Interactive problem solving environment *T-algebra*

Rein Prank
University of Tartu
Institute of Computer Science
rein.prank@ut.ee

What is T-algebra?

T-algebra is a Tiger Leap project (2004-…) for creating an interactive problem solving environment for Basic School algebra:

2) calculation of the values of numerical expressions
3) operations with fractions
4) solving of linear equations, inequalities and linear equation systems
5) operations with monomials and polynomials

Team

- Working group:

  R.Prank – teamleader
  M.Issakova, D.Lepp, V.Vaiksaar – authors of chapters
  E.Tõnisson, P.Luik, T.Lasn,
  Mart Oja, Maire Oja
  also: T.Lepmann, A.Palu, H.Jukk, K.Kokk

Problem types (51). Fractions

Problem types are taken directly from the textbooks

3) reduction of fractions
4) extension of fractions
5) conversion of fractions to common denominator
6) comparing fractions
7) addition and subtraction of fractions with common denominator
8) conversion of improper fraction to mixed number
9) addition and subtraction of fractions with different denominators
10) addition and subtraction of mixed numbers
11) conversion of decimal fraction to common fraction
12) conversion of common fraction to decimal fraction
13) decimal approximation of common fraction
14) multiplication of fractions
15) finding reciprocal value
16) division of fractions

Motivation (1)

When the student solves an expression manipulation task, he should **at each solution step**:

2. **choose a transformation rule** corresponding to a certain operation in the algorithm (or some simplification or calculation rule known earlier),
3. **select the operands** (certain parts of expressions or equations) for this rule,
4. **replace them** with the result of the operation.

Some more “creative” tasks (such as factorization or integration) are taught in less algorithmic style but the solutions are expected to consist of steps of the same structure.

Motivation (2)

For proper learning of expression manipulation as well as for assessment and diagnosis of knowledge gaps, **we need an environment where**

all the decisions and calculations would be made by the student,

the environment would be able to understand the mistakes and give feedback.

What sorts of environments are available? (1)

Rule-based environments (Mathpert, EGPY (Stanford); CAS). Student selects (subexpression and) rule, computer applies the rule.

* MathXpert (Michael Beeson)
http://www.helpwithmath.com/
1) Student marks a subexpression,
2) Program displays the menu with rules that are applicable to (some parts of) this subexpression,
3) Student selects a rule and the program applies this rule

Student's learning is passive.

Many mistakes are impossible.

What sorts of environments are available? (2)

Input-based environments (Aplusix).

APLUSIX
http://aplusix.imag.fr/
Student enters the solution line by line,
computer checks the equivalence and measures how well the subgoals are satisfied.

It is very hard to diagnose the errors more precisely than “not equivalent”

Prototype in Tartu

Formula Manipulation Assistant (H.Viira-Tamm, 1989-91)

- For Propositional Logic (expression of formulas through $\{\neg,\&\}$, $\{\neg,\vee\}$, $\{\neg,\supset\}$; DNF), MS DOS text mode
- Object-Action scheme:
  1) marking a subformula,
  2) input / selection of rule from menu
- Program checks:
  syntactical correctness of subformula marking,
  equivalence/rule-suitability,
  reaching of goal

Java version for Propositional and Predicate Logic is written in 2003 (V.Vaiksaar)

T-algebra environment

T-algebra uses Action-Object-Input scheme:
each solution step consists of **three sub-steps**:

1) selection of the operation from the menu,
2) marking the operand(s) in expression,
3) entering the result of the operation.

- T-algebra requires precise selection of operands
- First two substeps are “mixed together”

Is A-O-I-scheme very expensive?

(in comparison with pure-input interface)

- A = one selection in menu
- I (=Input of changed part) - is necessary in pure-input too
- But O (= marking of operands) is necessary for :
  - a) Copy whole line+(mark operands+Delete)+I,
  - b) Copy whole line +change/delete operands,
  - c) (Mark and copy unchanged parts) + I,
  - d) Input of whole line

In general our scheme requires the same amount of input as pure-input scheme!

Input modes in T-algebra

For entering the result the program has **three input modes**:

2. **Free,** $9ab^2 \boxed{-4a^2b} + 2ab - (5ab^2) - 2ab + abb$

4. **Structured,** $9ab^2 \boxed{-} \boxed{4} \boxed{a}^{\boxed{2}} \boxed{b}^{\boxed{}} + 2ab - (5ab^2) - 2ab + abb$

6. **Partial** $9ab^2 \boxed{-} \boxed{4} a^2b + 2ab - (5ab^2) - 2ab + abb$

Input mode for the task is fixed in task file

What can we diagnose in A-O-I-interface? (1)

Application of selected operation

- is impossible
- does not correspond to the algorithm

T-algebra uses first when Hint is asked

What can we diagnose in A-O-I-interface? (2)

Selection of operands

- Marked term is not a syntactically correct expression,
- Marked term is not a proper subexpression (order of operations misunderstood),
- Marked term has not the form required for selected rule,
- Operands do not satisfy the compatibility requirements (are not like terms, etc.),
- Operands do not satisfy the location requirements (do not belong the same sum, fraction, product etc.)

4),5) => parallel conversions not allowed

What can we diagnose in A-O-I-interface? (3)

Entered subexpression (result of conversion)

- is not a syntactically correct expression,
- should be preceded by a sign,
- should be put in parentheses (order of operations),
- has not the structure required for
  selected-rule (marked-operands) =…
  (not a monomial, wrong number of members etc),
- is not equivalent with marked part,

- Concrete parts of Input do not have right value
- Selected operation with marked terms is not performed (nothing reduced, terms are not moved to other side, …)

Some problems (1)

For clearer diagnosis we want the conversions to consist of only one application of the rule:

- No multi-rule conversions
- No parallel conversions

But nevertheless:

- How much pre-processing to accept by marking?
- How much post-processing to accept by input?

Probably we will have standards for this in year 20??

Conclusions

Three-stage dialogue for expression manipulation

- Is intuitively understandable for the students
- Requires the same amount of keyboard/mouse work as pure input
- Provides the program with information about intentions of the student
- Allows to point in error messages to real places of mistakes

Requires

- Standardization of understanding the rules
- Separation of mathematical errors and conflicts with customs of concrete program