

ОБ ОДНОМ ПОДХОДЕ К ПРОЕКТИРОВАНИЮ АЛГЕБРАИЧЕСКИХ ТИПОВ ДАННЫХ

В.С. Песчаненко

Институт кибернетики им. В.М. Глушкова НАН Украины,
03680, Киев, проспект Академика Глушкова, 40
vladim@ksu.ks.ua

Робота присвячена методам вирішення проблем які виникли при проектуванні алгебраїчних типів даних шкільної системи комп'ютерної алгебри ТерМ. Описана ієрархія алгебраїчних типів даних цієї системи, розглянуті основні методи проектуванні та реалізації алгебраїчних обчислень, які ґрунтуються на використанні технологій символічних перетворень.

The article is dedicated to the methods for solving the problems, aroused during design of school computer algebra system TerM algebraic data type. The algebraic data type hierarchy of the present system is described in the article; also the basic methods of design and realization of the algebraic computing, based on the usage of symbolic conversion technologies, are examined.

Введение

Существующие коммерческие системы компьютерной алгебры (Derive, Mathematica, Maple, MathCad и др.) в основном решают проблему поддержки профессиональной математической деятельности. Отметим, что использование таких систем в учебных целях в школе ограничено. Среди нескольких причин выделим главную: они ориентированы на получение ответа математической задачи, а не на получение хода ее решения, что является главной целью ученика. Поэтому системы компьютерной алгебры учебного назначения (системы школьной компьютерной алгебры) должны поддерживать ход решения математической задачи. Эта поддержка заключается либо в генерации шага решения задачи в соответствии с командой пользователя, либо в проверке математической правильности шага решения, если он выполнен пользователем.

Математическая деятельность пользователя заключается в следующем:

- ⇒ построении математических объектов;
- ⇒ распознавании свойств математических объектов;
- ⇒ преобразованиях математических объектов.

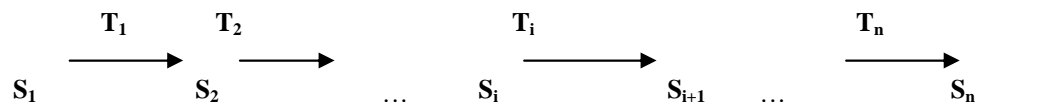
Математическая деятельность осуществляется в рамках соответствующей **предметной области**, описываемой конструктивно и аксиоматически. Это означает, что:

- ⇒ 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 собой 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



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

С логической точки зрения ход решения задачи – это логический вывод в соответствующей эквациональной теории, который, как известно, заключается в подстановке термов вместо предметных переменных и замена термов равными термами.

ТерМ – это единственная система школьной компьютерной алгебры, которая, в числе прочего, поддерживает математическую деятельность учащегося.

Еще одна принципиальная отличительная особенность ТерМ заключается в том, что в системе реализован полный набор учебных объектов, что позволяет эффективно организовать учебный процесс. Это – дидактические материалы (математический учебник, задачник, справочник, тетрадь) и математические

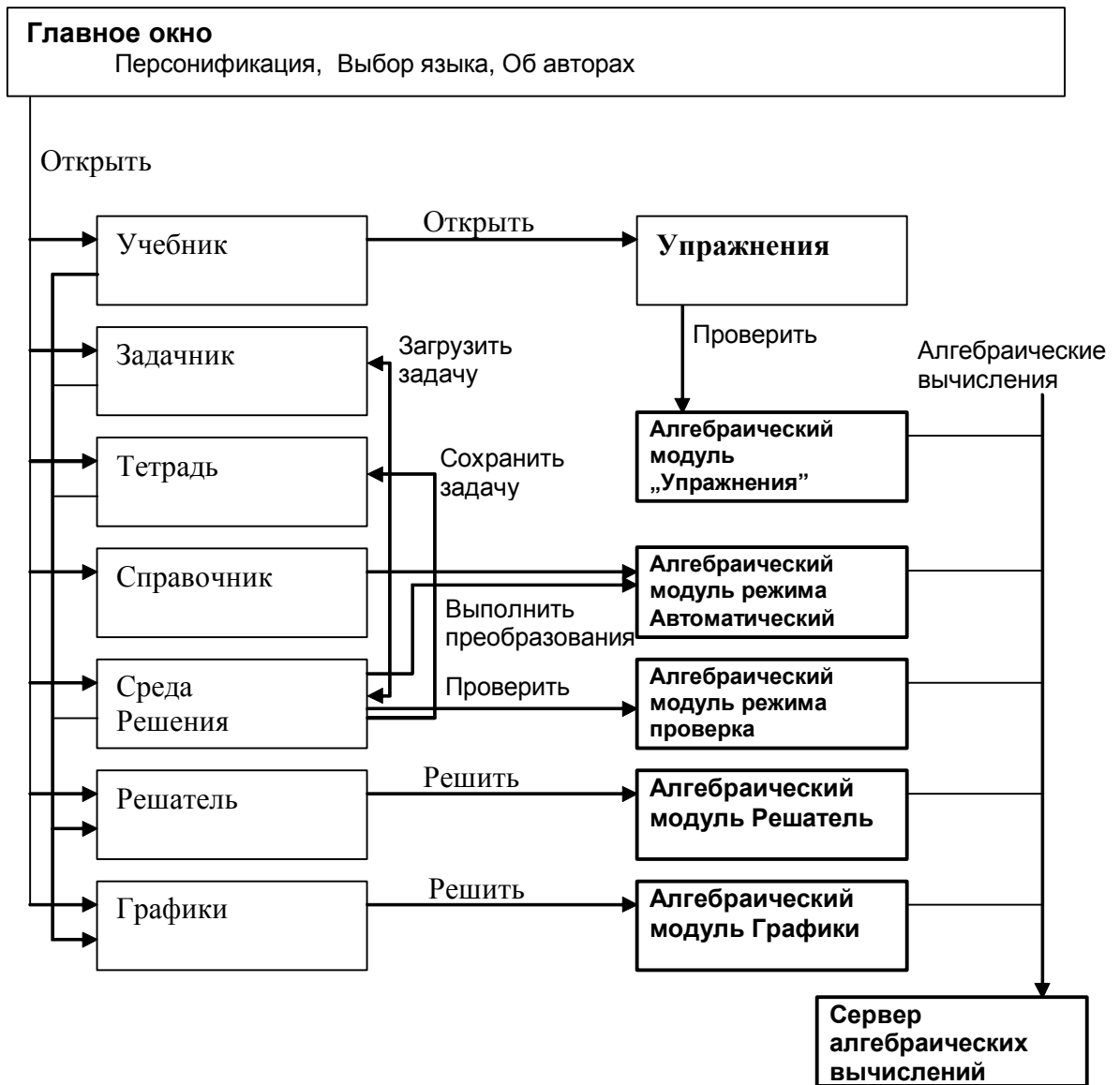


Рис.1. Объектная модель ТерМ

инструменты (среда решения (задач) решатель и графики). Общая объектная модель системы ТерМ показана на рис. 1.

Жирным выделены программные модули, реализация которых требует использования методов компьютерной алгебры и технологий алгебраического программирования.

Таким образом, с точки зрения логической архитектуры система ТерМ – типа клиент-сервер. Серверная часть ТерМ, в свою очередь, распределена на двух уровнях – реализации алгоритмов решения системных задач и реализации алгебраических вычислений в многосортной алгебре, представляющей соответствующую предметную область.

Уровень реализации алгоритмов решения алгебраических задач представлен алгебраическими модулями соответствующих учебных объектов и режимов.

Уровень реализации алгебраических вычислений представлен модулем *сервер алгебраических вычислений*.

Таким образом, возникает важная проблема проектирования многосортных алгебр. При проектировании основные усилия направлены на описания вычислений значений выражений над такими многосортными алгебрами.

Постановка задачи

В соответствии с техническим заданием на систему ТерМ 7–9 необходимо реализовать вычисления в следующих числовых алгебрах: целые, действительные, рациональные, комплексные и радикальные числа. При этом вычисления должны быть реализованы с неограниченной разрядностью. Отметим, что рациональные числа нужно представить тремя изоморфными реализациями: алгеброй дробей, алгеброй смешанных дробей и алгеброй десятичных периодических дробей. Кроме того, система ТерМ должна поддерживать вычисления в

таких алгебрах множеств, как алгебра конечных множеств) и алгебра числовых интервалов. Соответственно, в системе ТерМ нужно поддерживать вычисления в следующих символьных алгебрах данных: линейные, векторные, аффинные пространства линейных комбинаций, кольца многочленов от одной и многих переменных, поля рациональных функций от одной и многих переменных.

Таким образом, возникает проблема эффективной реализации, а, следовательно, и качественного проектирования алгебраических типов данных для системы типа ТерМ.

Проектирование иерархии алгебраических типов ТерМ осуществлялось в терминах частично-упорядоченных многосортных алгебраических систем [2].

Многосортные алгебры

Под многосортной алгеброй (МСА) T будем понимать многоосновную алгебраическую систему $A(A_1, \dots, A_k)$ с главным носителем A , носителями алгебр аргументов A_1, \dots, A_n , Σ , сигнатурой предикатов Π , сигнатурой констант E (выделенных элементов носителя).

$$T_{\Sigma, \Pi, E}[A(A_1, \dots, A_n)]. \quad (1)$$

Алгебры аргументов, в свою очередь, определяются как многосортные. Например, на рис. 2 показан фрагмент структуры многоосновности. Стрелочки, проведенные на нем из A_1, \dots, A_k в A означают, что сигнатура МСА A определяется через сигнатуры A_1, \dots, A_k .

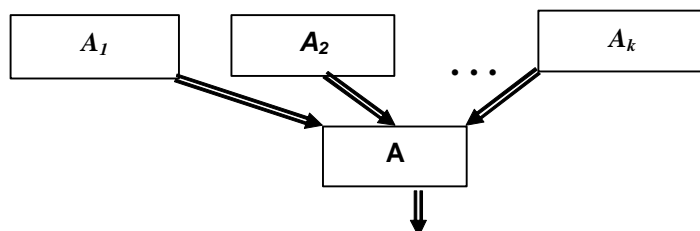


Рис. 2. Фрагмент структуры многоосновности

Мы описываем структуру МСА в терминах наследования алгебр, расширения алгебр и морфизмов алгебр.

Алгебра—элемент иерархии МСА описывается сигнатурой, носителем, алгоритмами реализации операций сигнатуры и набором аксиом.

Проектирование осуществляется в 2 этапа: прототипирования, анализ свойств прототипа и реализации окончательной версии.

Прототипирование заключается в построении МСА в терминах наследования, расширения, морфизмов и реализации построенной иерархии на языке АПЛАН в системе алгебраического программирования АПС [3].

Реализация заключается в выделении абстрактных алгебр, в терминах которых описываются алгебры иерархии и их реализации на нижнем уровне АПС.

Наследование

Механизм наследования является одним из основных характерных механизмов для объектно-ориентированного проектирования программных систем. При проектировании алгебр данных использование этого механизма имеет свои отличительные особенности. На содержательном уровне свойство наследования (B наследует A) означает, что:

- а) сигнатура B наследует сигнатуру A ; На B могут быть определены собственные элементы сигнатуры-операции, константы и отношения;
- б) на B могут быть доопределены правила вычислений (интерпретирующие правила) для каждого элемента сигнатуры A ;
- в) на B могут быть выполняться некоторые дополнительные свойства (например: коммутативность, ассоциативность, идемпотентность, отсутствие делителей нуля и т.д).

Пусть $T_{\Sigma_T, \Pi_T, E_T, AX_T}[A]$, $Q_{\Sigma_Q, \Pi_Q, E_Q, AX_Q}[B]$ — две многосортные алгебры, в которых дополнительно к (1) описаны еще системы аксиом AX_T , AX_Q . Будем говорить, что алгебра Q наследует алгебру T , если

$$A \triangleleft B, \quad \Sigma_T \triangleleft \Sigma_Q, \quad \Pi_T \triangleleft \Pi_Q, \quad E_T \subseteq E_Q, \quad AX_T \subseteq AX_Q. \quad (2)$$

Обозначение $A \blacktriangleleft B$ означает, например, что носитель B является уточнением носителя A в том смысле, что любой элемент B можно получить из некоторого элемента A путем доопределения (уточнения) некоторых фрагментов его структуры. Аналогичный смысл имеют и другие обозначения в (2). Некоторая алгебра может наследовать свойства и нескольких алгебр. В этом случае наследуемые свойства должны быть совместимыми.

Для строгого определения механизма наследования нам необходимо более детально рассмотреть те средства описания алгебр данных, которые используются в определении наследования.

Примеры наследования:

1. Абстрактная группа наследует свойства абстрактной полугруппы. В этом примере свойства полугруппы дополнены операцией a^{-1} и соответствующей аксиомой.

2. Абстрактное коммутативное кольцо наследует свойства абелевой группы (группы по сложению) и коммутативной полугруппы. Например, показано, множественное наследование. (рис. 3).

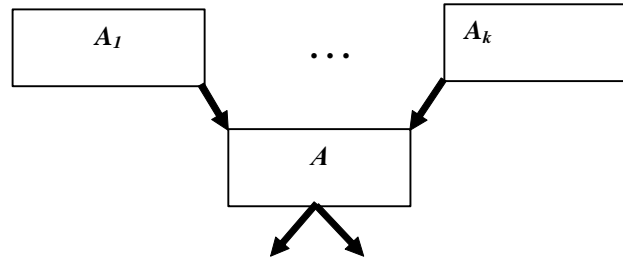


Рис. 3. Фрагмент структуры наследования

Стрелочки, проведенные из A_1, \dots, A_k в A на рис. 3 означают а), б), в) (см. выше).

Расширения

Пусть A и B - подобные алгебры. Алгебра B называется *расширением* алгебры A , если $A \subseteq B$ и ограничение алгебры B на подмножестве A совпадает с алгеброй A . Конструкция расширения широко используется для введения в алгебру новых операций или придания алгебре новых свойств.

Расширение алгебры описывается с помощью *конструкций* и *вложений*. Конструкции – это определения структуры носителя расширения алгебры, а вложения – это определения функций редукции элементов носителя расширения, принадлежащих расширяемой алгебре (функции преобразования типов).

Примером расширения может быть построение поля рациональных чисел R как расширения кольца целых чисел Z :

- элементами расширения называют конструкции вида a/b , где $a, b \in Z, b > 0, GCD(a, b) = 1$,

$$R = \{a/b, a, b \in Z, b > 0, GCD(a, b) = 1\}; \tag{3}$$

- на множестве R определяют операции кольца $Z(+, -, *)$. Например,

$$a/b + c/d = Reduce((a * d + c * b)/(b * d)); \tag{4}$$

- расширяют сигнатуру R , дополняя её операциями $(/, 0^{-1})$. Например,

$$(a/b)/(c/d) = Reduce((a * d)/(b * c)); \tag{5}$$

- осуществляют вложения кольца Z в R , определяя изоморфизм Z в R тождеством вложения

$$a/1 = a \text{ (функция Reduce)} \tag{6}$$

На рис 4, например, стрелка из A_1 в A_2 означает, что в A_2 является расширением A_1 определенное с помощью *конструкции* и *вложения*.

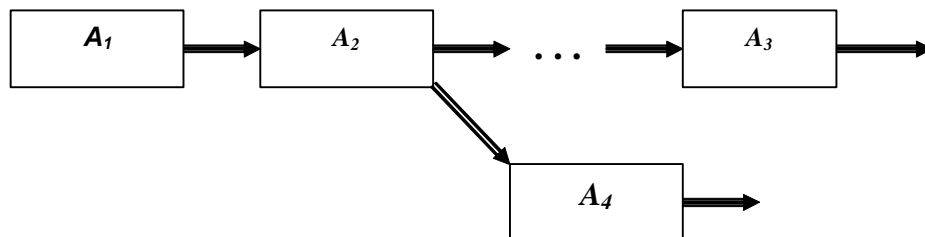


Рис. 4. Фрагмент структуры расширений-вложений

Морфизмы

Пусть A и B – подобные алгебры. Алгебра B называется гомоморфной алгеброй A , если существует такое отображение $\varphi: A \rightarrow B$, что для любой операции $F(a_1, \dots, a_n)$, сигнатуры Σ_A $\varphi(F(a_1, \dots, a_n)) = F(\varphi(a_1), \dots, \varphi(a_n))$, аналогичные по смыслу соотношения имеют место для предикатов и констант сигнатуры E_A и Π_A .

Подобные алгебры A и B называются изоморфными, если существует взаимнооднозначный морфизм: $\varphi: A \leftrightarrow B$.

Морфизмы алгебр – удобное средство для описания алгебраических вычислений во многих ситуациях. С точки зрения языка программирования морфизмы – это функции преобразования типа. Рассмотрим пример морфизма.

Алгебра A – рациональных чисел над полем целых чисел. Это означает, что элементы A имеют вид: $u = a/b$.

Алгебра B – десятичных периодических дробей над полем целых чисел в сигнатуре обобщенной числовой функции $PeriodNumeric$. Это означает, что элементы B имеют вид: $v = PeriodNumeric(Whole, PPeriod, Period)$, где $Whole$ – целая часть, $PPeriod$ – число до периода, $Period$ – число периода.

Вычисления значений выражений и решение других алгебраических задач коротко и эффективно реализуются в алгебре A , а вывод необходимо осуществлять в сигнатуре алгебры B .

Поэтому должны быть описаны алгебры A и B , причем вычисления в B описываются в терминах изоморфизма $\varphi: A \leftrightarrow B$ следующим образом:

$$\varphi((.)/(.)) = PeriodNumeric(Whole(.), PPeriod(.), Period(.)), \quad (7)$$

$$\varphi^{-1}(PeriodNumeric(Whole(.), PPeriod(.), Period(.))) = (.)/(.). \quad (8)$$

Если f, g – элементы алгебры B , и $*$ – одна из бинарных операций сигнатура B , то результат $f*g$ вычисляется по формуле: $f*(g = \varphi^{-1}(\varphi(f) * \varphi(g)))$.

В более общем виде, если F – выражение над A , то его значение вычисляется как $\varphi^{-1}(\varphi(F))$.

Двойная стрелка на рис. 5 из A_1 в A_2 означает, что реализованы $\varphi: A_1 \rightarrow A_2, \varphi^{-1}: A_2 \rightarrow A_1$, а стрелка из A_2 в A_3 – $\varphi: A_2 \rightarrow A_3$.

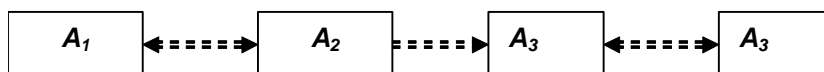


Рис. 5. Фрагмент структуры морфизмов, двухсторонними стрелками обозначены изоморфизмы

Алгебраические типы данных ТерМ

В результате проектирования алгебраических типов данных ТерМ получена следующая диаграмма наследования (рис. 6), морфизмов (рис. 7), расширений (рис. 8).

MultyRadical – алгебра данных (АД) $T_{\Omega}(A)$ с сигнатурой $\Omega = \{<, >, ==, +, -, *, /, ^\}$ и носителем $A = \{a + b\sqrt{c}; a \in MultyRadical; b, c \in Coef\}$. **MultyRadical** – кольцо радикалов над полем **Coef**.

Radical – АД $T_{\Omega}(A)$ с сигнатурой $\Omega = \{<, >, ==, +, -, *, /, ^\}$ и носителем $A = \{a + b\sqrt{c}; a, b, c \in Coef\}$. **Radical** – полугруппа с одним образующим элементом.

Complex – алгебра данных (АД) $T_{\Omega}(A)$ с сигнатурой $\Omega = \{<, >, ==, +, -, *, /, ^\}$ и носителем $A = \{a + i * b; a, b \in Coef\}$. **Complex** – поле над полем **Coef**.

Coef, **RatPolynom**, **RatMultyPolynom** – будут рассмотрены подробно на рис. 8.

Изоморфизм **Rational** и **PeriodNum** был описан выше. Их изоморфизм с **Real** – известен. **MultyPolinom** – описанный на рис.9.

RecPolynom – АД $T_{\Omega}(A)$ с сигнатурой $\Omega = \{<, >, ==, +, -, *, /, ^, NOD, NOK\}$ и носителем $A = \{a_n x^n + \dots + a_i x^i + \dots + a_0, x^i \in Degree, a_n, \dots, a_0 \in MultyPolynom\}$.

RecPolynom – кольцо многочленов от одной переменной над кольцом многочленов от нескольких переменных.

Variable – алгебра даних (АД) $T_{\Omega}(A)$ с сигнатурой $\Omega = \{<, >, ==\}$ и носителем $A = \{a, \dots, z, A, \dots, Z, \dots\}$. Эта АД используется для обозначения переменных, ее носитель состоит из больших и малых букв латинского алфавита, и этих букв с индексами.

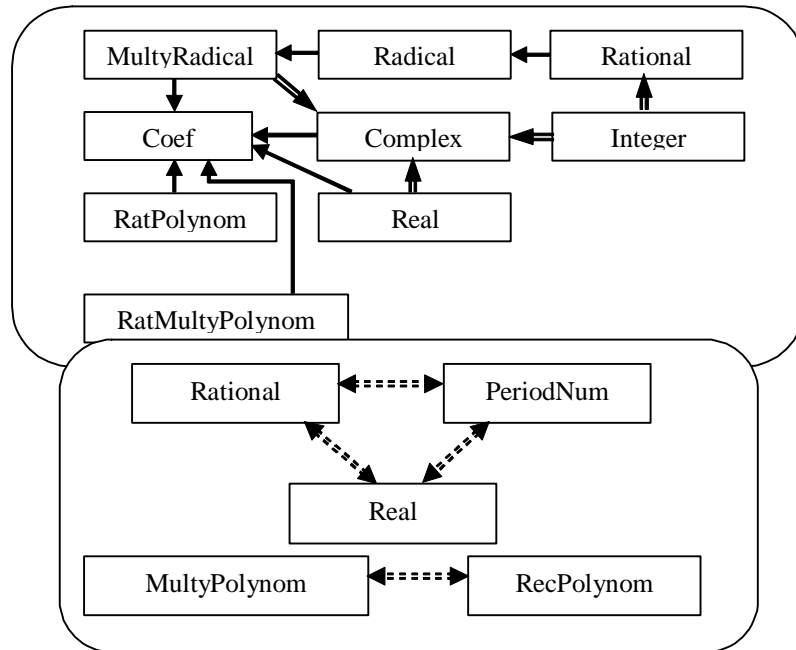


Рис. 7. Фрагмент диаграммы морфизмов МСА ТерМ

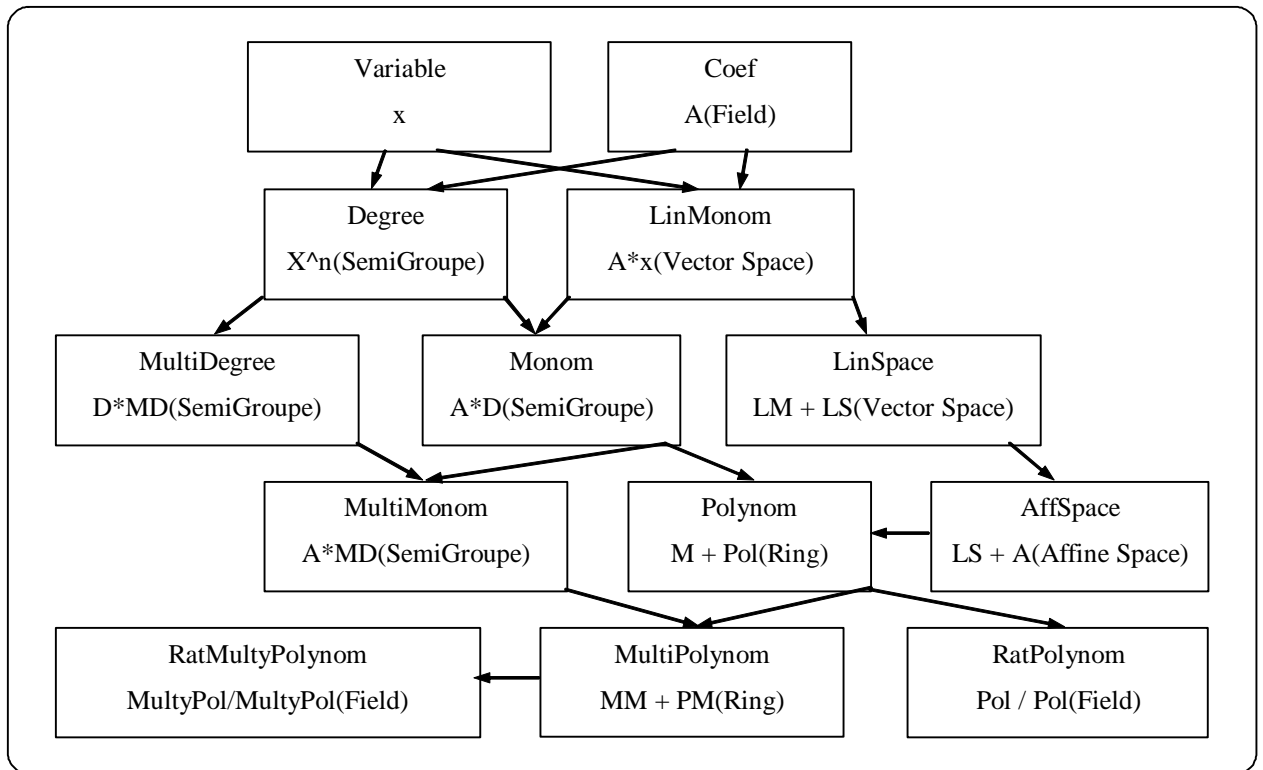


Рис. 8. Диаграмма расширений МСА ТерМ

Coef – АД $T_{\Omega}(A)$ с сигнатурой $\Omega = \{<, >, ==, +, -, *, /, NOD, NOK, ^\}$ и носителем A для которого могут быть определены такие операции. **Coef** – поле. Эта АД используется для обозначения коэффициентов МСА.

Degree – АД $T_{\Omega}(A)$ с расширенной сигнатурой $\Omega = \{<, >, ==, *, /, NOD, NOK, ^\}$ и носителем $A = \{v^{\wedge} c, v \in Variable, c \in Coef\}$. Degree – полугруппа с одним образующим элементом над полем **Coef**.

LinMonom – АД $T_{\Omega}(A)$ с сигнатурой $\Omega = \{<, >, ==, +, -, *, /\}$ (деление и умножение на число) и носителем $A = \{c * v, v \in Variable, c \in Coef\}$. LinMonom – одномерное векторное пространство над полем **Coef**.

MultyDegree – АД $T_{\Omega}(A)$ с сигнатурой $\Omega = \{<, >, ==, *, /, NOD, NOK, ^\}$ и носителем $A = \{d * a, d \in Degree, a \in A\}$. MultyDegree – расширение Degree по операции умножения, т.е. полугруппа степеней многих переменных над полем **Coef**.

Monom – АД $T_{\Omega}(A)$ с сигнатурой $\Omega = \{<, >, ==, +, -, *, /, NOD, NOK, ^\}$ и носителем $A = \{c * d, c \in Coef, d \in Degree\}$. Monom – расширение Degree по операции умножения, т.е. полугруппа мономов с одним образующим элементом над полем **Coef**.

LineSpace – АД $T_{\Omega}(A)$ с сигнатурой $\Omega = \{<, >, ==, +, -, *, /\}$ (сложение и умножение на число) и носителем $A = \{lm + ls, lm \in LinMonom, ls \in A\}$. LineSpace – расширение LinMonom по операции сложения, т.е. векторное пространство линейных комбинаций над полем **Coef**.

MultyMonom – АД $T_{\Omega}(A)$ с сигнатурой $\Omega = \{<, >, ==, *, /, NOD, NOK, ^\}$ и носителем $A = \{c * md, c \in Coef, md \in MultyDegree\}$. MultyMonom – расширение MultyDegree по операции умножения, т.е. полугруппа мономов многих переменных над полем **Coef**.

Polynom – АД $T_{\Omega}(A)$ с сигнатурой $\Omega = \{<, >, ==, +, -, *, /, NOD, NOK, ^\}$ и носителем $A = \{mn + pl, mn \in Monom, pl \in A\}$. Polynom – расширение Monom по операции сложения, т.е. кольцо многочленов одной переменной над полем **Coef**.

Affspace – АД $T_{\Omega}(A)$ с сигнатурой $\Omega = \{<, >, ==, +, -, *, /\}$ (умножение и деление на число) и носителем $A = \{ls + c, ls \in LineSpace, c \in Coef\}$. AffSpace – расширение LineSpace по операции сложения с числом, т.е. аффинное пространство линейных комбинаций со свободным членом над полем **Coef**.

MultyPolynom – АД $T_{\Omega}(A)$ с сигнатурой $\Omega = \{<, >, ==, +, -, *, /, NOD, NOK, ^\}$ и носителем $A = \{mm + mp, mm \in MultyMonom, mp \in MultyPolinom\}$. MultyPolynom – расширение MultyMonom по операции сложения, т.е. кольцо многочленов многих переменных над полем **Coef**.

RatPolynom – АД $T_{\Omega}(A)$ с сигнатурой $\Omega = \{<, >, ==, +, -, *, /, NOD, NOK, ^\}$ и носителем $A = \{x / y; x, y \in Polinom; GCD(x, y) = 1\}$. RatPolynom – расширение Polynom по операции деления, т.е. поле рациональных функций от одной переменной над полем **Coef**.

RatMultyPolynom – АД $T_{\Omega}(A)$ с сигнатурой $\Omega = \{<, >, ==, +, -, *, /, NOD, NOK, ^\}$ и носителем $A = \{x / y; x, y \in MultyPolinom; GCD(x, y) = 1\}$. RatMultyPolynom – расширение MultyPolinom по операции деления, т.е. поле рациональных функций многих переменных над полем **Coef**.

Реализация многосортной алгебры, выполненная в соответствии с диаграммами (рис. 6, рис. 7, рис. 8) позволила полностью реализовать стандартные алгоритмы символьных вычислений в классической алгебре. Однако эта реализация имеет недостатки: во-первых, рекурсивный спуск при вычислениях «проходит» около 10 систем переписываний; во-вторых, в тех случаях, когда схемы расширений алгебр подобны, желательно использовать методы параметрического программирования, что не предусмотрено диаграммой расширений (рис. 8).

Первый недостаток приводит к неоправданно большим затратам времени и памяти. Второй недостаток приводит к тому, что вместо того, чтобы повторно использовать код, приходится фактически копировались фрагменты уже написанного исходного кода, модифицировать и вставлять их в другое место программы.

Отметим, что эти проблемы можно решить средствами инсерционного программирования [4] и перенести на нижний уровень только числовые алгебры. В данной версии, однако, использовалась техника параметрического программирования. После прототипирования и определения основных свойств нижнем уровне были реализованы алгебры данных, указанные на рис. 9.

Алгебры данных **Variable**, **Coef**, **Degree**, **MultyDegree** – остались содержательно такими же, как и на рис. 7.

Factor – АД $T_{\Omega}(A)$ с сигнатурой $\Omega = \{<, >, ==, *, /, NOD, NOK, ^\}$ и носителем $A = \{c / x; c \in Coef; x \in Variable \cup Degree \cup MComponent\}$. Factor – расширение Degree, MComponent по операции умножения, т.е. полугруппа над полем **Coef**.

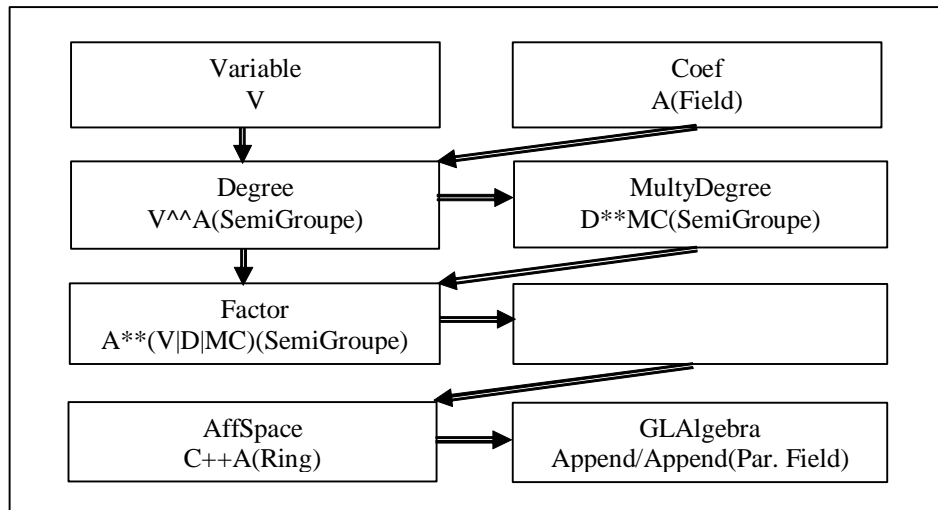


Рис. 9. Новая диаграмма расширений MCA ТерМ

LinSpace – АД $T_{\Omega}(A)$ с расширенной сигнатурой $\Omega = \{<, >, ==, +, -, *, /, ^{NOD}, NOK\}$ (умножение и деление на число) и носителем $A = \{f + c; f \in Factor; c \in A\}$. LinSpace – расширение Factor по операции сложения, т.е. кольцо над полем **Coef**.

AffSpace – АД $T_{\Omega}(A)$ с расширенной сигнатурой $\Omega = \{<, >, ==, +, -, *, /, ^, NOD, NOK\}$ (умножение и деление на число) и носителем $A = \{cp + c, cp \in Cpmponent, c \in Coef\}$. AffSpace – расширение Component по операции сложения, т.е. кольцо со свободным членом над полем **Coef**

GLAlgebra – АД $T_{\Omega}(A)$ с сигнатурой $\Omega = \{<, >, ==, +, -, *, /, NOD, NOK, ^\}$ и носителем $A = \{x / y; x, y \in Append, GCD(x, y) = 1\}$. RatPolynom – расширение Polynom по операции деления, т.е. поле частных над полем **Coef**.

Используя конкретное значение правой части в определении носителя Factor, несложно установить гомоморфизм между соответствующими алгебрами рис. 8 и рис. 9.

Поле такой реализации для проверки оптимальности перепроектированной диаграммы расширений MCA ТерМ, был поставлен вычислительный эксперимент. Небольшая часть результатов представлена в таблице.

Таблица. Результаты вычислительного эксперимента MCA ТерМ

	Пример	Время вычисления	Используемая память
Рис. 9	$(x + y + z)^{10}$	38 с	121 Мб
Рис. 10		3 с	4 Мб
Рис. 9	Время загрузки	10 с	32 Мб
Рис. 10	ядра системы	2 с	1 Мб

Таким образом, реализация на нижнем уровне расширений MCA системы решила вышеописанные проблемы.

Заключение

Использование понятий наследования, расширения и морфизмов приводит к качественной реализации таких многосортных алгебр, при условии оптимального проектирования расширений.

Построенные многосортные алгебры для ТерМ являются достаточно быстрой реализацией алгоритмов компьютерной алгебры и могут быть использованы для построения коммерческих программных продуктов в данной области знаний. Дальнейшее развитие ТерМ предполагает расширение поддерживаемых многосортных алгебр данных и повышение эффективности процедур интерпретации.

Опыт и традиции разработки приложений с использованием APS накапливаются и используются в Институте кибернетики им. В.М. Глушкова НАН Украины, Херсонском государственном университете и в других научных заведениях.

Выражаю благодарность научным руководителям А.А. Летичевскому, М.С. Львову, за постоянный интерес, руководство в проектировании ТерМ, адаптации APS и ценные замечания по содержанию данной работы.

1. Львов М.С. ТерМ VII – шкільна система комп’ютерної алгебри. Комп’ютер у школі та сім’ї. – 2004. – № 7. – С. 27–30.
2. Математическая логика в програм-мировании: Сб. статей М34 1980-1988 / Пер. с англ. – М.: Мир, 1991. – 408 с.
3. Kapitonova U., Letichevsky A., Volkov V., Lvov M. Tools for solving problems in the scope of algebraic programming. Lectures Notes in Computer Sciences. – 1995 –№ 958.. – PP. 31–46.

4. *Капитонова Ю.В., Летичевский А.А., Волков В.А.*. Дедуктивные средства системы алгебраического программирования // Кибернетика и системный анализ, 2000, 1, С. 17–35.
5. *Volkov V., Kuprienko A., Lvon M.* Applied Computer Support of Mathematical Training. Proc. of Internal Work Shop in Computer Algebra Applications, Kiev., 1993. – PP. 25–26.
6. *Lvon M.* AIST: Applied Computer Algebra System. Proc. of ICSTE'93. Kiev. – pp. 25–26.
7. *Песчаненко В.С.* Розширення стандартних модулів системи алгебраїчного програмування APS для використання у системах навчального призначення. Науковий часопис НПУ ім. МП Драгоманова Сер. № 2. Комп'ютерно-орієнтовні системи навчання –К.: НПУ ім. М.П. Драгоманова. 2005 – №3(10).