УДК 681.3

М.С. Львов, А.В. Спиваковский (Херсонский гос. пединститут, Украина)

МЕТОДЫ ПРОЕКТИРОВАНИЯ СИСТЕМ КОМПЬЮТЕРНОЙ ПОДДЕРЖКИ МАТЕМАТИЧЕСКОГО ОБРАЗОВАНИЯ.

Резюме. Настоящий доклад посвящен методам проектирования прикладных программных систем, поддерживающих практическую математическую деятельность учащегося, включая поддержку символьных преобразований. Предложена общая концепция такой системы, ее структурная схема, базирующаяся на естественной иерархии метаматематических понятий. В качестве примеров рассматриваются прототипы систем, поддерживающих математическую деятельность в различных предметных областях:

⇒ Линейная алгебра;

⇒ Тригонометрия.

Введение. Многие традиционные компьютерные курсы математики базируются на идеях программированного обучения, хотя и используют все аппаратные и программные возможности современной вычислительной техники и новые методы представления знаний. Наиболее развитой и совершенной как с методической, так и с технической точек зрения при этом оказывается лекционная часть курса. Проблема адекватной поддержки практических занятий по математике менее разработана и представляется нам актуальной.

Известные системы компьютерной алгебры (CAS) – Axiom, Mathematica, Maple, MathCad и др., в значительной мере решают задачу поддержки профессиональной математической деятельности, связанной с символьными вычислениями и численными расчетами. Вторая группа программных систем - системы автоматического доказательства теорем (ATP) обеспечивают поддержку важного аспекта математической деятельности - логического вывода.

В настоящее время активно разрабатываются проблемы интеграции CAS и ATP в единую систему [1]. Интегрированные системы в значительно большей степени оказываются пригодными для образовательных целей.

Предлагаемый ниже подход к проектированию интегрированных систем рассматривает с единых позиций и класс вычислительных задач, и класс задач на доказательства. Такой подход, с нашей точки зрения, является перспективным. Проектируемая в рамках этого подхода система получила рабочее название **TerM**.

Основное назначение системы TerM - компьютерная поддержка практических занятий и лабораторных работ по математике - т.е. активной математической деятельности пользователя (студента). В процессе такого рода деятельности студент использует теоретические знания, приобретенные на предыдущих стадиях обучения, для решения практических задач.

1. Концепция компьютерной поддержки математической деятельности.

Математическая деятельность пользователя заключается в:

⇒Построении математических объектов;

⇒Распознавании свойств математических объектов;

⇒Преобразованиях математических объектов.

© М.С.Львов, А.В.Спиваковский, 1998

Математическая деятельность осуществляется в рамках соответствующей **предметной области**, описываемой конструктивно и аксиоматически. Это означает, что:

⇒ математические объекты определены в предметной области в виде математических конструкций;

⇒ свойства математических объектов описаны аксиоматически;

⇒ преобразования объектов определены в виде списка допустимых (элементарных) преобразований;

Математическая деятельность направлена на решение **математической задачи** как основной семантической единицы деятельности. Ход решения задачи представляет из себя последовательность шагов, на каждом из которых пользователь:

⇒ распознает некоторое свойство математического объекта, определенного в решаемой задаче;

⇒ преобразует этот объект.

Таким образом, процесс решения задачи – последовательность вида

$$S_1 \xrightarrow{T_1} S_2 \xrightarrow{T_2} \quad \ldots \quad S_i \xrightarrow{T_i} S_{i+1} \quad \ldots \quad \xrightarrow{T_n} S_n$$

где S_i – математические объекты, а T_i – их элементарные преобразования

Математическая деятельность в рамках (изучаемой) предметной области опирается на использование средств и методов решения задач из других (более элементарных, изученных, усвоенных ранее) предметных областей.

Компьютерная поддержка математической деятельности заключается в предоставлении пользователю набора средств и инструментов, автоматизирующих и верифицирующих процесс решения математической задачи. Эта поддержка может осуществляться на нескольких уровнях:

Уровень 0: специализированный редактор, ориентированный на математическую деятельность (реализация адекватного представления математических объектов в описании хода решения математической задачи), снабженный математическими инструментами и пособиями (калькулятор, геометрические инструменты, справочник и т.д.)

Уровень 1: система, осуществляющая синтаксический анализ математических объектов и автоматизирующая процедуры их преобразований в ходе решения математической задачи;

Уровень 2: система, осуществляющая семантический анализ преобразований математических объектов (проверка правильности одного шага решения задачи);

Уровень 3: Система семантического и методического анализа хода решения математической задачи (проверка правильности хода решения задачи как с математической, так и с методической точки зрения);

II. Примеры.

Концепция компьютерной поддержки решения математической задачи реализована для нескольких классов задач в различных предметных областях (Например численные методы анализа [5]. Мы рассмотрим только две предметных области – линейную алгебру и тригонометрию.

Система Мир Линейной Алгебры. / 1989 г. /

Построение системы МЛА [2] – один из ранних и наиболее полных примеров применения описываемого подхода. В системе Мир линейной алгебры реализована поддержка процесса решения основных задач линейной алгебры в объеме курса алгебры для университетов и пединститутов [3]. Алгоритмы, реализованные в системе, используют технику элементарных преобразований матриц и методы компьютерной линейной алгебры над полем рациональных чисел.

Краткое описание системы.

1. **Математические объекты** - матрицы, векторы, многочлены.
2. **Специальные аксиомы** - аксиомы евклидового векторного пространства, аксиомы линейной алгебры (алгебры матриц);
3. **Преобразования**:
- элементарные преобразования матрицы (пары векторов);
- вычисления в линейной алгебре матриц;
- вычисления рациональных корней многочлена.
4. **Основные задачи**
- Решение системы линейных уравнений;
- Вычисление определителя матрицы;
- Обращение матрицы;
- Построение характеристического многочлена;
- Построение собственных векторов матрицы;
- Построение жордановой формы матрицы;
- Ортогонализация системы векторов.
5. **Ход решения**. Решение каждой задачи - последовательность шагов, на каждом из которых осуществляется одно из преобразований. На любом шаге возможен вызов эксперта, который подсказывает следующее действие и/или выполняет его.
6. **Системные теории и интегрированная оболочка (Сервис)**
- Автоматическая генерация заданий с рассылкой их по сети на рабочие места учащихся;
- Встроенные в HELP методические указания теоретического характера;
- Фиксация хода решения задачи с диагностикой системы в специальном файле с возможностью получения твердой копии;

Уровень системы.

В системе по существу реализован 3-ий уровень поддержки практических работ по линейной алгебре в рамках соответствующей учебной дисциплины для студентов университетов и педагогических институтов.

Система Тригонометрия. / 1993 г. /

В системе Тригонометрия [3] реализована поддержка процесса решения основных тригонометрических задач школьного курса:
- Упрощения тригонометрических выражений;
- Доказательство тригонометрических тождеств;
- Решение тригонометрических уравнений.

Алгоритмы, реализованные в системе, используют технику символьных преобразований и методы компьютерной алгебры.

Краткое описание системы.

1. **Математические объекты** - тригонометрические выражения, равенства, уравнения.
2. **Специальные аксиомы тригонометрии** представлены списком тригонометрических тождеств, классифицированных следующим образом:
   - Основные значения тригонометрических функций;
   - Формулы приведения;
   - Формулы удвоенного и половинного аргумента;
   - Формулы суммы;
   - Основные тригонометрические тождества;
   - Решения простейших тригонометрических уравнений.
3. **Преобразования** тригонометрических выражений классифицированы для каждой задачи, поддерживаемой системой:

Задача **Упрощения**

Тригонометрические преобразования, соответствующие тригонометрическим тождествам системы; тождественные алгебраические преобразования; прямые и обратные подстановки.

Задача доказательства **Тождества**

Преобразования упрощений;

Условно тождественные преобразования:

A*C = B*C, C <> 0 ==> A = B,   A/C = B/C C <> 0 ==> A = B

Задача решения **Уравнения**

- Преобразования упрощений;
- Условно тождественные преобразования:
  A*C = B*C, C =|= 0 ==> A = B, { C =|= 0 означает "C не имеет корней" }
  A/C = B/C C =|= 0 ==> A = B
- Преобразования факторизации:
  A*B = 0 ==> A = 0 или B = 0,
  A^k = 0 ==> A = 0
- Решения алгебраических уравнений степени N <= 3

4. Системные теории

Ход решения тригонометрической задачи – последовательность шагов, на каждом из которых осуществляется преобразование выражения. Каждый шаг решения содержит применяемое преобразование и преобразованное выражение.

Решение задачи может осуществляться в одном из режимов:

4.1.**Check-Step Mode** В этом режиме пользователь сам осуществляет преобразования выражений, а система проверяет правильность применяемых элементарных преобразований и правильность результата на каждом шаге.

4.2.**Auto-Step Mode** В этом режиме на каждом шаге пользователь указывает на выполняемое преобразование и место его применения в выражении, а система выполняет указания пользователя.

4.3. **Mixt-Step Mode** В этом режиме объединены возможности Auto-Step и Check-Step.

4.4 Сервис.

Ход решения задачи хранится в электронной тетради Copybook. Система снабжена Калькулятором, Построителем графиков и Справочником формул. Подсистема Solver генерирует решения основных задач, поддерживаемым в разделе Тригонометрия.

5. Заключение.

В системе реализован 2-ой уровень поддержки. Методы решения основных задач носят общий характер и могут быть использованы в реализациях других предметных областей элементарной и высшей алгебры: Алгебра радикалов; Логарифмы и экспоненты; Символьное дифференцирование; Символьное интегрирование и т.д.

Выбор тригонометрии в качестве предметной области, в реализации которой использованы символьные преобразования и методы компьютерной алгебры, обусловлен:

- большим количеством специальных соотношений, что позволило экспериментально оценить вычислительные ресурсы, необходимые системам такого рода;
- возможностью использования реализованных алгоритмов для проектирования систем поддержки задач геометрии.

III. Проект интегрированной системы TерM.

Система TerM проектируется как структура, представленная на рис 1.

Рис 1 Структура системы TerM

В основе структуры - представление ядра в виде иерархии метаматематических понятий

Данные → **Объекты** → **Задачи** → **Теории**

Алгебраические типы данных

Анализ курса математики показывает, что для реализации системы TerM необходимо реализовать следующие алгебраические типы данных:

Конструктивно-определенные алгебраические типы: линейные алгебры - алгебры матриц, линейные векторные пространства, полугруппы слов и группы подстановок, и другие алгебраические типы, используемые математическими теориями;

Символьные типы элементарных функций: Специальные классы элементарных функций, включающие логарифмы и экспоненты, операторы дифференцирования и интегрирования, рациональные тригонометрические функции и тригонометрические полиномы;

Теоретико-множественные и логические типы: алгебра множеств типа арифметических прогрессий, алгебра конечных подмножеств; алгебра логики, и т.д.

Символьные алгебраические типы: рациональные функции и полиномы многих переменных, линейные полиномы многих переменных и Pi;

Численные тригонометрические типы: поля деления круга и поля Галуа;

Численные алгебраические типы: комплексные числа, радикальные расширения поля рациональных чисел различных типов, рациональные числа, целые числа.

Основой (нижним уровнем) системы ТерМ является полиструктурная иерархия алгебраических типов данных, построенная как **структура** абстрактных (аксиоматических) и конструктивных алгебраических систем. Такой подход стал особенно популярным в построении CAS с появлением системы Axiom. Основными структурами иерархии являются:

- **структура наследования** абстрактных и конструктивных свойств алгебр;
- **структура основности** (многоосновных) алгебр;
- **структура расширений** (т.е. конструкций и вложений) алгебр;
- **структура морфизмов** подобных алгебр

Так, алгебра аргументов **ARG** подсистемы Тригонометрия - алгебра линейных комбинаций вида

$$\mathbf{A_1 * x_1 + A_2 * x_2 + \ldots + A_n * x_n + A_{n+1} * Pi}$$

- наследует абстрактные свойства линейного пространства;
- основана на поле скаляров **Scal** (поле скаляров Scal – параметр алгебры ARG)
- конструктивно представляет из себя линейное динамическое расширение конечномерных векторных пространств $\mathbf{W_0 \subseteq W_1 \ldots \subseteq W_i \subseteq \ldots}$, где Wi определены как:

$$\mathbf{W_0 = Scal,} \qquad \mathbf{W_{i+1} = W_i + Scal * x_{i+1},} \qquad \mathbf{0 * x_{i+1} = 0}$$

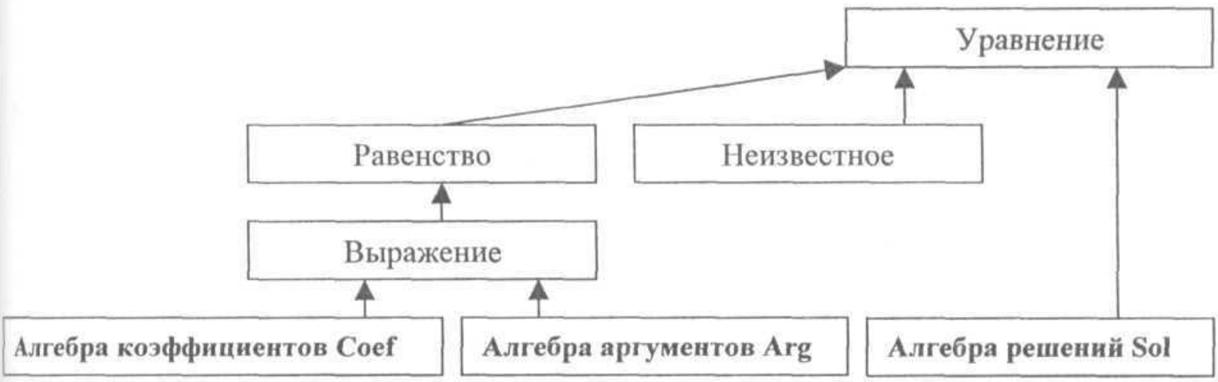
- Изоморфна алгебре многочленов одного переменного U, рассматриваемой как векторное пространство, если определить изоморфизм Fi следующим образом:

$$\mathbf{Fi(A_i U^i) = A_i * X_i}$$

Математические объекты

Следующий уровень системы – уровень конструктивных определений математических объектов – также представляет из себя полиструктурную иерархию, основанную на иерархии алгебраических типов. Характерные примеры математических объектов – уравнение, график функции, система линейных уравнений. Трансцендентное уравнение, например, как математический объект, описывается следующей структурой (рис. 2)

Рис.2 Структура объекта Уравнение



Отметим, что структура наследования алгебраических типов содержит описания методов решения уравнений в соответствующих алгебрах. Так, абстрактная алгебра **Поле частных** содержит следующее правило решения уравнения

$$\mathbf{(A*x = B) \;=\; (x = A^{-1}*B)}$$

Прикладные и системные задачи.

Как и нижние уровни системы, уровень Задачи также описывается в виде полиструктурной иерархии. Мы определяем, однако, два принципиально различных типа задач:

- Прикладные задачи;
- Системные задачи.

Прикладные задачи – это те математические задачи, на поддержку процесса решения которых ориентирована система. Системные задачи решают собственные проблемы системы – проблемы поддержки процесса решения прикладных задач.

Вот, например, список системных задач раздела Simplifications теории Check-Step Mode

- Проверка правильности специального элементарного преобразования;
- Проверка правильности применения специального элементарного преобразования;
- Проверка правильности применения общего преобразования;
- Проверка правильности выполнения подстановки;

Отметим, что описания методов решения прикладных задач – неотъемлимая часть иерархий нижнего уровня (математических объектов и алгебраических типов данных), в то время как <u>системные задачи образуют собственную, независимую иерархию</u>, только использующую иерархии нижнего уровня.

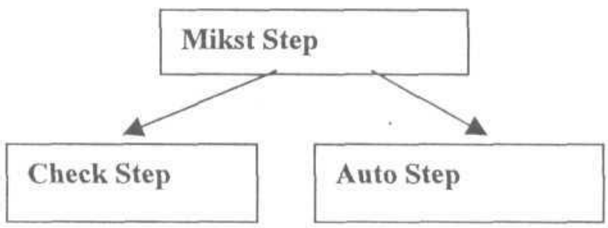
Теории

Теории – это структуры, объединяющие соответственно прикладные и системные задачи в единое целое. Так, структура основных задач линейной алгебры образует теорию Линейная алгебра, а структура тригонометрических задач – Тригонометрию. Поэтому и теории классифицируются как прикладные и системные. При этом Прикладная теория играет роль параметра Системной теории.

Режим системы ТерМ = <Системная теория> (<Прикладная теория>)

Так, в режим **Auto-Step(Тригонометрия)** поддерживает автоматическое решение иерархии тригонометрических задач

Иерархия системных теорий Тригонометрии



Математические инструменты – это специальные системные теории, предоставляющие пользователю дополнительные средства для решения задач. Это, например, Калькулятор, Решатель, Доказыватель, Построитель графиков. Перечислим основные задачи этих инструментов:

Калькулятор: Найти значение выражения (каноническую форму); Факторизовать выражение; Преобразовать к другому виду, и т.д.

Доказыватель: Проверить тождество; Проверить условное тождество;

Решатель: Решить уравнение; Найти количество решений на отрезке; Определить существование решения;

Построитель графиков: Построить график; Построить график производной: Построить аппроксимирующий график;…

Математическая литература: Рабочая тетрадь; Справочник математических формул; Задачник; Учебник.

IV Методология проектирования системы TerM.

При проектировании вычислений в алгебрах данных, а также системных и прикладных задач основным инструментом являются алгебраические преобразования выражений (математических объектов). Алгебраические преобразования естественным образом описываются т.н. системами переписывающих правил (RSys).

Удобным механизмом описания преобразований с помощью RSys являются разнообразные правила обхода преобразуемого выражения - так называемые стратегии переписывания (RStr).

Использование RSys и RStr как основных средств - отличительная особенность т.н. языков алгебраического программирования. Для прототипирования и проектирования нижнего уровня системы TerM мы используем экспериментальную систему алгебраического программирования APS [6], разработанную в ИКАН Украины.

Проектирование иерархий алгебраических данных, математических объектов и прикладных задач требует средств объектно-ориентированного алгебраического проектирования. С этой целью разработано ОО-расширение языка APLAN - языка системы APS, включающее необходимые средства. Этот язык используется как язык спецификаций задач нижнего уровня системы.

Вот, например, описания вычислений в поле частных области целостности в терминах систем переписывающих правил:

```
Add (IntDomain x, y, u, v) {
        x/v + u/v = (x + u) !/ v,
        x/y + u/v = (x*v + u*y) !/ (y*v)),
          x + u/v = (x*v + u)/v,
          x/y + u = (x + u*y)/y
        };

Sub (IntDomain x, y, u, v){
        x/v - u/v = (x - u) !/ v,
        x/y - u/v = (x*v - u*y) !/ (y*v)),
        x - u/v = (x*v - u)/v,
        x/y - u = (x - u*y)/y
        };

Mult(IntDomain x, y, u, v){
        x/v * u/v = (x*u) / v^2,
        x/y * u/v = (x*u) !/ (y*v),
        x * u/v = (x*u) !/ v,
        x/y * u = (x*u) !/ y
        };

Div (a, b; IntDomain x, y, u){
        (x/u) / (y/u) = x !/ y,
        a / b = a * Inverse(b),
        (x/u) / y = x !/ (y*u),
        x / y = Simplify(x, y)
        x !/ y = Simplify(x, y)
        };

Pow(a; QuotPart x,y; Integer n){
        n<0 -> (a^n = Inverse(a)^(-n)),
             (x/y)^n = x^n/y^n
        };
```

Проектирование верхнего уровня системы осуществляется снизу-вверх в рамках объектно-ориентированного подхода в соответствии с иерархией метаматематических понятий в любой из систем, поддерживающих ОО подход (например Visual C++).

Структуры верхнего уровня используют структуры нижнего уровня (соответствующие процедуры верхнего уровня параметризованы именами нижнего уровня). Поэтому проектирование структуры определенного уровня может быть осуществлено (относительно) независимо от проектирования других структур. Так, проектирование алгоритма решения квадратного уравнения $aX^2 + bX + c = 0$ не зависит от алгебры коэффициентов, если в ней реализованы все необходимые для вычислений операции, а проектирование системной теории - режима Check-Step не зависит от списка специальных аксиом и элементарных преобразований предметной области.

Авторы выражают благодарность чл.корр. АНУ А.А.Летичевскому, доц. В.А.Крекнину, С.В.Конозенко, В.С.Левашеву, В.Г.Маринченко, А.В.Булату за долговременное и плодотворное сотрудничество.

1. Homann K., Calmet J. Combining Theorem Proving and Symbolic Mathematical Computing. // Lect. Notes of Comp. Sci. (to print)

2. Спиваковский А.В. Педагогические программные средства: объектно-ориентированный подход. // Информатика и образование. - 1990.- № 2, С. 71-73.

3. Співаковський О.В., Крекнін В.А. Лінійна алгебра. Навч. посібник. – Херсон, 1997. - 148 с.

4. Львов М.С. Алгоритмы компьютерной поддержки решения школьных алгебраических задач и их реализация в APS. // Кибернетика (в печати)

5. Bulat A.V., Lvov M.S, Marinchenko V.G. Electronic Table Shifting According to Data. // Proc. of the Intern. Conf. on Computer Tech. in Aducation (ICCTE'93).- Kiev, Sept. 14-17, 1993.- pp 153-155.

6. Letichevsky A.A., Kapitonova J.V., Konozenko S.V. Computations in APS. // Theoretical Computer Science 1993.- 119.- pp. 145-171