_2 \in \mathbb{F}$, кроме случая $b_1 \leq a$ либо $b_2 \leq a$.

Данное определение отличается от определения 2.1 областью действия свойства дистрибутивности внутри решетки $\mathbb{F}$ — в определении 2.4 она сужается. Такая дистрибутивность определяется исключительно свойствами отношения $R$ (точнее — его редукции), а влияние отношения частичного порядка $\leq$ при этом исключено.

**Определение 2.5**. Отношение $R$ на $\mathbb{F}$ называется логическим с ограниченным обобщением, если оно содержит $\leq$, транзитивно и нестрого дистрибутивно.

Аналогичным образом вводятся соответствующие определению 2.5 понятия логического замыкания и редукции. Для целей настоящей статьи потребуются также теоремы о существовании, ориентированные на формулировку определения 2.5. Они являются новыми, но мы приведем здесь лишь формулировки этих теорем. Их доказательству автор намерен посвятить отдельную работу.

**Теорема 2.1**. Для произвольного бинарного отношения $R$ на решетке $\mathbb{F}$ существует логическое замыкание в смысле определения 2.5. Оно представляет собой транзитивное замыкание некоторого отношения $\tilde{R} \supseteq R$, построенного по данному отношению $R$ за конечное число шагов.

**Теорема 2.2**. Для произвольного бинарного отношения $R$ на решетке $\mathbb{F}$ существует логическая редукция в смысле определения 2.5. Ее построение основано на транзитивной редукции указанного в теореме 2.1 отношения $\tilde{R}$.

Ответ на вопрос о единственности логической редукции отношения $R$ непосредственно связан с единственностью транзитивной редукции $\tilde{R}$. Согласно [9], для отношения, представленного ациклическим графом, транзитивная редукция единственна. Соответственно единственна и логическая редукция отношения $R$, имеющего в определенном смысле ациклическую структуру. Мы не будем здесь подробно останавливаться на исследовании понятия ациклического логического отношения. Отметим лишь, что модель корректной с точки зрения ООП системы типов устроена именно таким образом.

## 3. ПРИМЕНЕНИЕ К ИССЛЕДОВАНИЮ ИЕРАРХИЙ ТИПОВ

При разработке объектно-ориентированных программных систем часто создаются обширные, многоуровневые иерархии классов. Применение средств автоматизированного проектирования и языка UML помогает спроектировать общие связи между основными классами, но при этом не всегда удается четко отслеживать динамику развития всех многочисленных типов. Атрибуты, реализуемые в некотором классе, вдруг оказываются уже созданными ранее в классе-предке или недоступными в некотором параллельном классе, который также требует их использования. Эти и подобные им проблемы особенно часто имеют место в системах, дорабатываемых в течение длительного времени параллельно с эксплуатацией.

Рефакторингом называется модернизация объектно-ориентированного кода с целью повышения его эффективности, читаемости и облегчения сопровождения [7]. Существуют различные средства автоматизированного рефакторинга (см. [7] и Интернет), особенно много их для языка Java. Наша теория дает новую математическую основу построения подобных средств. Кроме учета обычных требований к качеству кода, она позволяет реализовывать бизнес-правила, специфичные для конкретных задач.

Мы рассмотрим здесь пример одновременного моделирования двух отношений между классами в ООП — обобщения и агрегации [10]. Конечная цель этой модели состоит в устранении дублирования кода, что является одной из наиболее важных задач рефакторинга.

Итак, пусть элементами решетки $\mathbb{F}$ являются типы рассматриваемой иерархии. При этом частичный порядок $\leq$ — это отношение обобщения (наследования). Если класс $a$ является предком класса $b$, то $b \leq a$.

Кроме того, для любых $a, b \in \mathbb{F}$ определены две операции. Объединение $a \vee b$ — это наименьший общий предок $a, b$, пересечение $a \wedge b$ — наибольший общий потомок $a, b$ (вторая операция более актуальна в системах с множественным наследованием). Чтобы определение решетки было корректным, добавим два специальных элемента: $I$ — универсальный тип (общий предок, он присутствует во многих объектно-ориентированных системах) и $O$ — фиктивный потомок всех классов.

Интересные изоморфизмы для решеток типов обсуждаются в [2]. Во-первых, всем примитивным атрибутам, фигурирующим в каких-либо типах, назначаются уникальные простые числа. Каждый тип представляется произведением простых чисел своих атрибутов. Тогда $b$ — наследник $a$, если число, сопоставленное типу $b$, делится нацело на число, связанное с типом $a$. В этом случае $a \vee b$ — наибольший общий делитель $a$ и $b$, $a \wedge b$ — наименьшее общее кратное $a$ и $b$. Универсальному типу $I$ соответствует число 1, наименьшему типу $O$ — произведение чисел всевозможных атрибутов.

Другая интерпретация представляет каждый примитивный атрибут типов в виде битовой строки. Пусть общее количество этих атрибутов во всех классах равно $n$. Каждому атрибуту сопоставим строку длиной $n$ с единицей в позиции, равной его уникальному номеру, и нулями в остальных позициях. Тип идентифицируется побитовым ИЛИ всех его атрибутов. Тогда $b$ является наследником $a$, если содержит единицы в тех же позициях, что и $a$, т. е. $a$ И $b$ $= a$. В данном случае справедливо $a \vee b = a$ И $b$, $a \wedge b = a$ ИЛИ $b$. Универсальный тип $I$ представляется строкой из нулей, наименьший тип $O$ — строкой из всех единиц.

Эти интерпретации рассматриваются в [2] лишь для простых классов, в которых все атрибуты примитивны (не являются объектами). Мы обсуждаем более сложный вариант. На вышеуказанной решетке $\mathbb{F}$ рассмотрим $R$ — бинарное отношение агрегации: если объект класса $a$ в качестве атрибута (поля) содержится в классе $b$, то $(b, a) \in R$.

Рассмотрим теперь логическое замыкание $R^L$ отношения $R$ в смысле определения 2.5. Общую семантику этого отношения можно сформулировать как «обладание набором атрибутов». Так, если $b \le a$, то по принципу наследования класс $b$ обладает набором атрибутов класса $a$. Если $(b,a) \in R$, то и в этом случае $b$ получает возможности класса $a$ посредством своего поля.

Семантика $R^L$ требует дополнительного разъяснения, касающегося основного свойства логического отношения $R^L$ — дистрибутивности. Предположим, что $(b,a),(c,a) \in R^L$. Это означает, что каждый из классов $b$ и $c$ обладает набором атрибутов класса $a$. Предположим, что в обоих случаях этот набор передается не в порядке наследования. Тогда справедливое в силу нестрогой дистрибутивности $R^L$ соотношение $(b \vee c, a) \in R^L$ говорит о том, что элемент $b \vee c$ также обладает атрибутами класса $a$. С точки зрения возможностей проектирования классов это вовсе не обязательно. Если два потомка имеют одинаковые атрибуты, то почему их предок должен обладать этими атрибутами? С другой стороны, согласно принципу рефакторинга в такой ситуации должно быть произведено «поднятие» общих атрибутов, т. е. перемещение в родительский класс. Таким образом, логика отношения $R^L$ предусматривает автоматическое решения важной задачи — устранение дублирования кода.

При этом существенным является наличие условия нестрогой дистрибутивности. Соблюдение полной дистрибутивности в нашей модели могло бы привести к некорректным результатам. Чтобы в этом убедиться, достаточно рассмотреть пример $(b,a),(a,a) \in R^L$. Если бы имела место полная дистрибутивность отношения $R^L$, то выполнялось бы $(b \vee a, a) \in R^L$. Общий предок $a$ и $b$ может обладать всеми атрибутами класса $a$ лишь в случае, когда им является сам $a$. В остальных случаях полученное соотношение семантически некорректно.

Заметим также, что приведенные выше изоморфные числовые представления решетки сохраняют свой смысл применительно к отношению $R^L$. Если $(b,a) \in R$, то числовая характеристика класса $a$ равным образом участвует в формировании характеристики класса $b$, независимо от того, является $b$ наследником $a$ или же содержит его объект в качестве атрибута. Определенные таким образом числовые интерпретации адекватно характеризуют архитектуру отношения $R^L$.

В заключение сформулируем один из результатов, который дает представленная в данном разделе модель иерархии типов. Этот результат опирается на сформулированные в п. 2 теоремы 2.1 и 2.2. Пусть для данного отношения $R^L$ построено указанное в формулировке теоремы 2.1 отношение $\tilde{R}$. Согласно теоремам 2.1 и 2.2, транзитивное замыкание $\tilde{R}$ совпадает с $R^L$, а его транзитивная редукция дает логическую редукцию отношения $R$. Заметим также, что ациклическая структура иерархий типов дает гарантию того, что граф отношения $\tilde{R}$ не содержит циклов. В этом случае согласно [9] транзитивная редукция $\tilde{R}$ существует и единственна. Таким образом, справедливо следующее утверждение.

**Теорема 3.1**. Для любой корректной с точки зрения ООП иерархии типов существует наименьшая эквивалентная система типов, не содержащая дублируемых атрибутов. Алгоритм построения этой системы типов сводится к нахождению транзитивной редукции некоторого ациклического графа.

Имеются и другие возможности для применений рассмотренной модели.

## СПИСОК ЛИТЕРАТУРЫ

1. *Тейз А., Грибомон П. и др.* Логический подход к искусственному интеллекту: от классической логики к логическому программированию. Пер. с франц. — М.: Мир, 1990. — 432 с.

2. *Sowa, J.F.* Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole Publishing Co., Pacific Grove, CA, 1999.

3. *Махортов С.Д.* Логические отношения на решетках. // Вестник ВГУ. Серия физика, математика. — Воронеж, 2003, 2. — С. 203—209.

4. *Махортов С.Д.* О редукции логических отношений на решетках. // Вестник факультета ПММ: Вып. 5. — Воронеж: ВГУ, 2004 — С. 172—179.

5. *Махортов С.Д.* Логические уравнения на решетках. // Вестник ВГУ. Серия физика, математика. — Воронеж, 2004, 2. — С. 170—178.

6. *Sowa, J.F.* Conceptual Structures. Addison-Wesley, Reading, MA, 1984.

7. *Фаулер М.* Рефакторинг: улучшение существующего кода. Пер. с англ. — С-Пб: Символ-Плюс, 2004. — 432с.

8. *Биркгоф Г.* Теория решеток. Пер. с англ. — М.: Наука, 1984. — 568 с.

9. *Aho, A.V., Garey M.R. and Ulman J.D.* The transitive reduction of a directed graph. SIAM J. Computing 1:2, pp. 131—137.

10. *Фаулер М., Скотт К.* UML. Основы. Пер. с англ. — С-Пб: Символ-Плюс, 2002. — 192 с.