

АВТОМАТИЗАЦИЯ РАБОТЫ С КОНСТРУКТИВНЫМИ ЗАДАЧАМИ НА ПРИМЕРЕ РАЗРАБОТКИ МОДУЛЯ ВВОДА ГРАФИКОВ ФУНКЦИЙ В СИСТЕМЕ ПОДДЕРЖКИ ДИСТАНЦИОННЫХ НАУЧНЫХ СОРЕВНОВАНИЙ

Аннотация

Статья посвящена разработке программной поддержки важного для развития математического мышления типа математических задач, условием которых являются свойства неизвестного объекта, а решение состоит в построении объекта с заданными свойствами. Условия задач задаются набором предикатов, характеризующих правильный ответ. В предикатах могут использоваться стандартные математические операции и числовые функции, логические связки, кванторы всеобщности и существования. Разработанная технология демонстрируется на задачах, связанных с построением графиков функций. Для решения поставленной методической задачи рассматривается упрощенный вариант ввода графика: непрерывные его части вводятся «по точкам» и интерпретируются как кусочно-линейные функции. Для ввода асимптотического поведения используется ввод точек графика в специально выделенные области окна с системой координат, что является формализацией условностей, применяемых при рисовании графиков «от руки».

Ключевые слова: электронное обучение, верификация утверждений, предикаты, визуализация определений предела, ввод графиков.

ВВЕДЕНИЕ

Многим известна замечательная книга Б. Гелбаума и Дж. Олмстеда «Контрпримеры в анализе» [1]. В этой книге собраны примеры конструирования различных функций, благодаря которым читателю становится ясно, чем один класс функций отличается от другого. Такое знание принципиально важно для глубокого овладения материалом курса математического анализа, поскольку простые примеры, приводимые в учебниках, обычно принадлежат нескольким различным классам. Это не позволяет обучаемому полноценно использовать такие примеры как типовые образцы изучаемых классов для обдумывания задач и предвидения результатов. Без умения конструировать контрпримеры

обучаемый будет испытывать затруднения при осуществлении мысленных экспериментов с функциями различных классов.

На наш взгляд, умение конструировать примеры является важной частью понимания не только вузовского, но и школьного материала и позволяет увязать формальное представление доказательства с его психологическим восприятием. Умение быстро конструировать примеры функций с заданными свойствами может служить основой для самостоятельной формулировки и вывода общих теорем.

Обратим внимание ещё на одну проблему, истоки которой находятся в школе, но в полной мере проявляются при изучении математического анализа в вузе.

Основой вузовского курса математического анализа является понятие беско-

нечно малых и бесконечно больших величин. Традиционная нотация определенных, описывающих эти и связанные с ними понятия, опирается на язык «эпсилон-дельта», который впервые появляется при изучении этих понятий, а повторно встречается уже при изучении исчисления предикатов в курсе математической логики.

Одновременное появление этих двух трудностей – языка предикатов и понятия бесконечно малых – сразу приводит к появлению множества студентов, не понимающих математический анализ и начинающих воспринимать определения формально.

Между тем, на наш взгляд, эта проблема относится к числу тех же проблем, что и трудности изучения геометрии на основе теории геометрических преобразований, обнаруженная в 80-х годах прошлого века при введении в школах новой программы изучения математики, которая получила название программы Колмогорова. А именно, причина проблем кроется в отсутствии практического опыта работы с наглядными инструментальными моделями, адекватно отражающими сущность математических понятий. Поэтому, если перед изучением общих понятий обучаемый получит инструмент для работы с языком предикатов и инструмент для работы с определениями на языке предикатов и самостоятельно построит и исследует описанные им самим объекты, то определения из декларативных утверждений превратятся в инструмент практических построений, экспериментов и построения суждений.

Для реализации данного класса задач на уровне школы нами предлагается система, которая поддерживает составление и решение задач на построение графиков функций по их свойствам. Для описания свойств используется язык исчисления предикатов. Конструирование функции предполагает поточечный ввод графика кусочно-линейной функции и использование специальных приемов для указания асимптотического поведения функции. Использование данной системы позволяет конструировать функции с различны-

ми сочетаниями расположений нулей функции, промежутков знакопостоянства, экстремумов, промежутков возрастания, разрывов, промежутков выпуклости и вогнутости.

В настоящее время существует множество систем, предполагающих графическое отображение информации, однако все они основаны на одной из двух стратегий:

- табличный ввод данных,
- задание функций формулой.

Например, все системы поддержки символьных вычислений (Maple, Mathematica, Maxima, MathCad и др.) позволяют изображать график функции, если известна формула или уравнение, задающие функцию [2].

Системы, специально сделанные для использования в школе, например «Живая математика» или «Автограф», также позволяют изображать графики явно или параметрически заданных функций, строить графики по точкам, соединяя точки графика отрезками прямых («программа «Автограф» позволяет также изображать графики некоторого класса неявно заданных функций). В то же время ни одна из известных систем не предполагает ввод графика от руки, с тем чтобы рассматривать его как объект анализа.

Определенные шаги в этом направлении были сделаны в программных реализациях механизма «анкета функции», которые нашли своё отражение в программных продуктах [3, 4]. Однако в них, вместо графического ввода, рассматривался интерфейс заполнения «анкеты» функции, в которой указывались основные свойства, определяющие вид графика. В качестве обратной связи ученику демонстрировался предполагаемый график, и, если этот график не совпадал с ожидаемым, ученик мог перейти к исправлению ошибок в анкете функции. Одной из проблем в реализации этой технологии является потенциальное наличие противоречий (в случае, если анкета заполнена неправильно), на которые программа должна реагировать. В указанных выше продуктах эта задача решалась по-разному: в про-

грамме «Лабораторные работы по математике» противоречивые свойства отбрасывались, в программе «Математика. Практикум: 5–11 классы» при наличии противоречий в анкете график не строился совсем.

Реализация ввода графика «от руки», предпринятая в данной работе, предполагает дополнить «анкетный» способ задания функции, когда все составляющие «анкеты» функции можно задать графически. С другой стороны, программный продукт позволит формулировать новые задачи, предполагающие не единственный правильный ответ, причём задача не требует специального нахождения «правильного» ответа методистом, поскольку проверяться будут отдельные (указанные в условии) свойства графика функции.

МОДЕЛИРОВАНИЕ ВВОДА ГРАФИКА ЧЕРЕЗ ПОСТРОЕНИЕ «ОТ РУКИ»

При рисовании графика на листе бумаги рука ученика может скользнуть в сторону от проводимой линии. В этом случае ученик проводит по этому месту повторно, пока жирно не выделит нужную ему линию. При этом, разумеется, он не стремится изобразить график точно, он изображает лишь тенденции: «здесь график возрастает, здесь у него максимум, здесь график убывает» и т.д. Этим, в частности, объясняются те «ошибки», которые часто обнаруживаются у школьников при построении стандартных графиков: то парабола оказывается «заостренной» в вершине, то график синуса пересекает ось абсцисс под прямым углом. На наш взгляд, считать их ошибками можно, только если ученик изображает свойства *дифференцируемой* функции, то есть учитель проверяет то, насколько хорошо ученик владеет понятиями дифференциального исчисления и может их применять к построению графиков функций. Если же ученик хочет обратить внимание на точку экстремума (как в первом случае) или промежутки возрастания-убывания (во втором), то особенности поведения касательной могут не приниматься во внимание, так как *выб-*

ранная учеником графическая модель этих параметров не учитывает.

Таким образом, создавая модуль ввода графика, мы должны суметь «поймать» только те свойства, которые ученик хочет продемонстрировать. Например, если при движении «мышкой» рука дрожит и вместо монотонной функции появляется функция с большим числом локальных экстремумов, это явно не то, что программа должна «увидеть».

В данном случае мы считаем правильным соединить «определения» тех или иных свойств функции с условными представлениями их графиками так, чтобы, строя график, ученик однозначно передавал через графический интерфейс эту информацию. Для этого мы на первом этапе отказались от идеи «плавного» рисования графика, сохранив возможность ученику точно передавать свои намерения о поведении графика такого вида: «сейчас график должен идти вверх, далее у него максимум, после точки максимума линия графика должна пойти вниз» и т.д. Программа предоставляет ученику возможность рисовать график кусочно-линейной функции, щёлкая мышкой там, где ему нужно изменить характер поведения функции. При этом естественным образом визуализируются определения возрастания, убывания, экстремума, выпуклости, вогнутости и перегиба функции, считая, что экстремумы могут быть только в отмечаемых точках, а перегибы на отрезках, которые являются одновременно концами участков выпуклости одного знака и началом участков выпуклости противоположного знака (или в некоторых случаях образовывать промежутки с концами в отмеченных точках).

Обратим внимание ещё на такие особенности суждений ученика, которые не будет учитывать первоначально предлагаемый интерфейс (но которые учитываются в версии ввода «со сглаживанием»). Эти особенности отчасти объясняются традициями, а отчасти жизненным опытом, например, «около максимума график должен плавно закругляться». Разумеется «зак-

руглять» график нужно, только если мы говорим о функциях определенного класса, то же можно сказать о выпуклости. Однако, чем больше дискретная математика будет входить в курс традиционного обучения математике, тем более естественными будут становиться функции с «ломаными» графиками. В то же время в системе присутствует механизм аппроксимации (его обсуждение выходит за рамки данной статьи), который позволяет сделать графики настолько «плавными», насколько это делается от руки с сохранением положений важных точек, расположение которых может быть частью условия задачи.

КООРДИНАТЫ И ПАРАМЕТРЫ. ПРЕДИКАТЫ НА МНОЖЕСТВЕ ВВЕДЕННЫХ ТОЧЕК

В качестве основных объектов для интерфейса будут использоваться два вида объектов: точки и параметры. Точки – это пары чисел. Абсцисса и ордината точек могут использоваться как независимо, так и в указанном соответствии, в последнем случае будет использоваться индексация точек. Целесообразно рассматривать привязку точек к узлам сетки, тогда работать с координатами точек можно как с целыми числами.

Программа имеет три режима отрисовки графика:

1) график последовательности; в этом случае изображаются только точки и тенденции на бесконечности;

2) график кусочно-линейной функции; в этом случае график представляют собой несколько ломаных;

3) сглаженный график кусочно-линейной функции, на котором можно также продемонстрировать различные разрывы, асимптоты и предельное поведение.

Проекция построенного графика на ось абсцисс рассматривается как область определения вводимой функции, а проекция на ось ординат – как область значений. Точки графика, поставленные за крайними линиями экрана, связанными с асимптотикой функции, интерпретируются ина-

че, что будет обсуждено отдельно. Появление точек в этих областях означает продолжение областей определения и/или значений на соответствующие полуоси.

Свойства функций описываются часто используемыми в анализе логическими формулами с кванторами.

Например, формула $\forall x (f(x) > 0)$ соответствует высказыванию «график функции лежит выше оси абсцисс», а формула $\forall y (y > 0)$ соответствует высказыванию «все значения функции положительны». В данном случае оба высказывания эквивалентны.

Запись $\forall x (f(x) > 0 \rightarrow x > 0)$ формализует высказывание «функция может быть положительна только при положительных значениях аргумента». При сделанных замечаниях это может быть также формализовано, как $\forall y_1 (y_1 > 0 \rightarrow x_1 > 0)$.

Утверждения при их формализации могут содержать параметры, связанные кванторами. Например, утверждение об ограниченности функции сверху $\exists N (\forall x (f(x) < N))$ содержит параметр N . Кроме этого, параметры могут быть частью условия. Например, запись условия «график квадратичной функции $y = ax^2 + bx + c$ не имеет точек внутри третьей четверти» в пропозициональной форме будет иметь вид

$$\forall x (x < 0 \rightarrow ax^2 + bx + c \geq 0),$$

то есть будет предикатом от трех параметров a , b и c .

Проверка утверждений с параметрами, как правило, предоставляет большие сложности, чем проверка утверждений от конечного числа координат точек (исключение составляют случаи, когда множество значений параметра дискретно). Наиболее типичен случай, когда параметры являются вещественными числами. В этом случае нужно либо использовать теорему Тарского, что может привести к значительным вычислительным трудностям, либо остановиться на варианте выборочной проверки утверждения. Как правило, последний подход даёт хорошие результаты, если пользователь стремится решить задачу, а не исследовать границы применения выбранного подхода.

ОПИСАНИЕ СВОЙСТВ ГРАФИКОВ ПРЕДИКАТАМИ

Возможность сопоставить свойствам формальные описания позволяет использовать эти свойства в качестве элементов конструируемых ответов. Описанный механизм реализован в программе Verifier [6, 7]. Рассмотрим эти возможности на примерах.

Пример 1. При задании графика последовательностью точек между соседними точками возникают естественные отношения соседства, которые можно задать отношением между абсциссами точек $L(x_a; x_b)$, которая означает, что точка A является непосредственным левым соседом для точки B . Тогда формула $\exists x(L(x_a; x))$ означает, что точка A не является крайней справа.

Пример 2. Возрастание функции можно описать с предикатом

$$E(u; v) = (u > v \rightarrow f(u) > f(v)).$$

Тогда утверждение «функция является возрастающей» будет выглядеть следующим образом: $\forall u(\forall v(E(u; v)))$. Соседство точек будет описано более сложным предикатом:

$$S(u; v) = \neg(\exists w(u < w < v \vee v < w < u)).$$

Можно упростить его, учтя порядок соседей: $SL(u; v) = \neg(\exists w(u < w < v))$. После этого можно ввести предикат, соответствующий наличию строгого максимума в точке w :

$$MAX(w) = \exists u(\exists v(SL(u; w) \wedge SL(v; w) \wedge f(u) < f(w) \wedge f(v) < f(w))).$$

Тогда утверждение о том, что функция имеет строгий максимум в одной из промежуточных точек (которые соответствуют внутренним точкам непрерывного графика) запишется так: $\exists w(MAX(w))$.

Заметим, что преимущество поточечного задания графика является возможность формулировать и быстро проверять свойства заданной функции, так проверка осуществляется простым перебором точек.

Приведённые примеры демонстрируют несколько важных закономерностей:

– новые понятия и суждения можно формулировать как в формальном виде,

так и в словесном, сопоставляя взаимнооднозначно предложения русского языка (с переменными) формулам исчисления предикатов;

– язык описания можно расширять, добавляя новые определения, опирающиеся на определения, сформулированные ранее.

Заметим, что такой подход имеет несколько педагогических следствий:

1) возможно такое использование предложенного инструмента, когда ученик и компьютер меняются местами, и фактически ученик учит компьютер понятиям математического анализа, последовательно вводя новые предикаты как свойства изучаемых понятий (данный подход последовательно рассматривается в работах С.Г. Иванова [5]);

2) благодаря возможности верифицировать вводимые утверждения, ученики могут постоянно использовать учебный эксперимент, который является основой для формирования веры учеников в свои силы; различные возможности использования механизма верификации для поддержки задач математического анализа изложены в статье одного из авторов данной работы [6–8];

3) благодаря визуализации, формулируемые свойства можно увидеть на графиках функций;

4) использование школьного материала для формулировки понятий и суждений позволяет применить полученные результаты для новых форм обучения по школьной программе.

Перечисленные соображения демонстрируют возможности, которые предоставляет данный инструмент для расширения класса задач дистанционных математических олимпиад.

ОБОЛОЧКА «УЧЕНИКА» И ОБОЛОЧКА «УЧИТЕЛЯ»

Оболочки, упомянутые в названии, – дань традиции: учитель записывает условие задачи в форме предиката, а ученик строит график функции, удовлетворяющий

этим свойствам. Однако нетрудно и обратить цель работы: ученик может сам составлять предикаты, предполагая, что они описывают некоторые классы функций, а затем экспериментировать со своим определением.

Рассмотрим функции этих двух составляющих системы:

1) **Подсистема для ввода базовых предикатов и задания их словесных описаний** позволяет каждую задачу ассоциировать с некоторым высказыванием, например, высказывание $\exists w(MAX(w))$ соответствует высказыванию «функция имеет максимум», которое естественным образом трансформируется в задачу «Постройте функцию, имеющую максимум», а высказывание $\exists w(MAX(w) \wedge w > 0)$ аналогично соответствует задаче «Постройте функцию, имеющую максимум при положительном значении аргумента». Можно предложить ученикам самим составлять формально-логические описания задач, предлагая словесные формулировки, например «График функции может иметь максимум только в правой полуплоскости». Ответ: $\forall w(MAX(w) \rightarrow w > 0)$.

Замечание. Полезным будет сравнить приведенные две задачи, чтобы, конструируя графики, найти примеры, на которых высказывания дают разные ответы.

2) **Подсистема для ввода графиков функций.** В этой системе имеется возможность рисовать графики, которые верифицируются на различных суждениях. В стандартном варианте это задача «Постройте график функции, обладающей следующими свойствами».

Однако представляют интерес и другие педагогические варианты использования возможностей созданной системы. Например, «чёрный ящик». В программе спрятано некоторое свойство графика, которое ученик должен «открыть». Для этого ученик делает эксперименты, вводя разные графики и наблюдая за отличием реакции на спрятанное свойство от реакции на свойство, которое он конструирует как решение задачи. Здесь возможны варианты с сохранением введённых гра-

фиков, генерированием случайного графика, проверкой на некотором заданном множестве графиков с разнообразными свойствами (базовом множестве).

ПОСЛЕДОВАТЕЛЬНОСТИ И КУСОЧНО-ЛИНЕЙНЫЕ ФУНКЦИИ

Заметим, что поточечный ввод графика является естественным для описания следующих классов функций:

1) для функций натурального аргумента или последовательности; заметим, что для описания последовательности очень часто применяется индуктивное описание, когда последовательность задаётся несколькими первыми членами (на этой идее основана Интернет-энциклопедия целочисленных последовательностей [9], в которой описано более 170 тысяч известных последовательностей, а сама последовательность находится по предъявленным нескольким последовательным её членам);

2) для кусочно-линейных функций (поведение графика за пределами изображаемой области определяется поведением его крайних участков – лучей);

3) для описания функций «линейных в малом», то есть для кусочно-дифференцируемых функций, которые включают все изучаемые в школе классы функций; в этом случае график строящейся кусочно-линейной функции рассматривается как аппроксимация исходной функции, а после построения осуществляется операция сглаживания, которая производится таким образом, чтобы «уловить» те сочетания точек, которыми человек, вводящий график, хотел передать непрерывные его изменения, и обратно, «уловить» места разрывов, изломов и пр.

В первом случае свойства функции проверяются для членов последовательности, изображенных на экране. Для этого конечного множества точек проверяемые свойства будут определяться точно. Как правило, приводимый отрезок последовательности берётся таким, чтобы представить все её свойства до момента стабилизации. За пределами начального отрезка

важны асимптотические свойства изображаемой последовательности, представление которых описано в другом разделе статьи.

Во втором случае функция может быть описана точно с помощью набора отрезков и не более чем двух лучей, которыми могут быть крайние звенья графика кусочно-линейной функции. При отрисовке этих функций используется знак «стрелочка» для указания того, что граничная точка не входит в звено графика. Также этот инструмент удобен для изображения лучей.

В третьем случае график кусочно-линейной функции рассматривается как аппроксимация графика функции, при этом часто важны не столько точные значения функции, сколько её тенденции на том или ином участке: непрерывность и разрывы, монотонность и экстремумы, нули и интервалы знакопостоянства, выпуклость-вогнутость и наличие точек перегиба. Кроме того, при изучении анализа большую роль играет асимптотика функций как на бесконечности, так и в окрестности точек разрыва. Для описания поведения функции в этих случаях используются специальные средства, составляющие важную часть данной статьи.

СРЕДСТВА ВИЗУАЛИЗАЦИИ

Интерфейс изображения графика устроен так, чтобы вводимое множество точек определяло график функции. Для этого блокируется возможность построить очередную точку так, чтобы она лежала левее или была на той же вертикальной прямой, что и текущая. Двойной щелчок означает разрыв функции (такую же роль играет щелчок правой кнопкой при построении очередной точки). На рисунке в этом случае будет отсутствовать отрезок прямой, соединяющий последовательные точки.

Для изображения асимптот используются специально проведенные вертикальные и горизонтальные линии, попадание точек за которые интерпретируется специальным образом.

Изображению асимптотического поведения функций посвящен следующий пункт (рис. 1).

Например, функция, введенная через график, который показан на рис. 1, автоматически интерпретируется как

- 1) определенная для всех действительных чисел;
- 2) имеющая два нуля, один из которых отрицательный, другой положительный (имеется возможность указать точные значения корней или их свойства, например, указав, что положительный корень по абсолютной величине больше отрицательного), которые определяют промежутки знакопостоянства;
- 3) имеющая минимум, до которого функция монотонно убывает, а после которого монотонно возрастает;
- 4) два участка выпуклости вверх и один участок выпуклости вниз (вогнутости);
- 5) кроме того, на минус бесконечности функция ограничена, а на плюс бесконечности стремится к плюс бесконечности.

Приведём (неполный) перечень условий, на основе которых построен алгоритм сглаживания. В данной статье этот алгоритм подробно не обсуждается.

Сглаживанием мы будем называть замену кусочно-линейной функции дифференцируемой так, чтобы смоделировать сглаживание, которое делает ученик при рисовании рукой. При сглаживании графика на промежутке, где функция непрерывна, будут учтены следующие соображения.

- 1) точки экстремумов сохраняют свои положения;

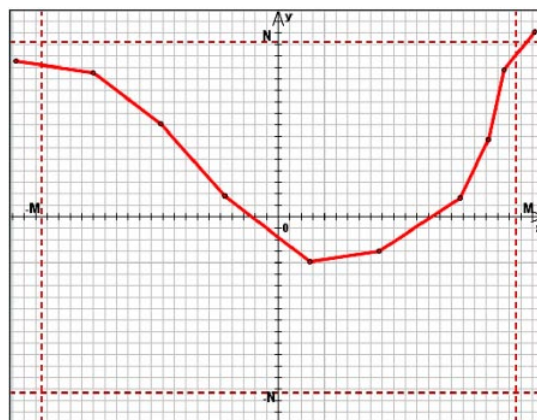


Рис. 1

2) если экстремум нестрогий, то между крайними точками постоянства строится отрезок прямой;

3) если более трех точек, идущих подряд, находятся на одной прямой, то, кроме крайних звеньев этого участка, остальные остаются без изменения;

4) участки монотонности сглаженной функции (как по оси абсцисс, так и по оси ординат) не меняются;

5) асимптотический характер определяется расположением участков графика в крайних («бесконечно удаленных» областях);

6) промежутки выпуклости и вогнутости сохраняются с точностью до естественной вариации расположения точки перегиба (которая не может быть определена точно по графику кусочно-линейной функции) и принятия решений (автоматически или предлагая ученику варианты решения) в случае конфликтных ситуаций, которые могут возникнуть при недостаточном количестве поставленных точек;

7) при изображении участков выпуклости и вогнутости часть точек, лежащих на участках выпуклости одного знака, могут игнорироваться (это можно считать моделированием повторных движений карандашом по одному участку графика, чтобы сглаживание получилось лучше, после того как ученик увидел, что некоторые участки графика конфликтуют с другими).

ЗАДАНИЕ БЕСКОНЕЧНО БОЛЬШИХ И БЕСКОНЕЧНО МАЛЕНЬКИХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Заметим, что классический курс анализа начинается с изучения пределов последовательностей, после чего результаты переносятся на пределы функций. В предлагаемой нами модели, ученик в первую очередь встречается с последовательностями. Единственным отличием является то, что абсциссы создаваемых точек графика не обязательно являются целыми (что, впрочем, легко исправить, привязав точки и вынуждая ученика ставить точки гра-

фика с последовательными целыми абсциссами).

В этом случае нам бы хотелось, чтобы интерфейс для ввода графика функции позволял естественным образом придавать смысл таким высказываниям как:

$\forall N (\exists M (\forall x (x > M \rightarrow f(x) > N)))$ – определение предела последовательности, равного «плюс бесконечности»,

$\forall \varepsilon > 0 (\exists M (\forall x (x > M \rightarrow |f(x) - \varepsilon| < \varepsilon)))$ – определение предела последовательности, равного нулю.

Проблемой здесь является необходимость проверки (и рисования на графике) бесконечного числа точек. Однако ни одна иллюстрация, естественно, такого требования обеспечить не может. На рисунках обычно принимаются условности, которые мы формализуем следующим образом.

Рядом с краями поля графика функции находятся две вертикальные и две горизонтальные линии, попадание графика за которые символизирует поведение «на бесконечности» и конечность или бесконечность предела.

Здесь мы считаем важным придать точкам в этих областях свойства, которые бы соответствовали высказываниям, определяющим различные виды предельного поведения.

Так, например, первое из приведенных выше определений читается так «для любого числа N найдётся такое M , что при всех x больших M , значение функции в них будет больше N ». Верхняя горизонтальная прямая в данном случае интерпретируется как $y = N$, а правая вертикальная – как $x = M$. Если при построении графика одна или несколько точек попадают в правый верхний угол, то это интерпретируется как описанный выше вид асимптотического поведения. В этом случае точкам графика, попавшим в эту область, будет приписано свойство быть связанными с прямыми $x = M$ и $y = N$, так чтобы при движении этих прямых сохранялось относительное положение точек в крайних областях (то есть, чтобы величина сдвига этих точек относительно крайних прямых не менялась).

Для второго примера понадобятся две горизонтальные прямые, близкие к оси абсцисс $y = -\varepsilon$ и $y = \varepsilon$. Точки, поставленные между этими прямыми правее прямой $x = M$, интерпретируются как лежащие ближе к оси, чем горизонтальные прямые, поэтому при их сближении ординаты поставленных точек уменьшаются пропорционально приближению горизонтальных прямых к оси абсцисс (происходит гомотетия относительно проекций точек на ось абсцисс).

Таким образом, появляется возможность динамической увязки точек с параметрами, которая адекватно визуализирует сформулированное в предикатной форме определение.

Для демонстрации одностороннего или разностороннего приближения к нулю или бесконечности можно использовать построение нескольких точек (как минимум двух), которые будут находиться в одной из областей разного знака или в разных областях. Это опять же соответствует традиции изображения различного предельного поведения на рисунках.

При изучении асимптотики функций необходимо ввести в рассмотрение возможность построения вертикальных асимптот. Механизм построения вертикальной асимптоты срабатывает автоматически, как только очередная точка графика поставлена в одну из граничных горизонтальных полос. При этом выделяется полоса, в которую попадают три последовательных по горизонтали узла. Если предыдущая точка не была обозначена как крайняя (нажатием правой кнопки мыши), то участок между ней и текущей точкой интерпретируется как «хвост асимптоты». На экране это изображается так: через средний узел проводится вертикальная асимптота, а в текущей точке, расположенной в соседнем слева узле, рисуется стрелочка, изгиб которой и приближение к асимптоте соответствуют обычному изображению асимптотического поведения графика. Если предыдущая точка была крайней точкой ветви графика, то изображение зависит от того, где находилась

предыдущая точка: во внутренней области или граничной. В первом случае поведение графика означает, что предел слева совпадает со значением функции, а предел справа – бесконечность (с тем или иным знаком). Во втором случае предел с обеих сторон равен плюс или минус бесконечности. В первом случае последние две точки могут находиться на соседних вертикальных линиях сетки, причём левая из них будет обозначать асимптоту для правой ветви графика, во втором случае между ними должна быть одна линия сетки, которая соответствует асимптоте обеих ветвей (если при рисовании графика ветки графика будут разнесены по горизонтали дальше, чем указано здесь, это будет интерпретировано как сужение области определения).

Более сложным является представление на рисунке конечных односторонних пределов. Для этого вводится механизм «эпсилон-дельта» прямоугольников (или ловушек), которые придают попавшим в них точкам свойства пропорционального изменения вместе с границами прямоугольников. Точный смысл использования «эпсилон-дельта ловушек» состоит в следующем. Координаты точки $M(x_i; y_i)$, поставленной внутри «ящика» $[a - \delta; a + \delta] \times [b - \varepsilon; b + \varepsilon]$, должны делать следующее высказывание истинным: $\forall \varepsilon > 0 (\exists \delta > 0 (|x_i - a| < \delta \rightarrow |y_i - b| < \varepsilon))$. Построение «эпсилон-дельта ловушки» осуществляется выбором соответствующе-

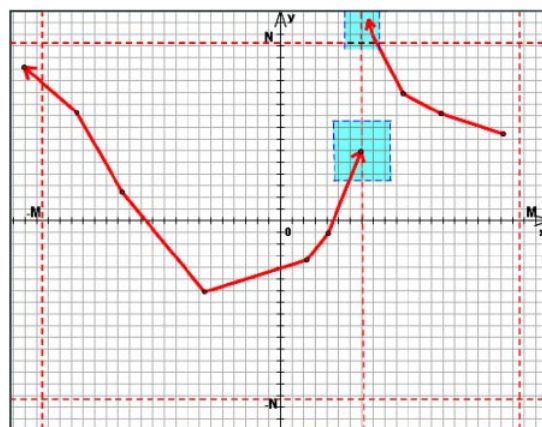


Рис. 2

го инструмента и щелчком на построенной точке графика. При этом на экране изображается квадратная область, центр которой совпадает с выбранной точкой. Построенная область включает 25 узлов сетки (две линии сетки слева и две справа позволяют передать характер стремления к пределу: приближается сверху, снизу или немонотонно). Построение «эпсилон-дельта ловушки» блокируется, если её изображение имеет пересечение с граничной областью (для разрешения проблемной ситуации достаточно уменьшить масштаб рисунка).

Для демонстрации значения любого предела будет использоваться стрелка на конце ломаной. Рисование стрелки, как уже говорилось выше, будет использоваться также для демонстрации бесконечных пределов и для поведения на бесконечности.

Использование стрелок для иллюстрации конечных пределов могло бы совсем заменить использование «эпсилон-дельта ловушек», однако при этом пропадёт возможность демонстрации динамической зависимости координат точек от параметров «эпсилон» и «дельта», которое особенно важно, если в область ящика поставлено несколько точек.

Представленный на рис. 2 введённый график передаёт следующие свойства функции:

1) область определения функции состоит из двух промежутков: полубесконечного открытого промежутка (интервала) и конечного промежутка, не включающего левую границу, но включающего правую;

2) на левом промежутке области определения функция ограничена, имеет один минимум, до которого функция убывает, а после которого возрастает, два промежутка выпуклости и один – средний – вогнутости;

3) на правом промежутке из области определения функция не ограничена сверху, монотонно убывает и является вогнутой;

4) в точке разрыва функция имеет конечный положительный предел слева, к

которому функция стремится, монотонно возрастающая, и предел «плюс бесконечность» при стремлении к точке разрыва справа.

ПРИМЕРЫ ОЛИМПИАДНЫХ ЗАДАЧ

Для того чтобы проиллюстрировать некоторые возможности разработанного инструмента, рассмотрим несколько примеров задач по математике, которые можно использовать на олимпиадах различного уровня.

Первая серия задач связана с построением графиков по заданным свойствам.

Задача 1-1. Построить график функции, имеющий два максимума и ни одного минимума.

Задача 1-2. Построить график функции, у которой значение функции в точке максимума меньше значения в точке минимума.

Комментарий. Для решения этих задач нужно догадаться, что у функции должен быть разрыв.

Задача 1-3.

а) Построить график нечётной функции, проходящий через точки $A(-4; 2)$, $B(1; 2)$, $C(3; -3)$.

б) Построить график нечётной функции, проходящий через точки $A(-4; 2)$, $B(1; 2)$, $C(3; -3)$ и имеющий не более одного минимума.

Комментарий. Для решения пункта а) этой задачи нужно достроить симметричные точки по определению нечётной функции, а потом провести через эти точки график. Для решения пункта б) нужно догадаться, что у функции должен быть разрыв.

Второй тип задач относится к классическим – «построить график функции, заданной формулой или текстовым описанием».

В данном случае следует ограничиться кусочно-линейными функциями, однако среди них есть много достойных внимания.

Задача 2-1. Над графиком функции $y = x$ выполнены следующие преобразования:

- 1) $f(x) \rightarrow f(x-1)$,
- 2) $f(x) \rightarrow |f(x)|$,
- 3) $f(x) \rightarrow f(|x|)$,
- 4) $f(x) \rightarrow f(x)-1$,
- 5) $f(x) \rightarrow |f(x)|$.

Постройте получившийся график.

Задача 2-2. Функция $y = \text{sign}(x)$ определяется следующим образом:

$$\text{sign}(x) = \begin{cases} 1 & \text{при } x > 0 \\ -1 & \text{при } x < 0 \\ 0 & \text{при } x = 0 \end{cases}.$$

Нарисуйте график функции

$$y = \text{sign}(x+1) + \text{sign}(4-x).$$

Замечание. Если в предыдущем типе задач мы имели дело по существу с последовательностями, и нас интересовали только значения функции в конечном числе точек, то данный тип задач предполагает проверку равенства функций в бесконечном числе точек. Для этого нужен аппарат сравнения конечного множества отрезков, которыми задается вводимая функция и может быть задан ответ. Сравнение множеств отрезков не представляет трудностей, если учесть, что один отрезок может быть представлен объединением нескольких (перекрытием отрезков можно пренебречь, передав слежение за отсутствием таких ситуаций интерфейсу ввода).

Другой тип задач связан с аккуратной записью на языке пропозициональных формул тех или иных свойств графиков функций и с экспериментами с примерами функций, позволяющими сделать самопроверку гипотез.

Задача 3-1. Для решения задач будем использовать язык, формальное определение которого дано в справке. Для примера, утверждение о том, что график лежит в верхней полуплоскости записывается так $\forall x (f(x) > 0)$. Запишите на этом языке пропозициональных формул свойство «график функции лежит выше биссектрисы первого и третьего координатных углов».

При решении таких задач в процессе подготовки к олимпиаде можно исполь-

зовать режим верификации решений. В этом режиме будут случайно генерироваться примеры функций, на которых эталонный предикат будет сравниваться с введенным предикатом. Примеры, на которых значения предикатов различаются, демонстрируются участнику.

ЗАКЛЮЧЕНИЕ

Представленный подход демонстрирует возможность соединить традиции графического отображения информации с однозначным способом их интерпретации, что позволяет расширить множество способов ввода решения задачи при дистанционном обучении, приблизив их к тем, посредством которых происходит обычное (очное) общение учителя и ученика. Последнее является важным условием эффективной реализации Интернет-олимпиад по математике [10].

Вторым важным аспектом представленного подхода является возможность описывать условие задачи формально, используя язык логики первого порядка. Этот формализм, с одной стороны, даёт возможность для автоматической проверки решения без специально указанного эталонного ответа или решения [11]. С другой стороны, возможность использования языка логики позволяет поддержать задачи на аккуратное построение определений, что особенно важно на начальном этапе изучения высшей математики (анализа бесконечно малых).

Разработанный модуль позволяет поддерживать конструктивные задачи на понятия монотонной функции, экстремума, промежутков возрастания и убывания, асимптотического поведения функции, выпуклости и вогнутости графиков и различные сочетания этих свойств. Использование разработанного модуля позволяет ученикам тренироваться в формулировке аккуратных определений понятий математического анализа, таких как определение понятия предела на плюс-минус бесконечности, определение ограниченности функции сверху и снизу и др.

Литература

1. Гелбаум Б., Олмстед Дж. Контрпримеры в анализе. М.: Мир, 1967.
2. Рукишин С.Е. Технологическая поддержка стимулирующих занятий математикой // Компьютерные инструменты в школе, 2009. № 6. С. 3–12.
3. Каган Б., Никифоров М., Энтина С. Лабораторные работы по математике. [Электронный ресурс]: Учебное электронное издание / СПб.: Изд-во Центра профессионального обновления «Информатизация образования», 1992.
4. Математика. Практикум: 5–11 классы [Электронный ресурс]: Учебное электронное издание / Под ред. В.Н. Дубровского. М.: ЗАО «1С», 2004.
5. Иванов С.Г. Компьютерная поддержка решения математических задач как способ организации продуктивной деятельности учащихся / Диссертация на соискание ученой степени кандидата педагогических наук. М., 2004.
6. Манцеров Д.И. Среда Verifier-KD: верификация решений задач по математике // Компьютерные инструменты в образовании, 2006. № 4. С. 36–41.
7. Манцеров Д.И. Система верификации для параметрических классов задач по математике. «Научно-технические ведомости СПбГПУ № 5, 2008. Информатика. Телекоммуникации. Управление». – Федеральное агентство по образованию. Санкт-Петербургский государственный политехнический университет, 2008. С. 183–189.
8. Манцеров Д.И. Конструирование интерактивных задачников по математике на основе использования различных представлений изучаемого объекта. «Информационные технологии моделирования и управления. Научно-технический журнал. Выпуск 8(51), 2008». Воронеж: «Научная книга. 2008». С. 881–888.
9. The On-Line Encyclopedia of Integer Sequences... (OEIS...) [Электронный ресурс] / Режим доступа: <http://oeis.org/classic/> свободный, Загл. с экрана. Яз. англ./ (последнее обращение: 19.06.2010).
10. Рукишин С.Е. Классификация типов научных соревнований с автоматической обработкой решений // Научно-технический вестник СПбГУ ИТМО, 2010. № 3. С. 121–125
11. Богданов М.С., Дубров С.Ю., Поздняков С.Н. Декларативная модель формального описания математических задач в системе проведения олимпиад WISETASKS и ее автоматическая проверка. Труды научно-технической конференции «Компьютерное моделирование 2009», 23–24 июня 2009 года, Санкт-Петербург, изд-во Политехнического университета, 2009. С. 179–180.

Abstract

The article is devoted to development of software to support an important type of mathematical problems: to construct mathematical object when his properties are given. Problems descriptions are formalized by predicates including standard mathematical operations and numerical functions, logic operations and quantifiers. The developed technology is illustrated by the functional problems. To plot function by «hand» the authors consider its linear simplification. To mark asymptotic behavior some special tools are proposed.

Keywords: eLearning, statements verification, predicates, limit definitions visualisation, function plot input.

*Манцеров Дмитрий Ирикович,
инженер по тестированию
фирмы «Zodiac Interactive»,
mdi82@mail.ru,*

*Рукишин Сергей Евгеньевич,
доцент кафедры математического
анализа РГПУ им. Герцена,
vliuser@gmail.com*



Наши авторы, 2010.
Our authors, 2010.