

Логика веры

В.И. Шелехов

*Институт Систем Информатики им. А.П. Ершова СО РАН,
Новосибирск*

Логика веры обычно определяется как расширение математической логики дополнительной эпистемической модальностью, характеризующей ту или иную степень уверенности по отношению к интересующим нас логическим утверждениям. Часто логика веры строится на базе темпоральной логики. В настоящей работе логика веры рассматривается по отношению к научным теориям в социологическом аспекте, а не математическом.

Категорический императив академика А.Д. Александрова

Категорический императив академика Александра Даниловича Александрова формулирует важнейшее требование к нравственной позиции ученого. Ученый обязан стремиться к объективности своих исследований: «Рассматривать предмет, явление, отстраняя по возможности все личное, преодолевая свои предрассудки; стараться вникнуть, исследовать и понять, «как оно есть на самом деле», а не так, как кажется с первого взгляда или хочется, чтобы было, считаться с фактами и логикой, а не со своими предубеждениями и мнениями авторитетов» [1, 4].

Вера, как признание чего-либо истинным с силой, превосходящей аргументы фактов и логики (В. Соловьев), противна науке, и она также противна подлинной нравственности, если человек из веры отстраняется от фактов и не принимает логики [1, 4].

Требование объективности исследований, обязательное в мире науки, можно пытаться распространять на общий случай. В ситуации принятия жизненно важного решения, когда человек пытается определить, обладают ли интересующие его объекты (предметы или явления) нужными ему свойствами. Довольно часто утверждение того, что интересующий нас объект обладает нужными свойствами, не подкрепляется знаниями (фактами и логикой). В этом случаях человек действует на основе *веры* в то, что объект обладает нужными свойствами.

Понятие веры

Вера – признание чего-нибудь истинным без опоры на факты и логику, лишь на основании внутренней (субъективной) уверенности. Вера – это надежда на достоверность наших утверждений. Вера что-то вроде интуиции или гипотезы. Вера всегда есть риск.

подавляющее большинство утверждений принимается на веру. У отдельного человека нет возможности исследовать каждое утверждение. Он использует знание, полученное другими людьми, обществом в целом. В данном случае мы говорим о *доверии* человека другим людям, получившим нужное знание.

Вера относится к утверждениям, формируемым за пределами совокупного знания, накопленного человечеством, из области еще непознанного [2]. В человеческой деятельности вера возникает по отношению к аспектам деятельности, для которых знание отсутствует.

Понятие веры отличается от привычного представления о вере, как вере религиозной. Религиозная вера – гипертрофированное выражение веры, признающей религиозные утверждения сильнее аргументов фактов и логики.

Приведенное здесь понятие веры не является принципиально новым. Оно присутствует в работе [5] А.Д. Александра при анализе различных нюансов соотношения науки, веры и религиозной веры.

Палитра атрибутов веры

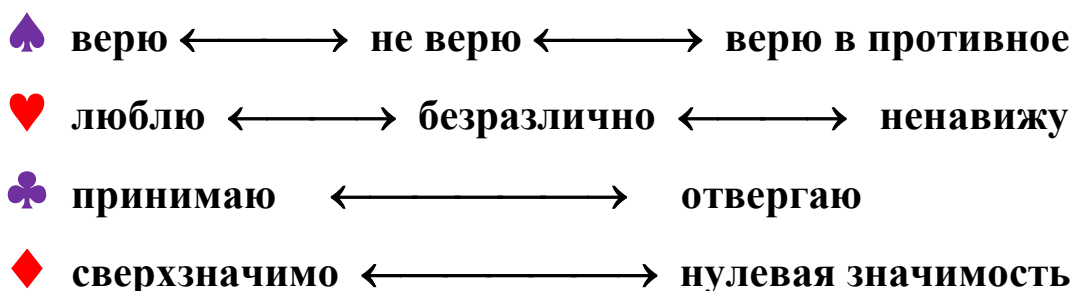
Можно ли измерить веру? Мы различаем сильную веру, слабую веру и безверие. *Сила веры* – главный атрибут, определяющий веру. Возможны другие, сопутствующие атрибуты.

Ситуация, определяемая принимаемым на веру утверждением, может быть оценена как положительная или отрицательная. Атрибут *сила любви* определяет степень положительного или отрицательного отношения к ситуации.

Вера является источником практических действий человека в соответствии с данной верой. Человек принимает веру в своих действиях. Но не всегда. *Степень принятия* (или непринятия) может быть разной.

Предмет веры имеет важное, существенное значение для человека. *Степень значимости* может быть разной. В случае нулевой значимости первые три атрибута теряют всякий смысл.

Итак, набор атрибутов веры следующий: сила веры, сила любви, степень принятия и степень значимости. Минимальные и максимальные значения приведены ниже.



Атрибуты не ортогональны. Между ними есть корреляция. Например, в большинстве случаев сильная вера согласуется с высокой степенью принятия. Тем не менее, возможны самые разные, даже парадоксальные, их сочетания. Например:

- «Я в это почти не верю, но это мне очень нравится, хотя и не много для меня значит»
- «Я в это не верю, и это мне не нравится, однако я это принимаю, потому что так хочет шеф»

Атрибут «сила любви» может отсутствовать в случаях, когда предмет веры не окрашен как положительный или отрицательный.

Инструментов для оценки значений атрибутов веры не существует. Возможна лишь экспертная оценка. Число различных значений каждого из

атрибутов не может быть большим. Не более семи различных дискретных значений. Поскольку экспертная оценка приближительна.

Вера и наука

Научному исследованию, как и любой области человеческой деятельности, сопутствует вера. Необходимой предпосылкой всякого научного исследования является *научный базис*, определяющий совокупное знание и опыт исследователей. Цель исследования – расширить существующий научный базис, найти новые данные, находящиеся за границей текущего знания. Вера в научном исследовании относится к тем аспектам, для которых знания нет. Это может быть связано с методами исследования, вопросами существования объектов с нужными свойствами и другими особенностями.

Областью веры является и сама наука. Исследователь должен верить в перспективность конкретного научного направления, а также что работа в этом направлении ему по силам и даст требуемый результат. При потере веры исследователь перестает работать.

Область веры охватывает весь текущий мировой фронт научных исследований.

Большая часть научного знания принимается ученым на веру. Такое неизбежно при выходе в смежные научные направления. Объем знаний настолько велик, что нет возможности проверить его истинность самостоятельно. Следствие этого – доверие авторитетам.

Утверждения, базирующиеся на вере, часто оказываются ложными. Исследование приходит в состояние *тупика*. Обнаруживается, что желаемый результат недостижим. Распознавание состояния тупика, определение ложности исходных предположений и их корректировка – неотъемлемая часть научного исследования.

Тупик – типичное явление в научной деятельности. Поле научных исследований усеяно огромным множеством тупиков. Ученые подобны кротам, роющим норы в скалистом грунте. И натываются на гранитную стену.

Показательным примером является теория схем программ, инициированная классической работой Ю.И. Янова и несколькими великолепными статьями А.П. Ершова как, например, «О формализации понятия программы». Эти статьи вдохновили многих ученых. Цель теории: доказать разрешимость функциональной эквивалентности для достаточно представительного класса программ и построить полную систему эквивалентных преобразований. Если бы такое удалось, пролился бы золотой дождь¹ разнообразных приложений, в частности, в теории трансляции. Однако класс логико-термальной эквивалентности, доказанной В.И. Иткиным, слаб для приложений. А на более сильных классах неразрешимость функциональной эквивалентности. Тупик.

Это тяжелая трагедия. Результат напряженной работы коллектива ученых в течение более десятка лет оказался напрасным. Работы были прекращены. Р.И. Подловченко была единственная, кто сохранил веру в

¹ Высказывание А.П.Ершова

перспективность теории схем программ. И на удивление многих получала в этой области интересные результаты.

Религия в науке

Казалось бы, научный метод в принципе исключает саму возможность возникновения религии в сфере науки в сообществе ученых и специалистов. Однако.

Тупики в науке бывают разные. Глобальные для науки в целом или локальные в рамках некоторого направления исследований. Высокой или низкой значимости. Короткие по времени или длительные, когда осознание тупика затягивается на несколько десятилетий. Механизм религии иногда присутствует в длительных тупиках, затрудняя выход из них.

Особенности проникновения религии в науку рассмотрим на примере такой известной дисциплины, как структурное программирование. Более сорока лет считается классикой программной инженерии. Преподается во многих университетах. Подавляющее число ученых и специалистов уверены, что применение структурного программирования дает хороший результат. Критический фон присутствовал всегда. Незначительный.

Все началось с нескольких статей Эдварда Дейкстры, основателя концепции структурного программирования. Главная из них: «Заметки по структурному программированию» [6]. Э. Дейкстра ездил в разные страны со своими знаменитыми лекциями. Они были восприняты научной общественностью с большим воодушевлением. Ранее, после чувствительных неудач в крупных космических проектах из-за ошибок в программах, ученые лет десять говорили о кризисе в программировании. Теперь появилась надежда преодолеть кризис. Началось триумфальное шествие структурного программирования по всему миру.

Структурное программирование объявляет оператор `goto` вредным и опасным и предлагает строить программу, используя только правильные операторы: последовательный, условный и цикл `while`. Это возможно, поскольку в соответствии с теоремой Бёма–Якопини любую программу с операторами `goto` можно преобразовать в эквивалентную программу без `goto` введением дополнительных переменных. Дейкстра верил, что внедрение структурного программирования на порядок улучшит производительность труда программистов. Чего не произошло.

Э. Дейкстра настаивал, что изгнание `goto` не является главным в структурном программировании, которое, по его мнению, представляет систему методов. Однако большинство используемых методов появились раньше структурного программирования. По существу новым является лишь стиль доказательного построения программы, в частности, с использованием инвариантов циклов. Э. Дейкстра опубликовал много примеров программ с демонстрацией данного стиля, который, увы, не стал нормой повседневного программирования. Ожидалось, что лет через пять, каждый программист будет обязан снабжать все циклы инвариантами. Однако и в настоящее время подавляющее число программистов не способны писать инварианты циклов.

Не удалось изгнать `goto` из языков программирования. При этом в стандартах языков описание оператора `goto` стыдливо вставляется где-то очень далеко. Н. Вирт в своих языках Паскаль и Модула-2 использовал оператор выхода из цикла. При этом Вирт считал, что действует в духе структурного

программирования. Дейкстра, разумеется, возражал против использования заменителей `goto`. Многочисленные попытки отструктурировать язык Фортран не увенчались успехом. Фортран устоял. Оператора `goto` нет в языке Java. Зато там есть мощный аппарат исключений.

А.П. Ершов возможно был первым, кто сказал, что структурное программирование стало мировой религией. Свидетельство тому – появление «апостолов» – известных авторитетных ученых высшей квалификации, активных сторонников структурного программирования. Каково проявление этой религии? Появление оператора `goto` в программе до сих пор считается бесспорным свидетельством низкой квалификации и дурного вкуса. И что таких надо гнать. Изредка случается стихийный бунт одиночек, не признающих авторитетов и считающих использование `goto` правильным. Их увольняли, чтобы не портили стадо.

Итак, опасен ли оператор `goto`? Является ли программирование без `goto` чем-либо лучше?

Оператор `goto` был действительно опасен при программировании в цифровых кодах команд ЭВМ до появления первых ассемблеров. При неверном адресе команды поведение программы непредсказуемо. В ассемблерных программах такой проблемы уже нет.

В действительности, оператор присваивания намного сложнее и опаснее оператора `goto`. Можно избавиться и от операторов присваивания. Использовать функциональное программирование, которое, однако, неспособно полноценно заменить традиционное императивное программирование.

Оказывается, структурное программирование наносит вред в автоматном программировании [7]. Автоматная программа кодируется непосредственно с использованием меток и операторов перехода, что является естественным и правильным решением. Но противоречит структурному программированию. Чтобы избежать обвинения в некомпетентности, разработчики языков автоматного программирования вынуждены прибегать к разным трюкам. Например, А. Шальто использует switch-технологии [8]. Показано [9, разд. 4, рис.2], что программа при этом становится сложнее. Используется также механизм DSL-расширения² базового языка с представлением автоматной программы в псевдо табличном виде.

Литература

- [1] Александров А. Д. Наука и этика // Проблемы науки и позиция ученого. – Л.: Наука, 1988. – С 9-26.
http://www.belousenko.com/books/publicism/alexandrov_nauka_i_etika.htm
- [2] Вера (философия) // dianomica вики
[http://ru.dianomica.wikia.com/wiki/Вера_\(философия\)](http://ru.dianomica.wikia.com/wiki/Вера_(философия))
- [3] Вера // ЭСБЕ/Вера <https://ru.wikisource.org/wiki/ЭСБЕ/Вера>
- [4] Александров А. Д. Наука и этика // Избранные труды. – Новосибирск: Наука, 2008. – Том 3, С 241-256.

² DSL – Domain Specific Language

- [5] Александров А. Д. Научный поиск и религиозная вера. Раздел «Наука, нравственность и религиозная вера» // Избранные труды. – Новосибирск: Наука, 2008. – Том 3, С 566-570.
- [6] Э. Дейкстра. Заметки по структурному программированию // Дал У., Дейкстра Э., Хоор К. Структурное программирование / М.: "Мир", 1975. - С 7 - 97.
- [7] Тумуров Э.Г., Шелехов В.И. Технология автоматного программирования на примере программы управления лифтом. — ИСИ СО РАН, Новосибирск, 2016. – 18с.
<http://persons.iis.nsk.su/files/persons/pages/lift1.pdf>
- [8] Шалыто А. А. SWITCH-технология. Алгоритмизация и программирование задач логического управления. СПб: Наука, 1998.
<http://is.ifmo.ru/books/switch/1>
- [9] Шелехов В.И. Язык и технология автоматного программирования // «Программная инженерия», №4, 2014. – С. 3-15.
<http://persons.iis.nsk.su/files/persons/pages/automatProg.pdf>