

**О СОДЕРЖАТЕЛЬНОЙ ИНФОРМАЦИИ И РЕШЕНИИ ЗАДАЧ****Попов С.В.***ООО «Научно-внедренческая фирма БП+», Москва, e-mail: s-v-popov@yandex.ru*

В связи с использованием искусственного интеллекта (ИИ) для моделирования когнитивной деятельности человека интерес представляет зависимость сложности решения задач от спецификации предметной области, в которой задачи формулируются. Встречаются просто формулируемые задачи со сложным решением, и наоборот. В связи с этим желательно иметь универсальный показатель сложности задачи, который определял бы сложность ее решения. Исходим из того, что всякая задача, так или иначе, предполагает разбиение множества объектов предметной области на классы, определяющие подзадачи. В статье предполагается, что все отношения эквивалентности над множеством объектов образуют частично упорядоченное множество. Чем сильнее эквивалентность, тем больше неопределенности устраняется при решении и тем компактнее само решение. Так как невозможно ввести числовой параметр, измеряющий силу эквивалентности, то предлагается представить порождение классов как результат анализа состава объектов. Тогда можно построить бинарное дерево с единственным корнем, каждый путь которого описывает в точности один объект. Если в этом дереве ввести склейку узлов, представляющих один класс, то, чем сильнее эквивалентность, тем больше склеек в дереве. И все пути, ведущие из корня в один узел, определяют один класс, связанный с уникальной процедурой, которую необходимо выполнить, чтобы найти общее решение задачи. После этого вводят параметр *неопределенности* бинарного дерева, полагая, что дуги, ведущие из одного узла в его непосредственные потомки, равновероятны. Такая неопределенность связана с неопределенностью того, какие процедуры будут использоваться при решении задачи. При известном частотном распределении объектов относительно классов можно подсчитать неопределенность как характеристику предметной области. Показывается, что эта неопределенность может служить единственным параметром, определяющим сложность предметной области, который помогает оценивать сложность процедуры, решающей исходную задачу.

**Ключевые слова:** задача, эквивалентность, энтропия, предметная область, перебор, вычисление, решение, бинарное дерево

**ABOUT MEANINGFUL INFORMATION AND THE SOLVING PROBLEMS****Popov S.V.***LLC «Nauchno-vnedrencheskaya firma BP+», Moscow, e-mail: s-v-popov@yandex.ru*

In connection with the use of artificial intelligence (AI) for modeling human cognitive activity, it is of interest to study the dependence of the complexity of problem solving on the specification of the subject area in which the tasks are formulated. There are simply formulated problems with a complex solution, and vice versa. In this regard, it is desirable to have a universal indicator of the complexity of the problem, which would determine the complexity of its solution. We proceed from the fact that every task, one way or another, involves splitting a set of objects of the subject area into classes that define subtasks. The article assumes that all equivalence relations over a set of objects form a partially ordered set. The stronger the equivalence, the more uncertainty is eliminated in the solution, and the more compact the solution itself. Since it is impossible to enter a numerical parameter measuring the strength of equivalence, it is proposed to present the generation of classes as a result of the analysis of the composition of objects. Then you can build a binary tree with a single root, each path of which describes exactly one object. If a gluing of nodes representing one class is introduced in this tree, then the stronger the equivalence, the more glues there are in the tree. And all paths leading to one node define one class associated with a unique procedure that must be performed to find a common solution to the problem. After that, the uncertainty parameter of the binary tree is introduced, assuming that the arcs leading from one node to its immediate descendants are equally probable. Such uncertainty is related to the uncertainty of which procedures will be used in solving the problem. With a known frequency distribution, the distribution of objects relative to classes can be calculated uncertainty as a characteristic of the subject area. It is shown that this uncertainty can serve as the only parameter determining the complexity of the subject area, which helps to assess the complexity of the procedure solving the original problem.

**Keywords:** problem, equivalence, entropy, subject area, iteration, calculation, solution, binary tree

В связи с широким проникновением искусственного интеллекта (ИИ) в различные сферы для моделирования когнитивной деятельности человека интерес представляет исследование сложности решения задач в зависимости от вида предметной области (ПО), в которой они формулируются. Чтобы установить такую связь, в первую очередь следует определить параметры, от которых зависит сложность ПО. Естественно, что это определяется формулируемой задачей, поскольку в одной ПО можно сформулировать несколько задач,

и сложность их решения не всегда определяется сложностью формулировки. Встречаются просто формулируемые задачи со сложным решением, и наоборот. В связи с этим желательно иметь универсальный показатель сложности задачи в конкретной ПО, который определял бы сложность ее решения. Как представляется, такой показатель должен быть связан с содержательной сложностью задачи независимо от краткости или громозкости ее формулировки. В настоящей статье предлагается подход, позволяющий ввести показатель

сложности задачи с опорой на ее содержательную формулировку.

Полагаем, что  $O$  есть множество объектов ПО. Каждый из них однозначно характеризуется конечным набором признаков. Например, это могут быть множество логических матриц с фиксированным значением  $n$  строк и  $m$  столбцов или множество образов, заданных своими элементами, где каждый элемент принадлежит конечному алфавиту. Одновременно с множеством  $O$  рассматривается множество  $O^*$  так называемых *недоопределенных* объектов, каждый из которых содержит лишь часть наборов признаков, характеризующих объекты. Например, если в качестве объектов рассматриваются матрицы с  $n$  строками, то в качестве недоопределенных объектов можно рассматривать матрицы с  $n_1 < n$  строками. Если объекты – это некоторые образы, то недоопределенные объекты – суть их собственные части. Понятно, что каждый недоопределенный объект можно доопределить, в общем случае, разными способами до объекта ПО. Будем полагать, что если недоопределенный объект  $o^*$  доопределяется до объекта  $o$ , то это происходит в результате конкатенации  $o^*$  с некоторым недоопределенным объектом  $o_1^*$ , что будем обозначать  $o = o^*o_1^*$ . Введем пустой недоопределенный объект  $\lambda$ , который можно доопределить до любого объекта. Однако для объекта или недоопределенного объекта  $o$  верно равенство  $o = o\lambda$ . Поэтому полагаем, что справедливо включение  $O \subset O^*$ . И в дальнейшем будем употреблять общий термин *объект*, выделяя *недоопределенные объекты* по мере необходимости.

Полагаем, что решаемая задача  $Z$  над ПО с множеством  $O$  объектов состоит в установлении некоторого свойства  $P$  над  $O$ . Поэтому задача  $Z$  определяется на множестве  $O^*$  отношении эквивалентности объектов  $o_1, o_2 \in O^* (o_1 \approx o_2 \Leftrightarrow \forall o^* \in O^* (o_1 o^* = o_2 o^* \Rightarrow P(o_1 o^*) = P(o_2 o^*)))$ . Однако два недоопределенных объекта  $o_1, o_2 \in O^*$  называются *разделяемыми*, если выполняется соотношение:  $\exists o^* \in O^* (o_1 o^* = o_2 o^* \Rightarrow P(o_1 o^*) \neq P(o_2 o^*))$ . В этом случае  $o^*$  называется *разделяющим объектом*.

Каждый класс эквивалентности определяет свою собственную подзадачу, которую необходимо решить, чтобы получить искомое решение общей задачи. В логике это называется *разбором случаев*. Например, задачу над множеством целых чисел можно представить как две задачи: одна предполагает рассматривать четные числа в качестве новой ПО, другая – нечетные. Другой пример: решение числовых головоломок. В процессе решения происходит

разбор случаев, каждый характеризуется уникальной комбинацией замен букв цифрами и требует своего решения. Поэтому исходная задача представляется совокупностью подзадач, решение которых приведет к окончательному результату.

Все отношения эквивалентности, которые можно ввести над множеством объектов ПО, образуют частично упорядоченное множество, по включению. Чем слабее эквивалентность, тем уже ее классы эквивалентности, т.е. тем слабее условия, при которых объекты эквивалентны. Самая сильная подразумевает эквивалентность всех объектов ПО. Самая слабая характеризуется классами, состоящими из одного объекта. В этом случае бессмысленно говорить о какой-либо связи объектов, задача не имеет содержательного смысла.

Для решения нетривиальной задачи над ПО обязательно наличие априорной информации о факторизации множества объектов. Это устраняет неопределенность, неизбежную при поиске решения и связанную с объединением в классы семантически совпадающих объектов. Чем сильнее отношение эквивалентности, т.е. чем шире ее классы, тем больше неопределенности устраняется при решении и тем компактнее решение. Например, при решении геометрической задачи, чем большим числом доказанных соотношений между фигурами располагаем, тем шире классы эквивалентности фигур на чертеже. И, как следствие, тем компактнее решение, так как всякое неизвестное, но обязательное для решения соотношение надо доказывать.

Пусть  $\pi$  есть всюду определенная арифметическая программа, вычисляющая предикат  $P$ . Если на ее вход поступает объект  $o \in O$ , то программа завершает работу в точности в одном из двух финальных состояний:  $c_0$  либо  $c_1$ , в зависимости от значения  $P(o)$ , соответственно 0 или 1. Если на вход программы поступает недоопределенный объект  $o^*$ , то он определяет одно вычисление  $\pi(o^*)$ , заканчивающееся в некотором, в общем случае, не финальном состоянии  $c^*$ . Поскольку  $\pi$  всюду определенная, то если на ее вход поступает любой объект, то ее вычисление завершается в некотором состоянии. Однако каждое вычисление программы  $\pi$  вызывается некоторым объектом  $o^*$ . Это вычисление может быть продолжено за счет различных вычислений, которые получаются, если объект  $o^*$  доопределять различными способами.

Следующим образом введем отношение  $\pi$ -эквивалентности: для объектов  $o_1, o_2 \in O^* (o_1 \approx_{\pi} o_2 \Leftrightarrow \forall o^* \in O^* (o_1 o^* = o_2 o^* \Rightarrow \text{вычисления } \pi(o_1 o^*) \text{ и } \pi(o_2 o^*) \text{ заканчива-$

ются в одном состоянии). Очевидно, что для эквивалентных объектов из  $\mathcal{O}$  финальные состояния совпадают. Отметим, что состояния, в которых завершаются вычисления  $\pi(o_1)$  и  $\pi(o_2)$ , могут быть различными. Но при всяком одинаковом продолжении  $o^*$  объектов  $o_1$  и  $o_2$  до полностью определенных объектов эти продолжения заканчиваются всегда в одинаковом состоянии, т.е. объекты  $o_1 o^*$  и  $o_2 o^*$  одновременно либо обладают, либо не обладают свойством  $P$ . В результате получаем следование:  $\forall o_1, o_2 \in \mathcal{O}^* (o_1 \approx o_2 \Rightarrow o_1 o^* \approx o_2 o^*)$ .

Покажем, что верно и обратное следование. Пусть два недоопределенных объекта  $o_1, o_2 \in \mathcal{O}^*$  находятся в отношении  $o_1 \approx o_2$ , и вычисления  $\pi(o_1)$  и  $\pi(o_2)$  заканчиваются в состояниях соответственно  $c_1$  и  $c_2$ . Допустим, что имеется такое продолжение  $o^*$ , что  $o_1 o^*, o_2 o^* \in \mathcal{O}$ , но из этих двух состояний  $c_1$  и  $c_2$  можно продолжить вычисления в разные финальные состояния. Следовательно, имеется одно продолжение для эквивалентных объектов, которое переводит два состояния  $c_1$  и  $c_2$  в два разных финальных состояния. Следовательно, продолжение  $o^*$  – разделяющее, поэтому объекты  $o_1, o_2$  не эквивалентны. Это противоречие. Следовательно, верно соотношение:  $\forall o_1, o_2 \in \mathcal{O}^* (o_1 \approx o_2 \Rightarrow o_1 o^* \approx o_2 o^*)$ .

Полагаем, что каждый объект из множества  $\mathcal{O}$  представляется бинарным вектором, а недоопределенные объекты представляются векторами над базисом  $\langle 0, 1, \_ \rangle$ . Здесь символом  $\_$  обозначены неизвестные или несущественные признаки. Поэтому все объекты ПО можно представить в виде бинарного дерева  $T$  с единственным корнем, в котором каждый узел имеет двух непосредственных потомков, один порождается дугой с меткой 0, другой – 1. Каждый путь из корня описывает в точности один, в общем случае, недоопределенный объект. Пути из корня, заканчивающиеся в висячих узлах, определяют объекты из множества  $\mathcal{O}$ .

Теперь, если в бинарном дереве ввести склейку узлов, представляющих один класс, то сила эквивалентности определяет число склеек. Чем сильнее эквивалентность, тем больше склеек в дереве. В результате висячие узлы любого поддерева, содержащего корень, представляют классы, объекты которых определяются путями, ведущими в один узел. А все пути, ведущие в один узел, определяют один класс. Последний определяет подзадачу, решение которой необходимо в рамках общего решения задачи. Однако невозможно ввести числовой параметр, который измерял бы силу априорно известной эквивалентности объектов ПО, поскольку эквивалентности образуют ча-

стично упорядоченное множество по включению. Тем не менее, силу эквивалентности можно охарактеризовать количеством склеек в бинарном дереве: чем больше склеек, тем эквивалентность сильнее.

Поскольку один узел определяет класс эквивалентности объектов, то сечение  $S$  дерева  $T$  определяет классы эквивалентности объектов, характеризующихся фиксированными наборами параметров. Тогда множество  $S$  определяет вычисления программы  $\pi$ , оканчивающиеся, по меньшей мере, в  $|S|$  различных состояниях. Для кодирования этих состояний в программе  $\pi$  требуется память не меньшая, чем  $\log_2 |S|$  бит [1–4]. Однако все объекты из  $S$  не могут быть закодированы  $s$  меньшей, чем  $\log_2 |S|$ , сложностью. Следовательно, в программе  $\pi$  имеется вычисление длины не меньше  $|S|$ . Так удалось оценить память арифметической программы в зависимости от максимального сечения бинарного дерева, представляющего объекты ПО.

Введем параметр *неопределенности* бинарного дерева, полагая, что дуги, ведущие из одного узла в непосредственные потомки, равновероятны. Неопределенность в таком случае связана с неопределенностью того, какие подзадачи будут решаться при решении общей задачи. Известно число подзадач, известно частотное распределение, пусть  $p_1, p_2, \dots, p_q$ , объектов ПО, с которыми они попадают в  $q$  классов, т.е. используются в качестве аргументов подзадач. Следовательно, можно подсчитать неопределенность  $H$  совокупности этих классов как характеристику исходной задачи над ПО [5].

Покажем, что так введенная неопределенность может служить параметром, определяющим сложность задачи. Для этого свяжем неопределенность задачи как неопределенность разбиения объектов ПО со сложностью вычислений программы, решающей исходную задачу. Здесь исходим из следующих соображений. Предполагается, что каждый класс определяет совокупность вычислений программы  $p$ , решающей исходную задачу, т.е. вычисляющей предикат  $P$ . Все вычисления программы представляются так называемой *разверткой*, т.е. бинарным деревом с единственным корнем, в котором вычислительные узлы не вызывают размножения вычислений, а логические представляются узлами с двумя потомками.

Связь между бинарным деревом разбора объектов ПО и разверткой устанавливается исходя из следующих соображений. Если в бинарном дереве разбора ПО имеются два узла, пусть  $n_0$  и  $n_1$ , определяющие разные классы объектов и, соответственно, различные подзадачи, то им в развертке  $R\pi$

программы  $\pi$  соответствуют два различных множества вычислений. Если допустить, что это не так, то приходим к заключению, что эквивалентность, определенную на множестве объектов ПО, можно усилить, объединив классы, соответствующие узлам  $n_0$  и  $n_1$ . В результате каждому поддереву  $T$  бинарного дерева разбора объектов ПО в разветвке  $R_p$  соответствует бинарное дерево вычислений с корнем, такое, что каждый узел  $n \in T$  определяет множество  $M_n$  вычислений. И разные узлы дерева  $T$  определяют разные множества вычислений. Тогда, если  $H$  есть неопределенность дерева  $T$ , то соответствующее бинарное поддерево разветки  $R_\pi$  имеет глубину не меньше, чем  $H$ , а число различных вычислений не меньше  $2^H$ . Так можно оценить как размер памяти программы  $\pi$ , так и число вычислений. Исходя из базиса арифметической программы оценивается сложность вычисления. Так, если базис состоит из логического оператора  $=$  и арифметических  $s(x)$  – прибавления единицы и  $0(x)$  – обнуления переменной, то средняя длина вычисления, определяемого поддеревом разбора ПО с неопределенностью  $H$ , не менее  $2^H$ .

**Информационные модели.** Считаем, что объект управления описывается уравнением  $y = f(x)$ , где  $x$  – это входные параметры,  $y$  – демонстрируемое поведение. Тогда его информационной моделью могут служить, например, таблицы базы данных, описывающие его поведение, или программа, вычисляющая функцию  $f(x)$ , и т.п.

**Пример 1.** Пусть информационная модель отражает совместимость комплектующих при сборке изделия с заданным функционалом. Если она адекватна, то удастся избежать сборки неправильно функционирующего изделия.

**Пример 2.** Пусть информационная модель отражает совместимость химических компонентов при синтезе вещества с заданными качествами. Если она не адекватна, то возможен синтез вещества с незапланированными качествами.

Сформулируем некоторые аналогии между понятиями теории информации и введенными представлениями. *Передачу информации* соответствует *наблюдаемая часть модели*, а *сигналу* – поведение этой части, выраженное в тех или иных терминах. *Канал связи* соединяет наблюдаемую часть модели с частью, на которую осуществляется воздействие. Приемником информации служит еще не наблюдаемая часть модели, на которую воздействует активная, функционирующая часть. И в зависимости от сигнала приемник переходит в какое-либо состояние.

**Пример 3.** Пусть  $M$  есть логическая матрица размерности  $[n \times m]$ ,  $M = M_1 + M_2$ ,  $M_1$  имеет размерность  $[n_1 \times m]$ ,  $M_2 = [n_2 \times m]$ ,  $n = n_1 + n_2$ . *Воздействиями* на модель является множество  $\Sigma$  всех означиваний  $n_1$  строк матрицы  $M_1$ . Эта подматрица есть передатчик, который преобразует воздействия в сигналы. Каналом является горизонталь, разделяющая матрицы  $M_1$  и  $M_2$ . Приемником является матрица  $M_2$ . Каждое воздействие  $\sigma \in \Sigma$  превращает в истину часть  $H\sigma$  столбцов матрицы  $M_1$ . Множество истинных столбцов  $H\sigma$  есть сигнал, передающийся от передатчика приемника. Приемник реагирует на такой сигнал однозначно, при последующем рассмотрении эти столбцы вычеркиваются из матрицы  $M_2$ . В результате выходные сигналы канала воздействуют на приемник, изменяя его состояние. Таким образом, воздействия  $\Sigma$  формируют выходные сигналы, а именно множество подматриц матрицы  $M_2$ .

Здесь под моделью понимается дискретная функция  $\varphi: A \rightarrow B$ , где  $A$  есть множество входных воздействий,  $B$  – выходных. Значения ее аргумента  $x$  представляют собой бесконечные бинарные слова. Каждое слово представляется конкатенацией конечной префиксной части, заканчивающейся самой правой единицей, и нулевым бесконечным суффиксом. Говорим, что двоичная последовательность, означающая часть аргументов  $x$ , является *частичным x-набором*. Аналогично будем говорить о *частичном y-наборе* выходных значений. Анализ поведения функции  $\varphi$  сводится к проверке равенства  $\varphi(\sigma_x) = \sigma_y$  для всякой пары  $(\sigma_x, \sigma_y)$  соответственно (в том числе частичных)  $x$ -набора и  $y$ -набора.

Допустим, что при всяком допустимом означивании подмножества  $x' \subset x$  входных параметров известна выходная реакция модели. В результате модель  $\varphi$  представляется объединением  $\varphi_1, \varphi_2, \dots, \varphi_q$  подмоделей, число которых не больше числа частичных  $x$ -означиваний. Следовательно, анализ поведения модели в зависимости от частичных входных параметров  $x'$  сводится к двум задачам.

1. Выяснить тип поведения из известных  $\varphi_1, \varphi_2, \dots, \varphi_q$ , демонстрируемый при известном входном векторе  $\sigma_x$ .

2. Исследовать каждую из подмоделей  $\varphi_1, \varphi_2, \dots, \varphi_q$ .

Пусть событие  $A$  встречается с частотой  $p_A$  и  $\gamma(A) = -\log_2 p_A$  есть показатель его неожиданности, зависящий от частоты  $p_A$ . Величину  $\gamma(A)$  назовем *частной неопределенностью  $H_A$  события  $A$* . Если имеются  $q$  различных значений  $A_1, A_2, \dots, A_q$  одной случайной величины, каждое из которых

встречается с частотой соответственно  $p_1, p_2, \dots, p_q$ , то усредненная неопределенность равна  $H = -\sum_{i=1,q} p_i \gamma(A_i)$ .

Если частичный набор  $\sigma_x'$  означает часть аргументов  $x' \subset x$ , то аргументы  $x'$  назовем *неподвижными*, а оставшиеся – *изменяемыми*. При означиваниях изменяемых аргументов так, что результирующие наборы принадлежат области определения  $Def \phi$ , какие-то компоненты  $y' \subset y$  могут принимать значения, не меняющиеся при означивании оставшихся компонентов аргумента. В этом случае  $x$ -набор  $\sigma_x'$  определяет функцию  $\phi | \sigma_x'(\mathbf{u})$ :

1) переменные  $\mathbf{u}$  суть изменяемые аргументы;

2) область определения  $Def \phi | \sigma_x'$  получается в результате ограничения области  $Def \phi$  означиваниями,  $x'$ -проекции которых совпадают с  $\sigma_x'$ ;

3) функция  $\phi | \sigma_x'(\mathbf{u})$  получается из функции  $\phi$  при аргументах с фиксированным частичным  $x$ -означиванием  $\sigma_x'$  игнорированием неподвижных  $y$ -компонентов.

**Пример 4.** Пусть  $\sigma_x = 01011 = 26_{10}$  есть частичное  $x$ -означивание аргументов двоичной функции прибавления единицы:  $\phi(x) = x + 1$ . Полагаем, что младшие разряды располагаются слева. Тогда функция  $\phi | \sigma_x'(\mathbf{u}) = 2^5 \mathbf{u}$ , поскольку неподвижные компоненты значений функции суть компоненты с нулевого до четвертого включительно, а остальные – изменяемые, и отсутствует перенос в пятый разряд. Все значения функции  $\phi(x)$  имеют своим префиксом 11011, если ограничить область ее определения двоичными аргументами с префиксом 01011.

Говорим, что  $x$ -наборы  $\sigma_x'$  и  $\sigma_x''$  эквивалентны относительно функции  $\phi$ , если функции  $\phi | \sigma_x'(\mathbf{u})$  и  $\phi | \sigma_x''(\mathbf{u})$  совпадают.

**Пример 5.** Для функции  $\phi(x) = x + 1$  означивания:  $\sigma_x' = 01011$ ,  $\sigma_x'' = 11111$ ,  $\sigma_x''' = 11011$  определяют:  $\phi | \sigma_x'(\mathbf{u}) = 2^5 \mathbf{u}$ ,  $\phi | \sigma_x''(\mathbf{u}) = 2^5 \mathbf{u}$ ,  $\phi | \sigma_x'''(\mathbf{u}) = 2^5 \mathbf{u} + 1$ . Префиксы значений функции  $\phi(x)$  суть следующие: 11011, если префикс аргумента  $\sigma_x'$ ; 0000, если префикс  $\sigma_x''$ ; 00111, если префикс  $\sigma_x'''$ . При  $\sigma_x''$  возник перенос в пятый разряд, в первом и третьем случаях переноса нет. Поэтому функция  $\phi | \sigma_x'''(\mathbf{u})$  отлична от  $\phi | \sigma_x'(\mathbf{u})$  и  $\phi | \sigma_x''(\mathbf{u})$ , наборы  $\sigma_x'$  и  $\sigma_x''$  эквивалентны, а  $\sigma_x'''$  не эквивалентен им.

Эквивалентность  $x$ -наборов сводит исследование исходной модели к описанию совокупности подмоделей. Число  $t$  классов определяет  $t$  различных подмоделей. И, если каждая подмодель предполагает собственное вычисление, то имеется  $t$  различных вычислений. Следующий пример де-

монстрирует содержательные рассуждения и введенные определения.

**Пример 6.** Пусть  $\pi(x)$  есть двоичная арифметическая программа, представленная своей диаграммой с двумя типами узлов: логических и арифметических. Тогда ее вычисления задаются входными значениями. Под *вычислением* понимаем последовательность арифметических и логических операторов, выполняющихся при заданном входе до остановки. В результате выходная переменная  $y$  принимает определенное значение.

Полагаем, что означивания  $\sigma_1$  и  $\sigma_2$  части  $x'$ ,  $|x'| = l$ , аргумента эквивалентны ( $\sigma_1 \approx \sigma_2$ ), если определяемые ими вычисления завершаются в одном и том же состоянии. Поэтому вычисления разбиваются на классы, определяемые эквивалентными означиваниями. И каждое частичное означивание определяет подпрограмму, выполняемую после продолжения этого означивания. Пусть  $\pi_1, \pi_2, \dots, \pi_q$  суть выделяемые подпрограммы, множество вычислений, определяющих подпрограмму  $\pi_i$ , обладает долей  $p_i$ ,  $i = 1, 2, \dots, q$  среди всех вычислений над частичными входами длины  $l$ . Тогда неопределенность вопроса: какое событие из  $\pi_1, \pi_2, \dots, \pi_q$  наступит при означивании  $\sigma_x$  части  $x'$  аргументов,  $|x'| = l$ , зависит лишь от долей  $p_1, p_2, \dots, p_q$ , и равна  $H_1 = -\sum_{i=1,q} p_i \log p_i$ .

Исследование программы как совокупности  $\pi_1, \pi_2, \dots, \pi_q$  подпрограмм, обладающих собственными множествами вычислений, соответственно  $T_1, T_2, \dots, T_q$ ,  $i = 1, 2, \dots, q$ , обладает неопределенностью  $H_2 = \sum_{i=1,q} p_i \log |T_i|$ .

Понятно, что исследование программы приводит к разным значениям неопределенности. Если  $l = 0$ , то неопределенность максимальная, поскольку не известны возможные вычисления. Когда известны все вычисления, то неопределенность равна нулю. В случае частичных означиваний,  $0 < l < n$  неопределенность равна сумме  $H_1 + H_2$ .

Обобщим предыдущие рассуждения следующим образом.

Пусть *Tree* есть растущее вниз однокорневое бинарное дерево, каждый висячий узел которого обозначает некоторый класс эквивалентности объектов ПО, и его энтропия равна  $H$ . Полагаем, что введенная эквивалентность определяет семантику ПО, когда одинаковые объекты суть элементы одного класса эквивалентности. Бинарное дерево с минимальной длиной пути из корня в листья и энтропией  $H$  имеет глубину  $H$  и в точности  $2^H$  висячих узлов.

Свяжем энтропию дерева *Tree* со сложностью вычислительной процедуры, ко-

торая не различает эквивалентные объекты, выдавая для них одинаковые ответы. Для этого допустим, что имеется некоторая вычислительная процедура  $\pi(x)$ , вычисляющая некоторую двоичную всюду определенную функцию  $y = f(x)$  от одного аргумента. Полагаем, что в процессе вычисления процедура  $\pi$  переходит из одного состояния в другое, и ее состояния представляются *внутренней памятью*, которая есть бинарный вектор неограниченной длины. При отсутствии входа процедура находится в фиксированном нулевом состоянии, когда память заполнена нулями. Каждому состоянию соответствует уникальный вид этого вектора с ненулевым ограниченным префиксом и бесконечным нулевым суффиксом. Каждый вход  $\sigma$  либо приводит к остановке процедуры в *финальном состоянии*  $End$ , либо процедура достигает промежуточного состояния  $S$ , из которого она продолжит вычисление при расширении входа новыми компонентами. Тем самым все входные последовательности разбиваются на классы эквивалентности: с каждым классом связано уникальное состояние процедуры, из которого она продолжает вычисление при расширении аргумента. И для разных классов продолжения вычисления различные.

Представим вычисления процедуры  $\pi(x)$  в виде следующего бинарного дерева  $Tree(\pi)$  (развертки). Корнем дерева служит состояние процедуры при пустом входе. Полагаем, что в корень ведет дуга, помеченная пустой меткой. Если некоторый узел  $n$  развертки представляет состояние  $S$  вычисления, полученное в результате подачи на вход процедуры бинарной последовательности  $\sigma$ , то из него выходят две дуги, помеченные 0 и 1. Каждая из них ведет в узел соответственно  $n_0$  и  $n_1$ , представляющие собой состояния, в которые переходит процедура  $\pi$  при входах соответственно  $\sigma 0$  и  $\sigma 1$ . Одинаковые состояния развертки склеиваются.

Пусть на вход процедуры  $\pi$  поступают все бинарные последовательности длины  $l$ . Часть ее развертки  $Tree(\pi, l)$ , представляющая собой начальное поддерево с длиной ветвей  $l$ , определяет эквивалентность на множестве этих входов: два входа эквивалентны, если соответствующие вычисления приводят к одному состоянию. Исходя из этого и полагая равновероятными дуги, ведущие из одного узла развертки и помеченные 0 и 1, подсчитаем энтропию  $H$  дерева  $Tree(\pi, l)$ . Тогда минимальная глубина двоичного дерева с энтропией  $H$  равна  $H$ , а число его висячих узлов –  $2^H$ .

Так как каждый висячий узел определяет одно состояние вычисления, то со-

стояний внутренней памяти будет не менее  $2^H$ . Следовательно, чтобы реализовать все вычисления, представленные деревом  $Tree(\pi, l)$ , необходима внутренняя память, содержащая не меньше  $H$  компонентов. В противном случае не удастся получить все классы эквивалентности входов процедуры.

В данном случае эквивалентность входов процедуры  $\pi$  определялась исходя из состояний самой процедуры. Также исходя из этого строилась развертка и определялась энтропия частичной развертки. Однако при изучении конкретной ПО нет априорной информации о поведении вычислительной процедуры. Поэтому интерес представляет такое определение эквивалентности над объектами ПО, чтобы разные классы определяли различные вычисления. Тогда по мощности фактор множества можно оценить размер памяти вычислительной процедуры, а затем исходя из особенностей ее вычислительных правил оценить сложность вычислений.

Исходя из этого рассмотрим вопрос, как выглядит неопределенность модели в зависимости от различных определений эквивалентности ее состояний при частичном означивании аргумента. В качестве примера рассмотрим формулу разложения логических функций (по переменной  $x$ )  $f(x) = x \wedge f(1) \vee \bar{x} \wedge f(0)$ , полагая, что логическая функция задана матрицей, представляющей ее КНФ. Тогда разложение по нескольким переменным можно представить бинарным деревом, в котором узлы суть логические матрицы, и при его построении матрицы, определяющие эквивалентные (в каком-либо смысле) функции, склеиваются. Эквивалентности в данном случае могут быть различные. Это могут быть: логическая эквивалентность, совпадение матриц с точностью до перестановки строк и столбцов, логическая эквивалентность, доказываемая за ограниченное число преобразований, буквальное совпадение матриц и т.д. Понятно, что все эквивалентности, не противоречащие формуле разложения, формируют частично упорядоченное множество с минимальным элементом – логической эквивалентностью (самой сильной) и максимальным – буквальным совпадением матриц (самой слабой). При различных эквивалентностях получаются различные бинарные деревья. При более сильной эквивалентности будет больше склеиваний, нежели при слабой.

Пусть при одном порядке переменных разложения исходной функции и при различных эквивалентностях  $E_1$  и  $E_2$ ,  $E_1 \leq E_2$  получаются два бинарных дерева  $T_1$  и  $T_2$ .

И при одной последовательности  $u$  разложения переменных получаются два сечения  $S_1$  и  $S_2$  бинарных деревьев  $T_1$  и  $T_2$ . Тогда энтропии  $H_1$  и  $H_2$ , определяемые сечениями соответственно  $S_1$  и  $S_2$ , находятся в отношении:  $H_1 \leq H_2$ . Следовательно, более сильная эквивалентность определяет меньшую энтропию, что вполне объяснимо: более сильная эквивалентность за счет более частой склейки узлов в бинарном дереве определяет меньшую энтропию. Содержательно это трактуется следующим образом: эквивалентность, подразумеваемая в разложении функции, устраняет часть неопределенности за счет дополнительных свойств рассматриваемых объектов. Поэтому неопределенность бинарного дерева меньше. Понятно, что так введенная энтро-

пия касается только вида бинарного дерева и не учитывает сложности вычисления эквивалентностей. А поэтому ничего нельзя сказать о реальной сложности вычислений, не располагая знаниями о сложности вычисления эквивалентности.

#### Список литературы

1. Шеннон К. Работы по теории информации и кибернетике. М.: ИЛ, 1963. 830 с.
2. Кудряшов Б.Д. Теория информации. СПб.: Изд-во Питер, 2018. 312 с.
3. Леонтьев В.К., Гордеев Э.Н. Комбинаторные аспекты теории информации. М.: МФТИ, 2019. 320 с.
4. Фурсов В.А. Лекции по теории информации: учеб. пособие / Под ред. Н.А. Кузнецова. Самара: Изд-во Самар. гос. аэрокосм. ун-та, 2016. 148 с.
5. Попов С.В. Логическое моделирование. М.: Тривант, 2016. 255 с.