МЕТОДОЛОГИЧЕСКИЕ АСПЕКТЫ
КОГНИТИВНЫХ ПРОЦЕССОВ
(Вычислительные системы)
2002 год Выпуск 170

УДК 519

ОБ ОДНОЙ КЛАССИФИКАЦИИ ЗАДАЧ С ФИЛОСОФСКОЙ ТОЧКИ ЗРЕНИЯ[1]

В.К.Самохвалова, К.Ф.Самохвалов

В жизни нас посещают или мы сами генерируем беспокойства, от которых мы хотели бы избавиться. Эти беспокойства бывают к тому же понятные нам или непонятные. В первом случае мы обычно говорим, что перед нами *задачи*, во втором — *томления*.

Далее, несмотря на то, что томления — это непонятные нам беспокойства, мы, тем не менее, умеем их отличить друг от друга. Тем более это относится к задачам. Кроме того, мы умеем томления отличать от задач.

Наконец, бывает в жизни и так, что некоторые конкретные томление мы собираемся понять и тем самым превратить его в задачу. В таком случае это исходное томление мы называем *непонятной задачей*.

Приведённую бытовую типизацию беспокойства можно утончить за счёт введения некоторых дополнительных разграничений среди тех беспокойств, которые являются понятными. Осуществить, опираясь на общий анализ критериев понятности произвольного беспокойства, такое утончение — это предложить некоторую классификацию задач с философской точки зрения.

Сделать это и тем самым несколько упорядочить интуиции, связанные с употреблением термина "задача", — цель заметки.

[1]Работа выполнена при поддержке гранта РФФИ № 00-06-80180.

1. Задачи и решения

Итак, чем же определяется понятность для человека той или иной беспокоящей его задачи? Самый общий ответ таков: данный человек *понимает* беспокоящуюся его задачу (а не просто томится ею) тогда и только тогда, когда он уверен, что любую вещь, буде она предъявлена его вниманию, он в состоянии распознать как утоление (решение) или как неутоление (нерешение) упомянутого беспокойства (задачи).

В самом деле, нелепо, например, говорить, что некто *понял* задачу, *решил* её, но *не* знает(продолжает беспокоится), решил ли он её. Либо он решил *чужую*, понятную нам, но не ему, задачу, либо он не дорешал свою — ему осталось еще *убедиться* в том, что найдена именно та вещь, которая *его* удовлетворяет. Точнее говоря, ему осталось ещё преобразовать найденную вещь в такую новую вещь, которая действительно *утолит* его первоначальное беспокойство. Вот эта вторая вещь и будет тогда подлинным *решением* его собственной задачи. С другой стороны, столь же нелепо говорить, что он *понял* задачу, но на некотором этапе, *пока он её не решил*, он *не* знает о (не уверен в) том, что он её не решил. Это так же нелепо, как всерьёз (не риторически) спрашивать себя: беспокоюсь ли я, когда я беспокоюсь?

Отсюда вывод: *понятная* человеку задача не может иметь неубедительных для него решений. Неубедительные решения — вообще не решения. Более того, они — убедительные *нерешения*.

Разумеется, нигде не предполагается, что всякая понятная человеку задач а обязательно *имеет* решения. Напротив, допустимо считать, что некоторая задача понятна человеку, но всякая вещь, предъявляемая его вниманию, оказывается *нерешением* этой задачи.

2. Теоретические и практические задачи

Условимся говорить: человек имеет дело с *теоретической* задачей, когда он понимает её таким образом, что заранее согласен с тем, что всякая вещь, если она окажется решением, непременно будет некоторым *текстом* на предварительно выбранном им языке. В любом другом случае мы предлагаем говорить, что человек имеет дело с *практической* задачей.

Очевидно, найти решение, если оно есть, какой-либо *теоретической* задачи — это произвести и опознать как утоление исходного беспокойства некоторый подходящий *текст*. А найти решение, если оно есть, *практической* задачи — это произвести и опознать как утоление исходного беспокойства нечто, вообще говоря, *отличное* от текста (например, пару — текст *плюс* эксперимент).

Поясним пользу только что введённого разграничения задач простейшим примером. Допустим, что требуется отыскать (вычислить) целое число, удовлетворяющее конкретному, но весьма сложному для ручной проверки математическому условию. Обычно в таком случае говорят, что тут — задача для программиста. Однако, это не совсем так. На самом деле здесь *две* задачи: одна — теоретическая — найти текст, состоящий из текста некоторой программы и текста соответствующего обоснования этой программы, и другая — практическая — фактически (экспериментально) осуществить на данном компьютере реализацию счёта по указанной программе. За первую задачу отвечает программист. За вторую — тот, кто поддерживает исправность компьютера. Если не ввести рассматриваемое разграничение задач, то не ясно, как правильно распределить ответственность за возможное неверное исполнение исходной потребности[2].

Так как все мыслимые тексты выбранного языка можно эффективно перечислить, то решение любой теоретической задачи может быть найдено, *если оно вообще есть*, на одном из шагов следующей тривиальной процедуры. Поочерёдно, в каком-либо заданном порядке, человек выписывает и пытается опознать как подходящие в качестве решения всевозможные тексты на приня-

---

[2]Другое дело, что первую (теоретическую) из этих двух задач не следует рассматривать совершенно изолировано от второй (практической). Иначе может получится конфуз — процесс фактического счёта на данном оборудовании будет ненадёжным (например, неустойчивым) или чрезмерно дорогим и т.п. В конце концов, вообще, первую задачу сразу желательно формулировать в виде: найти и обосновать *такую* программу поиска нужного числа, чтобы её реализация на данном компьютере была приемлем по всем данным параметрам.

том языке. Поскольку единственная важная при этом опознании характеристика текста есть его способность вызывать или не вызывать чувство (непосредственно испытываемое, раз задача человеку понятна) утоления исходного беспокойства, то в указанном механическом перечне рано или поздно появится любой текст, если он вообще есть, непосредственно опознаваемый как убедительное решение. *Если же теоретическая задача не имеет решения*, то указанная процедура не имеет конца.

Желая подчеркнуть эти две возможности, мы говорим, что всякая теоретическая задача *полуразрешима*.

3. Математические и формальные теоретические задачи

Ввиду того, что возможные решения любой теоретической задачи — некоторые тексты, то иногда полезно классифицировать теоретические задачи по виду этих текстов. В частности, если они (тексты) предполагаются математическими доказательствами, то соответствующая теоретическая задача называется *математической*. Решениями любой такой задачи являются некоторые доказательные рассуждения. Если же упомянутые тексты — формальные выводы в какой-либо заданной формальной системе[3], то соответствующая теоретическая задача называется *формальной* задачей. Решениями формальной задачи являются некоторые выводы в данной формальной системе, безошибочно опознаваемые в качестве таковых.

Нетрудно дать подробную классификацию формальных задач. Пусть, например, T — произвольное формальное исчисление. Условимся любую формальную задачу, для решения которой оно может быть употреблено, называть *T-задачей*. Тогда всякая T-задача — это некоторый алгоритм разбиения множества всех выводов в T на два эффективно разрешимых подмножества: подмножество выводов, опознаваемых как *решения* рассматриваемой задачи, и подмножество остальных выводов. И, наоборот, каждый алгоритм разбиения множества всех выводов

[3]Произвольное исчисление называется *формальным*, если эффективно разрешимы множество всех выражений языка этого исчисления и множество всех выводов в нём.

в T и два эффективно разрешимых подмножества — это некоторая T–задача, *если* при этом одно из подмножеств *объявляется* состоящим из *решений* такой T–задачи, второе — из *нерешений*. Если первое подмножество не пусто — задача имеет решение. Если пусто, — то не имеет решения. Всякое однозначно понимаемое (в рамках того или иного заранее принятого лингвистического соглашения) описание любого алгоритма такого разбиения — *формулировка* соответствующей T–задачи. Если некоторая формулировка некоторой T–задачи является выражением языка системы T, то эта формулировка называется *внутренней*, или формулировкой *в* (*внутри*) T. В любом другом случае она называется *внешней*, или формулировкой *вне* (*за пределами*) T.

Легко понять, что *любое* выражение φ в языке исчисления T можно (по очевидному лингвистическому соглашению) считать формулировкой в T какой–то T–задачи, так как φ эффективно определяет разбиение множества всех выводов в T на два подмножества: подмножество выводов, оканчивающихся данным выражением φ и подмножество остальных выводов. О любом выводе из первого подмножества мы говорим, что он *решает* (*является решением*) соответствующую(ей) *формальную*(*ой*) *T–задачу*(*и*) φ. О любом выводе из второго подмножества мы говорим, что он *не решает* (*не является решением*, или *является нерешением*) φ.

4. T–задачи "на доказательство" и T–задачи "на построение"

Чтобы пояснить определение формульных T–задач, рассмотрим простейший пример. Пусть PA — формальная арифметика Пеано в языке со стандартной арифметической сигнатурой, и пусть φ_0 — следующая формула этого языка: $x = 0'$. Очевидно, соответствующая формульная PA–задача φ_0 не имеет решений в PA, и так как ни один вывод в PA не оканчивается формулой φ_0 (если, конечно, PA непротиворечива). С другой стороны, формулу φ_0 можно воспринимать как запись уравнения и по этом основании считать её формулировкой PA–задачи α_0: найти решение этого уравнения. Очевидно, α_0 *имеет* решения в PA — к ним принадлежит всякий вывод в PA тождества $0' = 0'$. Следовательно, существуют PA–задачи, которые не являются формульными PA–задачами.

На этом примере читатель должен осознать, что, вообще, для любой формальной системы T множество формульных T–задач — *собственное* подмножество множества всех T–задач.

Несколько злоупотребляя школьной терминологией, можно называть такие T–задачи, которые *не* являются формульными T–задачами, задачами *на построение* (*в* T), а формульные T–задачи — задачами *на доказательство* (*в* T).

Что же касается *формульных* T–задач, то их можно называть, также несколько злоупотребляя школьной терминологией, задачами *на доказательство* (*в* T).

К этому следует добавить, что решение *любой* (с любой внутренней или внешней формулировкой) T–задачи некоторым определённым образом сводится к эффективному перебору и решению некоторых *формульных* T–задач. В самом деле, всякое решение любой T–задачи α, оканчивающееся, например, выражением φ, есть в то же время решение некоторой формульной T–задачи, а именно T–задачи φ. С другой стороны, для любого эффективно заданного исчисления T множество всех формульных T–задач эффективно перечислимо. Поэтому, эффективно перечисляя решения формульных T–задач и проверяя на каждом шаге этого перечисления, является ли рассматриваемое в данный момент решение формульной T–задачи φ решением также исходной T–задачи α, мы рано или поздно решим эту последнюю задачу, если она вообще имеет решение.

Таким образом, представляется оправданным считать, что возможности использования любого данного формального исчисления T для решения формальных теоретических задач предопределяются составом и свойствами множества всех формульных T–задач и множества всех T–задач на построение.

В [1–4] точным образом исследуется зависимость строения множества всех формульных T–задач от T в том частном случае, когда T–формальное исчисление в первопорядковом языке. В [5–7] исследуются некоторые представляющие специальный интерес для философии математики T–задачи на построение в T с формулировками в T, когда T — произвольное формальное исчисление в произвольном первопорядковом языке, объемлющем язык арифметики.

Читатель может подумать, что приведённая классификация задач мотивирована и полезна (если вообще полезна) исключительно в рамках занятий так называемыми "точными" науками. Однако мы склонны думать иначе. Нам представляется, что как раз сугубо гуманитарные дисциплины — такие, как этика, эстетика, вообще философская антропология — особенно изобилуют вопросами, которые выражают скорее томления, чем задачи. Поэтому именно на этой территории наша классификация приобретает, если мы не обольщаемся, поучительный дисциплинирующий характер.

Литература

1. САМОХВАЛОВ К.Ф. Относительно конструктивные системы //Языки спецификаций и логическое программирование. — Новосибирск, 1988. — Вып. 124: Вычислительные системы. — С. 99-113.

2. ОДИНЦОВ С.П. О связи относительно конструктивных систем с традиционными подходами //Теория алгоритмов и её приложения. — Новосибирск, 1989. — Вып. 129: Вычислительные системы. — С. 172–182.

3. КАЗАКОВ Е.В., МОСКВИТИН А.А., САМОХВАЛОВ К.Ф. Проект разработки языков спецификаций задач, ориентированных на пользователя //Модели когнитивных процессов. — Новосибирск, 1997. — Вып. 158: Вычислительные системы. — С. 63–94.

4. САМОХВАЛОВ К.Ф. Рабочая характеристика первопорядкового исчисления //Труды науч.-исслед. семинар Логического центра Института философии РАН. — М., 1999. — С. 96–104.

5. ЕРШОВ Ю.Л., САМОХВАЛОВ К.Ф. О новом подходе к философии математики //Структурный анализ символьных последовательностей. — Новосибирск, 1981. — Вып. 101: Вычислительные системы. — С. 141–148.

6. ЕРШОВ Ю.Л., САМОХВАЛОВ К.Ф. О новом подходе к методологии математики //Закономерности развития современной математики. — М., 1987. — С. 85–106.

7. ГОНЧАРОВ С.С., ЕРШОВ Ю.Л., САМОХВАЛОВ К.Ф. Введение в логику и методологию науки. — М.: Интерпракс, 1994.

Поступила в редакцию
19 декабря 2001 года