Design and Educational Applications of a Generic Step-by-Step Solver for Mathematical Problems Based on *Mathematica*

Bernhard Zgraggen

Mathematics and Engineering Sciences
Swiss Distance University of Applied Sciences
bzgraggen@fernfachhochschule.ch

1 Abstract

The starting point of this paper is a project at the Distance University of Applied Sciences of Switzerland called ‚*webSolutions'* which deals with the development and deployment of programs interactively generating detailed and dynamic step-by-step solutions to typical problems in higher mathematical education on the basis of *Wolfram Research, Inc.'s Mathematica*. The first part of the paper explains an approach for a generic step-by-step solver and its interface enabling tutors to create step-by-step solutions according to their mathematical context by an appropriate setting of parameters (mathematical rules, complexity parameters, parameters for solutions, general stop conditions, rules for output and display of expressions and mathematical explanations). The functioning of such an algorithm and its parameters are explained in detail on the basis of examples from distinct mathematical areas. The second part of the paper puts the emphasis on web-based applications of such a solver system in higher mathematical education.

2 Didactic Background

2.1 Problems with textbooks

A lot of mathematical textbooks contain disadvantages when beeing used in a distance educational environment. Among other unfavourable things we often find incompletely explained solutions and the static nature of prime examples, exercises and corresponding solutions. Furthermore, electronic reprocessing of parts of mathematical textbooks (e.g. in form of attachments to an electronic mail or spontaneous presentations to an audience) cannot be prepared or carried out easily within an appropiate space of time.

The impossibility to present complete solutions to more difficult exercises at home or in the classroom within an appropriate stretch of time is often complained by tutors and lecturers at the Distance University of Applied Sciences of Switzerland. Another complaint concerns the nature of many distance learners or the nature of the distance education system, respectively: Experience has shown that distance learners behave like distinctive individualists. So it is pretty normal that supporting students by electronic mail or within a web-based learning management system leads to redundancies as far as the questions and the answers are concerned. Forum discussions and group mails only partially alleviate these redundancies.

For technical, syntactical and semantic reasons it is typically difficult for students in a distance learning environment to ask specific questions concerning an actual or imagined mathematical problem. On the other side tutors and lecturers often are confronted with technical and syntactical difficulties when trying to answer mathematical questions by electronic means.
Successful applications of the current tools of *webSolutions* [1,2,6] require that students have prepared the theory corresponding to a problem area so far that they are familiar with the corresponding problems, mathematical terms and formalism.
When these conditions are fulfilled the current tools of *webSolutions* may help to reach the following goals: **1)** students learn and practise applying structured methods of solutions on the basis of concrete problems, **2)** students control and correct their own handwritten solutions to exercises, **3)** students, lecturers and tutors present, reprocess and communicate complex solutions within an appropiate stretch of time.

2.2 Advantages and disadvantages

The educational advantages of an application of the current tools can be summarized as follows:
1) Students and tutors have the possibility to receive step-by-step solutions to mathematical problems in their browsers, which probably are more structured, detailed and dynamic than solutions in ordinary textbooks. Thus the tools may help to defuse the shortages of solutions in textbooks and time in the classroom.
2) Students and tutors are not restricted to the exercises in their textbooks; the tools were designed to be flexible in order to be combinable with a wide range of teaching materials.
3) Accessing the tools is easy and extensively independent of the platforms and browsers of its users. The tools are based on a technology which supplies interfaces to the most important Internet technologies. Thus it is very easy to combine the tools, e.g., with electronic platforms.
4) The tools may support tutors when preparing lessons, presenting mathematical topics and coaching their students.

Of course, the currently published tools have many disadvantages. The most important among them are:
1) Creating and testing programs computing dynamic step-by-step solutions within *Mathematica* takes a lot of time.
2) Sometimes computed step-by-step solutions may be rather complex - although mathematically correct. Such results may normally confuse and discourage students.
3) The level of interactivity is quite low.
4) Tutors do not have possibilities to add their own material, to modify the existing material or to manipulate the program run fundamentally.

2.3 Improvements

There are, naturally, at least two points of view when thinking about possibilities for improvements: The student's view and the tutor's view.
From the student's viewpoint, to our opinion, the most important things to do are to raise the level of interactivity and to improve the user-friendliness. These goals can be reached, e.g., by **1)** allowing students to propose rules or interim results between the steps of the solution (e.g. certain derivation rule and the corresponding interim functional expression while trying to find the derivation of a function), **2)** enabling back - and forward browsing through the separate steps of the solution, and **3)** providing dynamically more information to certain interim results or terms when

desired (e.g. step-by-step explanations on how the zeros of the partial derivatives are found when considering an optimizaton problem in several variables).
An interactive application of Wolfram Research, Inc's software *webMathematica* [3] for elementary algebraic calculations was developped by Hitoshi Nishizawa [4].

From the tutor's viewpoint - in addition to the suggestions above - the most interesting improvements concern possibilities of adding own material, modifying the existing material and manipulating the functioning of the programs. These improvements may be realized by developping a kind of a generic algorithm allowing tutors **1)** to add mathematical transformation rules, formatting rules, output comments and further functionality to the programs, **2)** to modify transformation rules or to influence the foreground sequence of the rules and **3)** to set parameters controlling the program run.

3 Design of a generic Problem Solver

3.1 Introduction

A lot of mathematical problems can be solved using transformation rules. By repeated and careful application of such rules a mathematical expression is transformed until it fulfills a specific condition or matches a specific form. Typical examples for such problems are solving a system of linear equations or solving a linear optimization problem, finding a limit of a real function, computing the derivative or the integral of a function and simplifying an expression in a specific algebraic context.
Of course, *Mathematica* contains a large number of efficient and ingenious, generic algorithms (like ‚Solve' or ‚FullSimplify'), based on an extensive and cleverly devised collection of transformation rules. As far as we know, designs and implementations of these algorithms and the underlying collections of transformation rules are not public.
In this section we would like to show design and implementation possibilities for a parametrizable procedure computing and representing detailed step-by-step solutions on the basis of existing algorithms of *Mathematica*. By providing parameters for transformation rules, additional rule-conditions and rules for formatting the output as well as providing parameters allowing to influence the program run the procedure should be usable in different mathematical and didactic contexts.

3.2 First design

A first design for a generic algorithm for step-by-step solutions of rule-based mathematical problems is described in [5]. The algorithmic design corresponds to a combination of branch-and-bound and breadth-first searching strategies in the set of expressions computed by systematic application of the transformation rules. All computed expressions are collected and processed in a set (no duplicates) using efficient search and insert functions. Structural operations on mathematical expressions (interpreted as trees), pattern matching, reordering of transformations rules based on parametrizable ordering functions, additional conditions on transformation rules and proposed solutions, pre-computed solutions and bounds as computation time, depth and breadth of search tree, the number of computed expressions as well as general complexity functions measuring complexity of expressions and rules or bounding the complexity of expressions play an important role.

3.3 Parameters

3.3.1 General stop conditions

These conditions are determined by four parameters bounding the depth, breadth, number of computed expressions and computing time (seconds) of searching:

Example:

```
GeneralStopConditions := {Infinity, Infinity, Infinity, Infinity};
```

3.3.2 Prefunctions and postfunctions

The purpose of prefunctions or postfunctions is to do things before or after the beginning of the problem solving procedure, respectively. Like parsing or restructuring input expressions and setting attributes for mathematical operators. In the example below there is one prefunction defined to eliminate any associative structures for logical conjunctions and disjunctions.

Example:

```
PreFunctions :=
  {
   Module[{UnFlatteningRules},
    ClearAttributes[{And, Or}, Flat];
    UnFlatteningRules =
     {f_ || g_ || h_ || x___ :> ((f || g) || h) || x,
      f_ && g_ && h_ && x___ :> ((f && g) && h) && x,
      (f_ || g_ || h_) && x___ :> ((f || g) || h) && x,
      f_ && g_ && h_ || x___ :> ((f && g) && h) || x};

    ProblemExpression = ProblemExpression //. UnFlatteningRules;

   ]
  };

PostFunctions := {};
```

3.3.3 Complexity functions

The complexity functions are described by four functional parameters for measuring simplicity of expressions, ordering the transformation rules and bounding the complexity of expressions during the program run. In the example below the function ‚LeafCount' counts the number of leaves in a mathematical expression represented as a tree.

Example:

```
ComplexityFunctions := {LeafCount, LeafCount, LeafCount, (True) &};
```

3.3.4 Functions for possible solutions

These functions are described by four functional parameters for the computation of a proposal solution, defining conditions for expressions to be accepted as solutions, comparing expressions to the proposed solution and comparing expressions among themselves.

Example:

```
ProposedSolutions := {
  {(Module[{output},
      output = FullSimplify[#, ComplexityFunction → ComplexityFunctions[[1]]];
      ClearAttributes[{And, Or}, Flat];
      output = output //. {f_ || g_ || h_ || x___ :→ ((f || g) || h) || x,
         f_ && g_ && h_ && x___ :→ ((f && g) && h) && x,
         (f_ || g_ || h_) && x___ :→ ((f || g) || h) && x,
         f_ && g_ && h_ || x___ :→ ((f && g) && h) || x};
      output
     ]) &},

  {}, {SameQ }, {SameQ}
 };
```

3.3.5 Extended transformation rules

This parameter consists of a list of extendend transformation rules also containing additional conditions on rules, rule names and explanations. Below only two items are shown (distribution rules from Boolean Algebra):

Example:

```
{{(f_ && (g_ || h_)) :→ ((f && g) || (f && h))}, {Hold[g === ! f || h === ! f]},
 {"Distribution"}, {"a ∧ (b ∨ c) = (a ∧ b) ∨ (a ∧ c)"}},
{{(f_ || (g_ && h_)) :→ ((f || g) && (f || h))}, {Hold[g === ! f || h === ! f]},
 {"Distribution"}, {"a ∨ (b ∧ c) = (a ∨ b) ∧ (a ∨ c)"}}
```

3.3.6 Display rules for transformation rules and expressions

These parameters determine the mathematical notation and style of rules and expressions when displayed to the user.

Examples:

```
RuleFormattingRules :=
  {{f → a, g → b, h → c}, {Or[x_, y_] :→ TraditionalForm[x || y],
    And[x_, y_] :→ TraditionalForm[x && y], Not[x_] :→ TraditionalForm[! x]}};
```

```
ExpressionFormattingRules :=
  {{Or[x_, y_] :→ TraditionalForm[x || y], And[x_, y_] :→ TraditionalForm[x && y],
    Not[x_] :→ TraditionalForm[! x]}};
```

3.4 Problem expression and program run (example)

The example below arises from the mathematical problem of simplification of Boolean expressions.

Example:

```
ProblemExpression = ((a || (b || c)) && (c || ! a))
StepByStepSolution

{! a && b || c, {0},
 {! a && b || c,
  {{20, {1}, c || (a || b)}, {4, {}, c || (a || b) && ! a}, {19, {2}, ! a && (a || b)},
   {23, {2}, ! a && a || ! a && b}, {12, {2, 1}, False}, {20, {}, ! a && b || c}}
 }, 64}
```

Comments:

The first entry in the list `(!a && b || c)` represents the solution found by the mathematical expert system (*Mathematica*). It was computed using the first parameter of ‚ProposedSolutions' (cf. 3.3.4).

The second entry `{0}` encodes that the step-by-step-algorithm found a solution which matches the proposed solution.

The third entry is a list consisting of the single steps of the solution, e.g. the first part of this list is equal to `{20, {1}, c || (a || b)}`. This means that the first rule applied is the rule with number 20 (commutative rule), this rule was applied at position `{1}` in the expression tree and the result of this application yields the partial expression `c || (a || b)` (at position `{1}`).

The fourth entry (64) indicates the number of computed expressions during the program run.

4 Educational Applications

4.1 Styled display of transformed partial expressions for simplifying Boolean expressions

This example shows how the result of StepbyStepSolution may be displayed with styles (transformed partial expressions are bold-printed) and explanations.

The following initial expression was entered:

$((a \lor b) \lor c) \land (c \lor \neg a)$

The Mathematica system proposes the following solution to your problem:

$\neg a \wedge b \vee c$

```
The Step-by-Step Solver terminated searching with the following state(s):
```

Proposed solution found .

```
The Step-by-Step Solver found the following optimal solution
by applying 6 transformation rules:
```

$\neg a \wedge b \vee c$

```
Here are the 6 steps leading to the optimal solution:
```

Commutativity: The order of expressions in disjunctions has no influence.

$((\mathbf{a} \vee \mathbf{b}) \vee \mathbf{c}) \wedge (c \vee \neg a) = (\mathbf{c} \vee (\mathbf{a} \vee \mathbf{b})) \wedge (c \vee \neg a)$

Distribution: $(a \vee b) \wedge (a \vee c) = a \vee (b \wedge c)$

$(\mathbf{c} \vee (\mathbf{a} \vee \mathbf{b})) \wedge (\mathbf{c} \vee \neg \mathbf{a}) = \mathbf{c} \vee (\mathbf{a} \vee \mathbf{b}) \wedge \neg \mathbf{a}$

Commutativity: The order of expressions in conjunctions has no influence.

$c \vee (\mathbf{a} \vee \mathbf{b}) \wedge \neg \mathbf{a} = c \vee \neg \mathbf{a} \wedge (\mathbf{a} \vee \mathbf{b})$

Distribution: $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$

$c \vee \neg \mathbf{a} \wedge (\mathbf{a} \vee \mathbf{b}) = c \vee (\neg \mathbf{a} \wedge \mathbf{a} \vee \neg \mathbf{a} \wedge \mathbf{b})$

Contradiction: Conjunction of a boolean expression with its negation yields false.

$c \vee (\neg \mathbf{a} \wedge \mathbf{a} \vee \neg a \wedge b) = c \vee (\mathbf{False} \vee \neg a \wedge b)$

Commutativity: The order of expressions in disjunctions has no influence.

$\mathbf{c} \vee \neg \mathbf{a} \wedge \mathbf{b} = \neg \mathbf{a} \wedge \mathbf{b} \vee \mathbf{c}$

4.2 Computing partial derivatives (styled output with partial expressions)

In this example the module StepByStepSolution is parametrized for computing partial derivatives. Again transformed partial expressions are boldprinted.

```
ProblemExpression = {h[x, f[x]], x};
StepByStepSolution
OutputModuleUnColoured[%] // TableForm
```

{f′[x] h^(0,1)[x, f[x]] + h^(1,0)[x, f[x]], {0}, {f′[x] h^(0,1)[x, f[x]] + h^(1,0)[x, f[x]],
{{16, {}, {h^(1,0)[x, f[x]], h^(0,1)[x, f[x]]}.d[{x, f[x]}, x]},
{15, {2}, {d[x, x], d[f[x], x]}}, {2, {2, 1}, 1}, {4, {1, 1}, f′[x]}}}, 6}

The following initial expression was entered:

$$\frac{\partial}{\partial x}(h[x, f[x]])$$

The Mathematica system proposes the following solution to your problem:

$$f'[x]\, h^{(0,1)}[x, f[x]] + h^{(1,0)}[x, f[x]]$$

The Step-by-Step Solver terminated searching with the following state(s):

Proposed solution found .

The Step-by-Step Solver found the following optimal solution by applying 4 transformation rules:

$$f'[x]\, h^{(0,1)}[x, f[x]] + h^{(1,0)}[x, f[x]]$$

Here are the 4 steps leading to the optimal solution:

Multi-dimensional chain rule: f[g]' is equal to f'[g] times g'.

$$\frac{\partial}{\partial x}(h[x, f[x]]) = \{h^{(1,0)}[x, f[x]], h^{(0,1)}[x, f[x]]\}.\frac{\partial}{\partial x}(\{x, f[x]\})$$

Vector derivative rule: Partial derivative of a vector is computed by partial derivation of its components.

$$\{h^{(1,0)}[x, f[x]], h^{(0,1)}[x, f[x]]\}.\frac{\partial}{\partial x}(\{x, f[x]\}) =$$
$$\{h^{(1,0)}[x, f[x]], h^{(0,1)}[x, f[x]]\}.\left\{\frac{\partial}{\partial x}(x), \frac{\partial}{\partial x}(f[x])\right\}$$

Identity rule: The derivative of the identity function is the constant 1.

$$\frac{\partial}{\partial x}(f[x])\, h^{(0,1)}[x, f[x]] + \frac{\partial}{\partial x}(x)\, h^{(1,0)}[x, f[x]] =$$
$$\frac{\partial}{\partial x}(f[x])\, h^{(0,1)}[x, f[x]] + 1\, h^{(1,0)}[x, f[x]]$$

Special function rule: Derivative known from data base.

$$\frac{\partial}{\partial x}(f[x])\, h^{(0,1)}[x, f[x]] + h^{(1,0)}[x, f[x]] = f'[x]\, h^{(0,1)}[x, f[x]] + h^{(1,0)}[x, f[x]]$$

4.3 Computing limits (web-Interface)

The figure below shows a web-Interface for the computation of limits. Here the problem was to compute the limit of Sin[r^2]/r^2 if r tends towards 0. Only the results are shown.

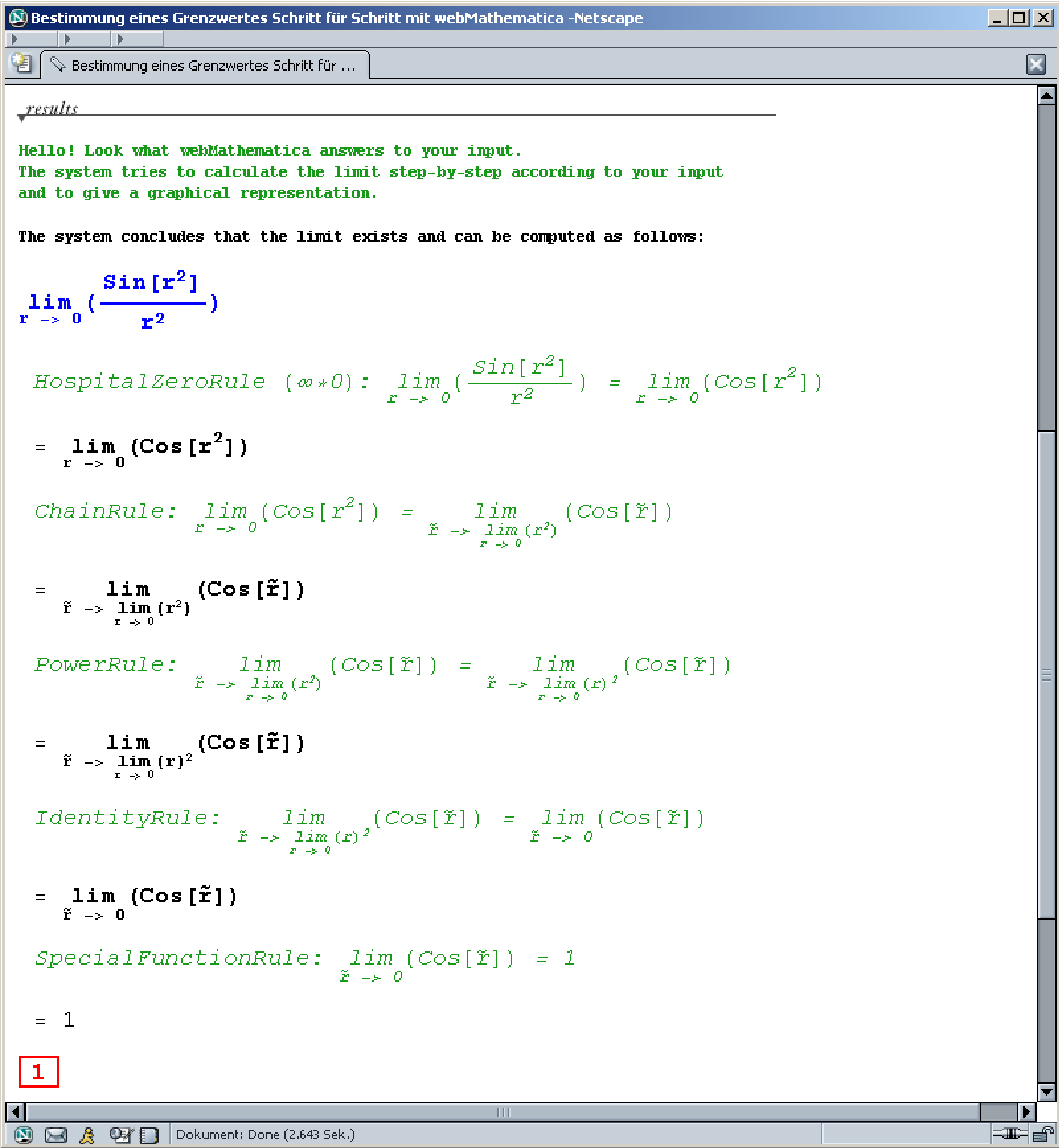

Figure 1: Results of the computation of the limit of Sin[r^2]/r^2 if r tends towards 0.

5 References

1 Project Site:
http://www.fernfachhochschule.ch/zgraggen/webmath/webmathematica-ffhs-publicity.html

2 Wolfram Research, Inc's Featured Project Site:
http://www.wolfram.com/products/webmathematica/examples/others.html

3 Wolfram Research, Inc.: WebMathematica Product Information.
http://www.wolfram.com/products/webmathematica/

4 Nishizawa, Kajiwara, Yoshioka (2003): *A tutoring system of symbolic Calculations supported by webMathematica*. Proceedings of the 5th International Mathematica Symposium IMS, p. 167 – 174, July 2003, London.

5 Zgraggen (2003). *Education, Design and Implementation aspects of a generic step-by-step solver based on Mathematica*. Proceedings of the 5th International Mathematica Symposium IMS, p. 199-206, July 2003, London.

6 Zgraggen (2003). *Education and design aspects of a generic step-by-step solver for mathematical problems in the World Wide Web*. Proc. 3rd ICTTL, Heidelberg, Germany, p. 339 – 344, July 2003.

7 Stephen Wolfram (1999). *The Mathematica Book*. 4th Edition. Wolfram Media, Inc.