

# Studenten Projekte im ISAC-Projekt

für Master, Projekt/Seminare, Bakalaureat

**Mitarbeit an einer kommenden Generation von Mathematik Assistenten (MAs)** ist das Ziel des ISAC-Projektes an österreichischen Universitäten. Die kommende Generation wird auf Theorem-Prover (TP) Technologie aufbauen und folgende Merkmale aufweisen:

1. *Alles* Mathematik-Wissen ist in lesbarem Format (*kein* Programm Code), z.B. Isabelle <http://isabelle.in.tum.de/dist/library/HOL/Rings.html> verwendet von ISAC<sup>1</sup>.
2. TP-basierte MAs decken den gesamten Bereich von “mathematischem Problemlösen” ab, von Spezifikation bis (automatischem!) Beweis der Postcondition.
3. Des weiten Bereiches wegen können TP-basierte MAs *schrittweises* Problemlösen modellieren — eingegebene Schritte werden *automatisch* überprüft.
4. Kombination von “deduction and computation” erzeugt “next-step-guidance”: das system kann einen nächsten Schritt vorschlagen, wenn Studierende stecken bleiben.

Diese Merkmale führen zu flexiblen Systemen, die die Ingenieurs-Ausbildung verändern werden. Zum Beispiel, Abbildung 1<sup>2</sup> zeigt ein-und-die-selbe Rechnung wie Abbildung 2 auf Seit 2 — das System passt sich den Interaktionen der User an, einmal an IngenieurInnen, ein andermal an Mathematik-Lernende:

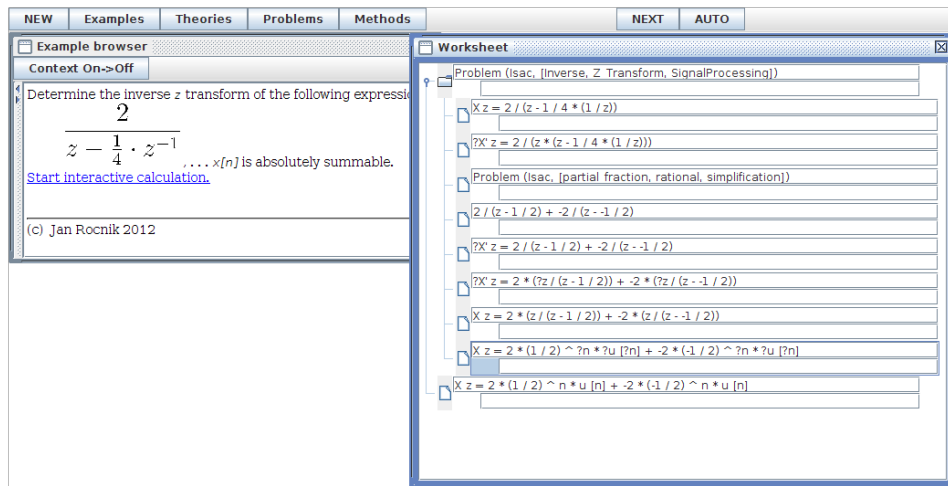


Abbildung 1: Sicht des Ingenieurs: inverse Z Transformation

**Die umseitig ausgeschriebenen Projekt-Themen** richten sich an Mathematik-Studenten, die an funktionaler Programmierung interessiert sind (ISAC verwendet Standard ML<sup>3</sup>). Das Ziel aller Themen ist ein “Re-Engineering von Computer Algebra”, das das Lösen von mathematischen Grundaufgaben interaktiv nachvollziehbar macht — und beim Implementieren tiefe Einblicke in Computer Mathematik vermittelt.

<sup>1</sup>[http://www.ist.tugraz.at/isac/index.php/Main\\_Page](http://www.ist.tugraz.at/isac/index.php/Main_Page) und <http://www.ist.tugraz.at/projects/isac/>

<sup>2</sup>bessere Bilder auf [http://www.ist.tugraz.at/isac/index.php/Development\\_Computer\\_Mathematics](http://www.ist.tugraz.at/isac/index.php/Development_Computer_Mathematics)

<sup>3</sup>[http://en.wikipedia.org/wiki/Standard\\_ML](http://en.wikipedia.org/wiki/Standard_ML)

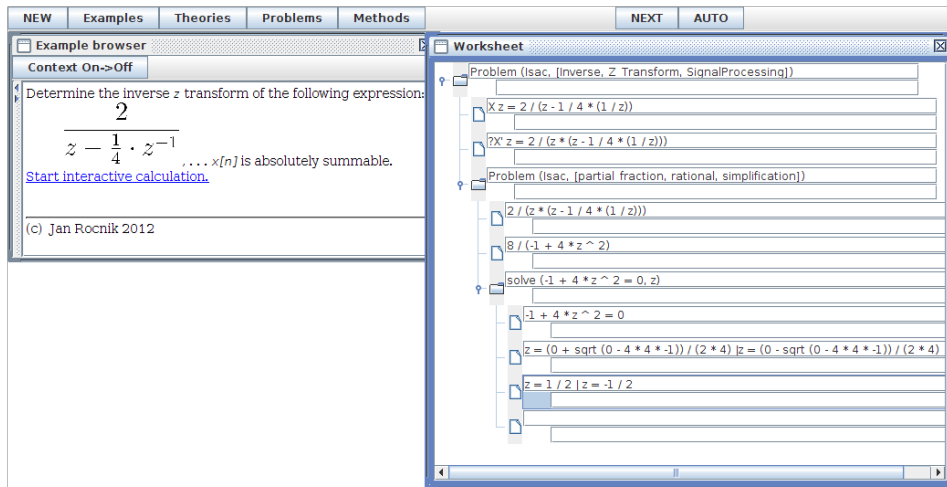


Abbildung 2: Sicht beim Mathematik-Lernen: Partialbruch-Zerlegung

**Algebraische Simplifikation in verschiedenen Bereichen** ist eine grundlegende Funktionalität von Computer Algebra; Simplifikation macht Ausdrücke “einfacher”:

$$\frac{2 * a + 3 * b}{b * c} + \frac{3 * c + a}{a * c} - \frac{2 * a^2 + 3 * b * c}{a * b * c} = \frac{4}{c}$$

*ISAC*'s Simplifier ist ausreichend für Unterrichtsversuche (ganze und rationale Ausdrücke), das “Kürzen von rationalen Polynomen” ist in Arbeit am RISC.

Das Ziel dieses Sub-Projektes ist, das Kürzen in den Simplifier einzubetten, den Simplifier allgemeiner zu machen (u.a. auf komplexe Zahlen auszuweiten) und auch spezielle Simplifier für *ISAC*'s Equation-Solver bereit zu stellen.

**Equation-Solver** ist eine weitere Grund-Funktionalität von Computer Algebra. Als TP-basiertes System stellt *ISAC* höhere Anforderungen an seinen Equation-Solver: das Finden *aller* Lösungen einer Gleichungen muss garantiert sein, und der Equation-Solver soll schrittweise Konstruktion von Lösungen unterstützen.

Ziel dieses Sub-Projektes ist, *ISAC*'s existierenden Equation-Solver allgemeiner zu machen, dabei die Simplifier aus dem obigen Sub-Projekt zu verwenden und den Equation-Solver zu erweitern (univariate Gleichungen über komplexen Zahlen, etc).

**Integration für single-stepping Systeme:** Integrieren, als “inverses Problem” zum Differenzieren, wird in Computer Algebra Systemen durch Algorithmen bewerkstelligt, die das Integral sofort in den Bereich komplexer Zahlen transferieren, dort lösen und zurück in den reellen Bereich transformieren — als “black box” Algorithmen.

Dieses Sub-Problem soll die “black box”es öffnen und mit ganz elementaren Integrations-Methoden beginnen. Elementare Methoden jedoch können nur kleine Klassen von Integralen lösen; es werden also mehrere Methoden implementiert — und die zugehörigen Klassen in einer Hierarchie angeordnet. Auf dieser Hierarchie hat *ISAC* ein Verfahren, die passende Klasse herauszufinden (wie das auch der Equation-Solver erfolgreich macht).

### Interessiert ?

Info auf <http://www.ist.tugraz.at/isac/index.php/Development.Computer.Mathematics>.

Prof. Wolfgang Schreiner [Wolfgang.Schreiner@risc.uni-linz.ac.at](mailto:Wolfgang.Schreiner@risc.uni-linz.ac.at) or

Dr. Walther Neuper [wneuper@ist.tugraz.at](mailto:wneuper@ist.tugraz.at)