Посов Илья Александрович,
Рукшин Сергей Евгеньевич

УДК 004.031.42

МОДЕЛЬ ОЦЕНКИ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ УЧАЩИХСЯ ПО РЕЗУЛЬТАТАМ РЕШЕНИЯ ЗАДАЧ

Аннотация

В статье описана модель оценки учебной деятельности произвольной формы на основе результатов решения задач. Модель необходима для проектирования систем, автоматизирующих вычисление результатов участия учеников в некотором учебном процессе. Кроме того, модель можно использовать для проектирования самой учебной деятельности, потому что она обобщает ряд принципов, на основе которых преподаватели выбирают способ оценки деятельности учащихся.

В статье предложена концепция упрощенной системы, автоматизирующей процесс подведения итогов преподавателем, вводящим вручную историю проверки сдаваемых задач.

Ключевые слова: образование, система проведения удаленных соревнований, модель соревнования, подведение итогов соревнования.

ВВЕДЕНИЕ

Любая учебная деятельность в области точных наук предполагает решение учащимися задач. В качестве примеров форм учебной деятельности можно привести работу студентов в течение учебного семестра, работу учеников в математическом кружке, тестирование, экзамен, олимпиаду. В зависимости от формы учебной деятельности результаты решения задач анализируются по-разному. В случае олимпиады внимание уделяется скорости, с которой учащимся было представлено правильное решение. В случае работы в течение семестра важным является, успел ли учащийся сдать решение задачи к определенному сроку. Различается также отношение преподавателя к неверным попыткам сдачи задачи. В некоторых случаях неверная попытка игнорируется, в некоторых – отрицательно влияет на оценку.

В данной статье авторы ставят цель описать модель произвольной формы учебной деятельности, предполагающей оценку учеников на основе результатов решения ими задач. Модель необходима для проектирования систем, автоматизирующих вычисление результатов участия учеников в некотором учебном процессе. Кроме того, модель можно использовать для проектирования самой учебной деятельности, так как она обобщает ряд принципов, на основе которых преподаватели выбирают способ оценки работ учащихся. Для олимпиад по программированию обобщение процесса подведения итогов предлагается в статье [1].

На основе предлагаемой модели авторы разрабатывают автоматизированную систему проведения соревнований DCES [2–6]. В данной статье на основе модели предлагается концепцию упрощенной системы, которая позволяет автоматически вычислять результаты работ учащихся на основе вводимых преподавателем данных, то есть в отличие от DCES, система не предполагает прямую работу с учащимися, а требует от преподавателя ручного ввода результатов проверки решений.

Далее для упрощения изложения, вместо произвольных форм учебных деятельностей, мы будем говорить только о «соревнованиях». Это не умаляет общности, но позволяет больше уделять внимания сравнению решений участников друг с другом, то есть говорить о том, что решения одних участников лучше решений других.

МОДЕЛЬ ЗАДАЧИ

Определим задачу как произвольное непустое множество с введенным на нем отношением линейного предпорядка. $P=(M,\leq)$. Мы называем отношением предпорядка транзитивное и рефлексивное отношение. В отличие от отношения порядка, предпорядок не требует выполнения свойства антисимметричности. Под линейным предпорядком мы понимаем предпорядок, у которого любые два элемента сравнимы, то есть для любых $a,b\in M$ либо $a\leq b$, либо $b\leq a$. Далее будем называть множество M с введенным на нем отношением линейного предпорядка упорядоченным множеством.

Такого минималистичного определения задачи достаточно для введения, но оно будет уточнено впоследствии.

Множество, соответствующее задаче, описывает все возможные ответы, которые могут быть даны на вопрос задачи, а отношение предпорядка необходимо для сравнения ответов. Другими словами, наличие отношения на множестве формализует тот факт, что некоторые решения задачи лучше других. Точнее, в соответствии с отношением, мы будем говорить, что одно решение задачи «не хуже» другого. Подобное представление задачи в виде упорядоченного множества M позволяет описать механизм оценки решений участников, но оно не подходит для обсуждения вопросов, связанных с представлением условия задачи. Будем считать, что участники каким-то образом сами понимают, какое множество ответов на задачу им доступно, и какой из ответов необходимо выбирать.

Множество ответов отличается для разных типов задач и для разных методов ввода ответов. Например, в случае, если в задаче при ответе на вопрос требуется выбрать один из нескольких вариантов ответа, множество M будет совпадать с множеством вариантов. Если для ответа на вопрос требуется ввести число, множество M будет являться соответствующим числовым множеством, если решением задачи является геометрическое построение, множество M будет множеством последовательностей операций, необходимых для построения чертежа.

В отличие от отношения порядка, отношению предпорядка не требуется быть антисимметричным, это означает, что задача может иметь два различных решения, каждое из которых не хуже другого. Такая ситуация соответствует тому, что некоторые решения задачи могут быть неразличимы для целей проверки, например, часто задача может иметь несколько правильных решений. В простейшем случае множество решений задачи разбивается на два подмножества $M=M_R\cup M_W$, соответственно, верных и неверных. Отношение предпорядка в подобной ситуации устроено следующим образом: $a\leq b\Leftrightarrow\neg(a\in M_R\wedge b\in M_W)$ для $a,b\in M$.

На множестве M, на котором задано отношение линейного предпорядка, могут быть заданы дополнительные производные отношения.

1. Отношение $\sim$ определяется как $a\sim b\Leftrightarrow a\leq b\wedge b\leq a$. Это отношение является отношением эквивалентности. Два решения задачи, удовлетворяющих этому отношению, будем называть «эквивалентными».

2. Отношение $<$ определяется как $a<b\Leftrightarrow a\leq b\wedge\neg\ a\sim b$. Это отношение является отношением строгого предпорядка, то есть транзитивным и антирефлексивным. Будем говорить, что решение a «хуже» решения b (b «лучше» a), если $a<b$.

На практике два решения редко сравниваются напрямую. Чаще каждое из решений исследуется независимо, результат проверки одного решения сравнивается с результатом проверки второго. Например, результатом проверки может являться число баллов, которое жюри выставило за решение задачи. В этом случае достаточно сравнить только выставленное число баллов. В связи с этим введем определение сводимости одного упорядоченного множества к другому.

Будем говорить, что упорядоченное множество $(M, \leq)$ сводится к упорядоченному множеству $(M', \leq')$, если существует отображение $f: M \to M'$, такое что

$$\forall a,b \in M \quad a \leq b \iff f(a) \leq' f(b).$$

Введенное отношение ~ позволяет описать устройство линейного предпорядка на множестве M: на фактормножестве $M/\sim$ существует отношение нестрогого линейного порядка M', такое что $a \leq b \Leftrightarrow \overline{a} \leq' \overline{b}$, где $\overline{a}$ – это класс эквивалентости в $M/\sim$, содержащий элемент a. При этом $f: a \to \overline{a}$ – это сведение упорядоченного множества M ко множеству $M/\sim$.

ИСТОРИЯ РЕШЕНИЙ

Определим понятие истории решений. При участии в соревновании участник посылает решение задачи несколько раз, каждую попытку ответа будем называть подходом. Количество подходов всегда ограничено, в некоторых случаях участнику доступно не более одного подхода, в других – несколько, возможно, не более трех, возможно двадцати. Если мы позволяем участнику неограниченное число подходов, то все равно стоит их ограничить количеством около 100, столько подходов нормальный участник сделать не может, а попытка атаковать систему большим количеством решений для проверки будет пресечена. Итак, можно считать, что каждый участник имеет ограниченное число подходов. Определим историю решений как конечное множество $H \subset [0,1] \times M$. Здесь M – это множество решений задачи $P = (M, \leq)$. Первое множество в прямом произведении – это момент посылки решения. Будем считать, что никакие два решения не могут быть посланы одновременно, то есть $(t_1, a_1) \neq \{t_2, a_2\} \in H \Rightarrow t_1 \neq t_2$. Моменты начала и окончания соревнования – 0 и 1 соответственно.

Приведем пример различия понятий «решение» и «история решений». Допустим, в задаче требуется найти как можно больше корней некоторого уравнения, и участник получает тем больше баллов, чем больше корней он найдет. Множество M в данном случае является *множеством подмножеств множества корней*. Другими словами, в одной посылке решения участник перечисляет все известные ему корни. Неправильно было бы формализовать данную задачу так, что одно решение соответствует одному корню, потому что тогда для такой задачи возникнет противоречие с ограничением на один подход (нельзя послать больше одного корня), и на множестве M не удастся естественным образом завести отношение предпорядка. Итак, за каждый подход участник посылает все известные ему корни.

Множеству историй решений в рамках одного соревнования соответствует некоторый линейный предпорядок. Он выражает тот факт, что результаты участников определяются не на основе одного решения задачи, а на основе всей истории их решений. Например, если участник послал правильное решение задачи, а другой тоже послал правильное, но перед этим посылал неправильное, обычно результат второго участника по задаче считается не лучше.

Приведем другой пример сравнения историй решений. Допустим, первый участник послал сначала правильное решение, а потом неправильное. Второй участник посылал решения в те же моменты времени, но сначала он послал неправильное, а потом правильное. Чей результат считать лучше? Если важным параметром для нас является скорость решения, что характерно для олимпиад, то лучшим будет тот, кто послал правильное решение быстрее. Если нам важна уверенность участника в своем решении, лучшим будет тот, у кого правильное решение является последним. В любом случае каждое соревнование формализует процесс сравнения историй решений, и мы можем считать, что на историях решений введено отношение линейного предпорядка. Это отношение не может быть произвольным, оно имеет определенное количество свойств, некоторые из них универсальны для всех соревнований и тестирований, другие являются естественными только в определенных педагогических ситуациях. Например, в некоторых случаях более быстрая посылка хорошего решения обязательно является плюсом для участника.

СРАВНЕНИЕ ИСТОРИЙ РЕШЕНИЙ

В общем случае сравнение историй решений двух участников может происходить произвольным образом. Приведем примеры некоторых нетривиальных и не использующихся на практике способов сравнения историй решений. Будем считать, что множества решений сводятся к некоторым числовым множествам, то есть к баллам. В первом примере предположим, что для оценки истории решений выбирается наилучшее и наихудшее решение и считается отношение соответствующих им баллов. Получается, что для победы участнику необходимо не просто сдать самое хорошее решение, но при этом не отсылать дополнительных плохих решений.

Во втором примере предположим, что для сравнения историй у каждого решения вычисляется отношение его количества баллов к моменту времени, когда это решение было отослано, далее окончательная оценка истории решений выбирается или как первое, или как максимальное из полученных чисел. В данном примере оценивается «скорость зарабатывания баллов», то есть чтобы победить, можно заработать меньше баллов, чем другие, но послав при этом свои решения значительно раньше остальных.

При создании системы, позволяющей подводить итоги соревнований, необязательно поддерживать описанные только что регламенты. Иначе преподавателю для объективной оценки истории решений придется основываться на полной истории участников по каждой задаче. Значительно проще было бы оценивать истории решений только на основе лучших или последних решений и, возможно, времени, когда эти решения были посланы.

Чтобы оправдать тот факт, что в типичной педагогической ситуации из всей истории решений преподавателя интересуют только лучшие или последние решения, перечислим некоторые свойства отношений линейного предпорядка на произвольном множестве историй решений. Следствием этих свойств будут утверждения о независимости историй решений от тех решений, которые не являются ни последними, ни лучшими.

Далее мы считаем, что нам дана задача $P=(M,\leq)$, у которого множество историй посылок упорядоченно отношением предпорядка $\leq$. $H, H_1, H_2 \subset [0,1]\times M$ – истории решений участников. $a, b \in M, b \leq a$ – решения участников, $t, t_1, t_2 \in [0,1]$, $t_1 < t_2$ – моменты посылки решения.

- *Свойство согласованности порядков.* Если истории решений двух участников отличаются только одним из решений, то истории решений сравниваются так же, как и отличающиеся решения: $\{(t,b)\}\cup H \leq \{(t,a)\}\cup H$. Это свойство логично требовать всегда. Обратное свойство также логично требовать всегда.
- *Свойство лишних подходов.* Посылка плохого решения, а потом хорошего всегда не лучше, чем посылка только хорошего решения: $\{(t_1,b),(t_2,a)\}\cup H \leq \{(t_2,a)\}\cup H$. Это свойство разумно требовать в любом соревновании.
- *Свойство ухудшения решения.* Послать хорошее решение, а потом плохое всегда не лучше, чем если послать только хорошее: $\{(t_1,a),(t_2,b)\}\cup H \leq \{(t_2,a)\}\cup H$. Это свойство разумно требовать в любом соревновании.
- *Свойство улучшения решения.* Всегда лучше послать сначала плохое решение а потом хорошее, чем послать только плохое: $\{(t_1,b)\}\cup H \leq \{(t_1,b),(t_2,a)\cup H$. Это свойство кажется естественным, и в большинстве соревнований оно выполняется. Но не составляет труда придумать пример соревнования, в котором это не так. Рассмотрим, например, соревнование со следующим регламентом оценки решений: из истории решений участника выбирается первое лучшее

решение и вычисляется, на каком подходе оно было получено. Если у двух участников лучшие решения совпадают, то лучшим считается тот из участников, кто получил это решение за меньшее число подходов. В этом случае, если $H=(t_3,c)$, где $t_3>t_2$, $b\le c$, свойство выполняться не будет.

Таким образом, вроде бы логичное свойство регламента соревнования поощрять исправления решений, не всегда используется на практике. Но проблема возникает из-за того, что мы говорим о паре двух решений независимо от оставшегося множества подходов H. Свойство улучшения решения может быть переформулировано в более слабой форме, добавляющей требования на историю решения H, например, можно требовать, что решение a является лучшим из решений в истории, либо, что в истории решений H нет подходов после подхода (t_2,a).

• *Свойство лишних удачных подходов.* Допустим, участник послал сначала хорошее решение, а после него плохое. Часто можно считать, что это лучше, чем если бы он послал только плохое решение. Слабое свойство лишних удачных подходов: $\{(t_2,b)\}\cup H\le\{(t_1,a),(t_2,b)\}\cup H$, сильное свойство лишних удачных подходов: $\{(t_1,b)\}\cup H\le\{(t_1,a),(t_2,b)\}\cup H$. В соревнованиях это выполняется не всегда.

• *Сильная независимость от времени.* Сильная независимость от времени означает, что для любого подхода не имеет значения, когда он был послан:

$$\{(t_1,a)\}\cup H\sim\{(t_2,a)\}\cup H.$$

Это свойство выполняется далеко не всегда, с ним порядок удачных и неудачных подходов значения не имеет.

• *Слабая независимость от времени.* Слабая независимость от времени, в отличие от сильной, разрешает изменять время подхода, но только если между двумя моментами времени не совершалось других подходов: $\forall t\in[t_1,t_2],\, c\in M:(t,c)\notin H\Rightarrow$
$\Rightarrow\{(t_1,a)\}\cup H\sim\{(t_2,a)\}\cup H$.

• *Наихудший результат решения.* История, которая не содержит ни одной попытки, является не лучшей, чем любая другая история. Это свойство также выполняется не всегда, изредка наличие неправильного решения может добавлять ученику отрицательное число баллов.

Список свойств может быть продолжен. Приведенные в начале главы регламенты соревнований не удовлетворяют большинству приведенных свойств.

Повторим, что изучение свойств конкретного регламента позволяет сделать более эффективным сравнение результатов двух участников на основе их историй решений. В общем случае для сравнения результатов необходимо иметь полные истории решений каждого из участников, но такие истории часто содержат много лишней информации, которая бесполезна для сравнения участников и которая не может быть удобно отображена для восприятия членами жюри и, возможно, другими участниками, имеющими доступ к таблице результатов. Например, если регламент имеет сильную или слабую независимость от времени, жюри не имеет смысла предоставлять полную информацию о времени каждого подхода, достаточно отображать только множество решений, которые участник посылал на проверку. В более простых случаях, если регламент обладает всеми перечисленными свойствами, то все истории решений, имеющие одинаковые наилучшие и наихудшие решения, оказываются эквивалентными:

Теорема. Если отношение предпорядка на множестве историй решений обладает свойствами «сильной независимости от времени», «улучшения решения» и «лишних подходов», то две истории решений, имеющие одинаковые наихудшие и наилучшие решения являются эквивалентными. Пусть $b\le a\in M$, H – некоторая история решений, где все решения находятся в диапазоне от b до a: $\forall(t,c)\in H:b\le c\le a$. Тогда

$$\{(t_1,b),(t_2,a)\}\sim\{(t_1,b),(t_2,a)\}\cup H.$$

Доказательство. В связи с сильной независимостью от времени мы можем считать, что в истории $\{(t_1,b),(t_2,a)\}\cup H$ все решения упорядочены по возрастанию. Проверим, что если $t_1\le t_2\le t_3$ и $b\le c\le a\in M$, то $\{(t_1,b),(t_2,c),(t_3,a)\}\sim\{(t_1,b),(t_3,a)\}$. Для этого напишем следующее неравенство:

$$\{(t_1,b),(t_3,a)\}\le\{(t_1,b),(t_2,c),(t_3,a)\}\le\{(t_1,b),(t_3,a)\}$$

Первое неравенство следует из свойства улучшения решений для решений b и c, второе – из свойства лишних подходов для решений c и a. Так как первые и последние истории совпадают, вся цепочка состоит из эквивалентных историй. Случай одного или нуля решений должны быть рассмотрены отдельно, строго говоря, в условии теоремы они исключены тем фактом, что утверждение предполагает наличие двух решений a и b.

Теорема показывает, что для сравнения историй решений двух участников в соревновании с регламентом, удовлетворяющим свойствам перечисленным в теореме, достаточно знать только наилучшее и наихудшее решение участников. Другими словами, упорядоченное множество историй решений *сводится* ко множеству $M \times M$ пар наилучших и наихудших решений с некоторым заданным на нем предпорядком.

Отдельно обсудим сведение истории из одного или из нуля решений. Если в истории одно решение, то можно считать, что оно тоже сводится к элементу множества $M \times M$: $(\{(t_1, m)\}) = (m,m)$. Сведение будет корректно в том случае, если $\{(t_1, m)\} \sim \{(t_1, m), (t_2, m)\}$, потому что история из правой части эквивалентности имеет тот же результат сведения. Но эти две истории действительно эквивалентны, если учесть все свойства из условия теоремы. (Необходимо воспользоваться тем, что решение m, с одной стороны, не лучше, с другой, – не хуже того же самого решения m. История, состоящая из нуля решений, может сводиться к любому элементу множества. Если выполняются свойства наихудшего результата решения, то пустая история сводится к решению, состоящему из двух наихудших решений.

В соответствии со свойством согласованности порядков, если само упорядоченное множество M сводится к M', где, например, M' является числом баллов за задачу, то множество историй решений сводится ко множеству $M' \times M'$. В приведенном примере это множество пар баллов за лучшее и худшее решение. Элементы множества, к которому свелось множество историй решений, могут быть отображены в общедоступной таблице результатов, они объективно отражают решение участника, и на их основе участники могут сравнивать свои решения.

Изучение типичных свойств регламентов позволяет обнаружить множества, к которым могут быть сведены упорядоченные истории решений. Эти множества необходимы для упрощения, увеличения наглядности, автоматизации процесса сравнения историй решений.

Приведенная выше теорема имеет достаточно простое доказательство, но при этом является слабым результатом. На практике сильная независимость от времени встречается не всегда, кроме того, выводом является независимость результатов сравнения историй от их наихудших решений. Из практики известно, что наихудшие решения рассматриваются преподавателями редко. Далее мы будем считать без доказательства, что для сравнения историй решений во всех встречающихся на практике педагогических ситуациях достаточно рассматривать только самое лучшее и последнее решение и для каждого из них учитывать, возможно, время сдачи и количество попыток, с которыми решение было сдано. Другими словами, мы будем считать, что всегда существует сведение $f : \{(t_a, a), (t_b, b)\} \cup H \to \{(t_a, a, n_a), (t_b, b, n_b)\}$, где a – наилучшее решение, b – наихудшее, а n – это номер попытки, с которой это решение было получено.

СРАВНЕНИЕ РЕЗУЛЬТАТОВ УЧАСТНИКОВ

Помимо перехода от сравнения решений к сравнению историй решений, также необходимо рассматривать переход от историй решений к группе историй решений. Причина в том, что участники соревнования и тестирования решают не одну задачу, а несколько.

Под участником мы понимаем некоторое отображение, которое сопоставляет задачам историю решений. Другими словами, участник, имея задачу, решает ее, то есть посылает несколько решений на проверку. Обсудим модель участника подробней, после того как введем оставшиеся модели.

На практике для выполнения сравнений удобнее всего использовать числовые множества, по этой причине мы часто говорили о сведении множеств решений задач к числовым множествам, то есть о выставлении баллов за решения. Будем называть числовое множество *естественно* упорядоченным, потому что на нем есть отношение порядка, удобное для вычисления вручную или на компьютере. Кроме того будем называть естественно упорядоченным множеством многомерные числовые множества, например, упорядоченные пары или тройки чисел. Порядком на таких множествах будет лексикографический порядок, он тоже удобен для ручных или компьютерных вычислений. Примером использования многомерного множества может являться сведение истории решений к паре из количества баллов за задачу и времени, когда это количество баллов было получено. При одинаковом количестве баллов лучшим будет считаться тот участник, который получил эти баллы раньше.

Модель учителя определяет процедуры проверки и сравнения решений участников для формирования результатов соревнования. Модель учителя состоит из:

1. *Функций, сводящих решения задач к естественным упорядоченным множествам.* Сведение к одномерному множеству соответствует *однопараметрической* проверке решений, к многомерному – *многопараметрической* проверке. Именно это сведение задает предпорядок на множестве решений каждой задачи соревнования, таким образом, задачи соревнования частично определяются выбранной моделью учителя (напомним, что каждая задача соревнования – это множество решений вместе с заданным на нем линейным предпорядком). Этот аспект модели учителя соответствует тому, что учитель проверяет и оценивает решения учеников. Каждому решению он сопоставляет числа или несколько чисел, например баллы.

2. *Функций, сводящих истории проверки решений к естественно упорядоченному множеству.* Эти функции задают предпорядок на историях решений каждой задачи, следовательно, позволяют сравнить нескольких участников по результатам решения отдельных задач. Понятие истории проверки решений аналогично понятию истории решений, но в качестве множества, элементы которого располагаются во времени, выступает множество результатов проверки, а не множество самих решений. Другими словами, будем считать, что при сравнения историй решений мы будем рассматривать не сами решения участников, а уже проверенные решения, то есть сведенные к некоторым естественно упорядоченным множествам. Такое допущение возможно в связи с принципом согласованности порядков. Естественно упорядоченное множество, к которому сводится множество историй проверок, будем называть множеством результатов решения задачи. Этот аспект модели учителя соответствует тому, что учитель на основе нескольких попыток ученика сдать задачу формирует окончательный результат ученика по этой задаче.

3. *Функций, сводящих множество участников к естественно упорядоченному множеству.* Элементы этого множества являются результатами участия в соревновании. Порядок на множестве задает порядок участников в турнирной таблице, то есть определяет призеров, победителей, полученную участником оценку, информацию о прохождении зачета. Этот аспект модели учителя соответствует тому, как учитель окончательно подводит итоги работы ученика на основе всех решенных задач.

Понятие учителя соответствует объективному учителю, проверяющему решения задач участников и подводящему итоги соревнования. В реальной ситуации под учителем можно понимать одного человека или коллегию из членов жюри. Как видно, модель учителя содержит значительное количество функций, то есть сложна для описания в каждой конкретной ситуации. Мы предложим упрощенный способ описания модели учителя при обсуждении системы для автоматизации процесса подведения итогов.

Результатом обработки является таблица результатов. Она строится на основе множества участников и множества задач соревнования. Таблица сопоставляет каждой паре из ученика и задачи результат решения задачи учеником. Кроме того, каждому ученику таблица сопоставляет результат его участия в соревновании, подведенный на основе результатов решения отдельных задач.

Приведенная выше теория линейных предпорядков на множестве решений и множестве историй решений, в первую очередь, необходима для автоматизации процесса проведения соревнований, потому что она позволяет описать регламенты широкого класса соревнований. Кроме того, приведенная теория лежит в основе метода выбора модели учителя перед началом соревнования, если такая модель еще не устоялась исторически для проводящегося соревнования. Кратко метод включает следующие шаги:

1. Описать несколько «типичных» учеников, которые, как предполагается, проявятся в соревновании. Ученик соответствует истории решений, поэтому описание типичного ученика представляет собой характерную историю решений, например, посылку постоянно улучшающихся решений задачи или, наоборот, случайно чередующихся по качеству решений.

2. Из списка свойств линейных предпорядков выбрать те, которые соответствуют педагогической ситуации, например, определить, требуется ли наказывать ученика за посылку плохого решения, если перед этим им уже было послано хорошее.

3. На основе выбранных свойств и теорем, одна из которых приведена и доказана в этой статье, определить множество, к которому сводится история решений задач. Приведенная теорема, указывает, что в некоторых случаях всю историю решений можно свести только к паре из двух решений – лучшему и худшему.

СИСТЕМА ПОДВЕДЕНИЯ И ВИЗУАЛИЗАЦИИ ИТОГОВ СОРЕВНОВАНИЙ

Опишем концепцию системы, которая позволяет сформировать таблицу результатов соревнования на основе описанного преподавателем регламента соревнования и введенной им истории подходов учащихся. Описание регламента естественно делать на некотором предметно ориентированном языке (ПОЯ), поэтому мы будем описывать только сам предлагаемый для системы ПОЯ и не будем говорить ни о каких других особенностях ее интерфейса [7]. Тот факт, что система основана на ПОЯ означает, что ее можно реализовать с помощью инструментов создания редакторов ПОЯ. Одной из таких систем является система MPS фирмы JetBrains. Она позволит по описанию языка создать специализированный редактор, предоставляющий все современные возможности редакторов, такие как подсветка синтаксиса и автодополнение.

Опишем сценарий работы преподавателя с системой. Перед началом соревнования (учебного семестра) преподаватель описывает модель соревнования, в течение соревнования (учебного семестра) он заполняет историю подходов участников. В каждый момент времени на основе модели и истории подходов система генерирует таблицу результатов, которая используется преподавателем для оценки успеваемости и учениками для слежения за своими успехами и долгами.

Процесс описания соревнования мы снабдим примером. Допустим, учебный процесс в течение семестра устроен следующим образом: студенты решают задачи и сдают их на проверку преподавателю. Если студент решил задачу неправильно, это никаким образом не влияет на его результаты, если он решил правильно, преподаватель решение засчитывает. Для сдачи зачета необходимо в течение семестра решить хотя бы 10 задач. Если студент не решил 10 задач вовремя, ему требуется решить уже 15 задач.

Для описания соревнования преподавателю необходимо задать параметры модели соревнования. Для этого он заполняет следующие данные:

1) список учеников,
2) список типов задач,
3) список задач,
4) описание сведения результатов проверок к результатам участия,
5) историю подходов участников.

Типы задач не являются частью модели соревнования, они необходимы для упрощения процесса описания задач. Вместо того чтобы описывать, каким образом каждая задача обрабатывается при подведении итогов, можно назначить каждой задаче ее тип, а далее описать способы обработки каждого из типов. Некоторые типы задач могут быть встроены в системы, некоторые задаваться преподавателем. Например, очень популярным способом оценки задач является назначение баллов за задачу, при этом время сдачи не учитывается, а из нескольких попыток сдачи выбирается наилучшая. Такой тип можно предоставлять преподавателям без необходимости описывать его в системе.

Наиболее просто в системе может быть описан список учеников, для этого надо перечислить их имена. Далее, при наборе истории подходов участники будут выбираться из этого заранее заданного списка. В нашем примере список мог бы выглядеть так (не будем придумывать имена):

```
Студенты:
Студент 1, Студент 2, Студент 3
```

Перейдем к процессу описания типов задач. Сразу приведем ту информацию, которую необходимо задать для описания типа, после обсудим ее подробнее:

1) описание множества решений задачи (необязательно),

2) метки для описания множества результатов проверки,

3) сведение истории результатов проверки к результатам решения задачи.

Каждой задаче, в соответствии с моделью задачи, соответствует множество возможных ответов. Для описания множества достаточно задать тип его элементов – числа, строки, комбинированный тип, то есть, например, два поля, первое является строкой, второе числом. Для задания псевдопорядка на множестве ответов достаточно указать некоторое естественно упорядоченное множество, к которому сводится множество ответов. Само сведение указывать не нужно, потому что процесс проверки и оценки решения является прерогативой преподавателя.

Напомним, что множествами с естественным порядком мы считаем числовые множества и множества, являющиеся декартовыми произведениями нескольких числовых множеств. Во втором случае для сравнения используется лексикографический порядок. Задавать множества в системе мы будем с помощью меток. Пример одномерного множества – «баллы», пример двумерного множества «баллы, штрафные баллы». Метки необходимы, для того чтобы придать смысл элементам множества. Кроме того, если считать метки заранее заданными в рамках системы, то метка может задавать и направление сравнения чисел. Например, для лучшего результата баллов должно быть больше, а штрафных баллов меньше. Аналогично метка может задавать и само числовое множество. Например, метка «зачтено» может соответствовать множеству $\{0, 1\}$, а «баллы» – целым неотрицательным числам.

Для целей подведения итогов само множество возможных решений задачи принципиального значения не имеет. При вводе истории подходов преподавателю достаточно вводить только результаты проверки. Сами решения участников нужны только для хранения истории и в рамках системы эти решения не анализируются.

Итак, при описании типа задачи первым делом необходимо указать одну или несколько меток. Эти метки описывают множество результатов проверки решения.

Далее необходимо описать сведение истории проверок к результату решения задачи. Результат решения задачи – это естественно упорядоченное множество, поэтому задать его опять необходимо с помощью меток. Как было указано в предыдущем разделе, для описания сведения достаточно иметь только информацию о лучшем и последнем решении, а для этих решений иметь только данные о моменте их сдачи и о количестве попыток, которые были до этого совершены. В разбираемом нами примере об увеличении количества задач, если они не сданы к сроку, будем считать, что результатом решения задачи является двумерное множество, первый элемент пары – это сдана ли задача вообще (то есть числовое множество со значением 0 или 1), второй элемент пары – вовремя ли сдана задача (1 – вовремя, 0 – не вовремя). Окончательно, тип задачи на предметно-ориентированном языке может быть описан примерно следующим образом:

```
тип Зачтено_и_опоздание
   проверка: зачтено
   результат: зачтено, опоздание
   сведение:
      зачтено = лучшее.зачтено
      опоздание = if лучшее.время >
                  20.12.2011 then 0 else 1
```

Строка «проверка» описывает множество результатов проверки перечислением меток. В данном случае метка перечисляется только одна, значит, при вводе данных преподаватель будет указывать только, зачел он студенту задачу или нет. Строки «результат» и «сведение» описывают формирование результата решения задачи. Результат будет состоять из двух чисел, помеченных метками «зачтено» и «опоздание», в первой указывается, зачтена ли задача, для этого выбирается лучшее решение и проверяется, зачтено оно или нет. Для проверки опоздания проверяется дата сдачи лучшего решения. В данном примере при выборе лучшего решения они будут сравниваться по значению «зачтено» в случае нескольких одинаковых результатов с лучшим значением параметра «зачтено», выбран будет первый по времени параметр. На практике история по каждой задаче будет либо пустой, либо состоять из одного подхода, потому что подходы с результатом «не зачтено» никаким образом не влияют на окончательный результат решения задачи, и преподаватель не будет тратить время на ввод информации обо всех подходах, а будет заносить только один результативный.

Отдельно стоит сказать о случае, когда история решения задачи пуста. В этом случае описанный выше результат сведения не определен. Необходимо либо добавить «значения по умолчанию» для результатов сведения, либо считать, что в случае пустой истории сведение выбирает минимальный элемент множества результатов проверки, что соответствует выполнению принципа наихудшего результата решения.

Следующим пунктом необходимо описать задачи соревнования. Здесь достаточно перечислить названия задач вместе с их типами:

```
Задачи:
Задача 1, Задача 2, Задача 3 :
                         Зачтено_и_опоздание
```

Обсудим сведение набора результатов решения задачи к результатам участия в соревновании. Результатом участия опять же должно являться естественно упорядоченное множество, чтобы участников можно было отсортировать по достигнутым ими результатам. В случаях, когда результатом подведения итогов соревнования необходимо определить, получил ли учащийся зачет, можно считать, что множество результатов участия состоит из двух элементов 0 и 1. В нашем случае зачет будет получен в случае, если у ученика есть 15 зачтенных задач, либо если есть 10 задач, зачтенных вовремя. Приведем возможный пример описания этих данных на предметно-ориентированном языке:

```
Результаты:
   результат: зачтено
   сведение:
      всего = sum(зачтено);
      вовремя = sum(зачтено * вовремя);
      зачтено = if всего >= 15 ||
                вовремя >= 10 then 1 else 0
```

В данном примере указано, что результат участия в соревновании является одномерным числовым множеством, помеченным меткой «зачтено». Для вычисления значения необходимо отдельно посчитать общее количество сданных задач и количество задач, сданных вовремя. Общее количество задач – это сумма значений «зачтено» каждой из задач соревнования. Задачи, сданные вовремя, получаются как сумма значений «зачтено*вовремя».

Все указанные выше данные вводятся преподавателем перед началом соревнования или учебного семестра и, возможно, незначительно меняются впоследствии. Данные для истории подходов участников вводятся только в течение соревнования или учебного семестра и выглядят примерно так:

```
Студент 1, Задача 1 -> зачтено:1,
                       время: 01.10.2011
Студент 2, Задача 3 -> зачтено:0,
                       время: 02.10.2011
Студент 2, Задача 3 -> зачтено:1,
                       время: 05.10.2011
```

Ввод истории при наличии автоматического дополнения, которое будет доступно, например, при использовании системы MPS, может происходить очень быстро с помощью выбора студентов и задач из списка.

Автоматически можно устанавливать и дату подхода. Язык можно сократить дальше, например, метка «зачтено» может обрабатываться редактором нестандартным способом, вместо ввода «зачтено:1» или «зачтено:0» он будет предлагать написать символ + или –. Статья [1] предлагает некоторые другие возможности упростить описание соревнования, для простых и часто используемых типов регламентов.

Литература

1. *Станкевич А.С.* Общий подход к подведению итогов соревнований по программированию // Компьютерные инструменты в образовании №2, 2011. С. 27–38.

2. *Рукшин С.Е.* Управление процессами проверки решений и подведения итогов в соревнованиях и тестированиях с автоматизированной проверкой решений // Компьютерные инструменты в образовании, 2010. № 4. С. 39–49.

3. *Посов И.А., Рукшин С.Е.* Архитектура системы проведения удаленных соревнований и организации работы с математическими задачами // Научно–технические ведомости СПбГПУ, 2010. № 3(101). С. 49–53.

4. *Рукшин С.Е.* Классификация типов научных соревнований с автоматической обработкой решений // Научно–технический вестник СПБГУ ИТМО, 2010. № 3(67). С. 121–125.

5. *Рукшин С.Е.* Сравнительные достоинства и недостатки дистанционных и традиционных олимпиад и их влияние на архитектуру автоматизированных систем поддержки дистанционных научных соревнований // Международный журнал Образовательные технологии и общество, 2010. Т. 13, № 3. С. 347–359.

6. *Рукшин С.Е.* Основы конструирования систем проведения дистанционных научных соревнований // Программные продукты и системы», 2010. № 3(91). С. 69–72.

7. *Фаулер М.* Языковой инструментарий: новая жизнь языков предметной области // http://www.maxkir.com/sd/languageWorkbenches.html. [Электронный ресурс] / MAXKIR.com. Самое интересное о разработке программного обеспечения. Электрон. дан. 27.12.2005. Режим доступа : *http://www.maxkir.com/sd/languageWorkbenches.html* свободный. Загл. с экрана. Яз. рус. (последнее обращение: 06.10.2010).

Abstract

The paper presents evaluation model of students activity in solving problems. The model can be used to design systems, that automatize the process of evaluation. Also the model can be used to design the training activity itself, because it generalizes some principles that are used by teachers to assess the students progress. The paper also presents a concept of a simplified system, that automatizes the process of summing up the solutions history that was entered by hand.

Keywords: Education, Distance Competitive Educational System, contest model, results summarization

Посов Илья Александрович,
ассистент кафедры ВМ–2 СПбГЭТУ «ЛЭТИ»,
iposov@gmail.com,

Рукшин Сергей Евгеньевич,
доцент кафедры математического анализа РГПУ им. Герцена,

vliuser@gmail.com