The Teaching Tool CalcCheck:  
A Proof-Checker for Gries and Schneider’s  
“Logical Approach to Discrete Math”

Wolfram Kahl  
McMaster University, Hamilton, Ontario, Canada  
kahl@cas.mcmaster.ca

Abstract. Students following a first-year course based on Gries and Schneider’s LADM textbook had frequently been asking: “How can I know whether my solution is good?”

We now report on the development of a proof-checker designed to answer exactly that question, while intentionally not helping to find the solutions in the first place. CalcCheck provides detailed feedback to LaTeX-formatted calculational proofs, and thus helps students to develop confidence in their own skills in “rigorous mathematical writing”.

Gries and Schneider’s book emphasises rigorous development of mathematical results, while striking one particular compromise between full formality and customary, more informal, mathematical practises, and thus teaches aspects of both. This is one source of several unusual requirements for a mechanised proof-checker; other interesting aspects arise from details of their notational conventions.

1 Introduction

When teaching a first-year course on Logic and Discrete Mathematics for Computer Science following Gries and Schneider’s textbook “A Logical Approach to Discrete Math” (“LADM” for short) [GS93] for the first time, I obtained feedback from students feeling that the book did not contain sufficiently many worked examples, that insufficient solutions for exercises were available[1], and, especially, that they felt at a loss since they did not see any way of knowing how good their answers were before the marked assignment was returned to them.

The following year (2011), I therefore started to implement “CalcCheck”, a tool intended mainly as a proof-checker for the calculational proof style taught by LADM. For the time being, the usage paradigm of CalcCheck is the same as that of Spivey’s Z type-checker fuzz: CalcCheck also operates on LaTeX source by parsing and analysing the contents of specific formal environments, and providing feedback on those. Using LaTeX as input syntax has the advantage that students learn a general-purpose skill, with only very little formalism-specific overhead.

1 An “Instructor’s Manual” containing solutions exists, but is made available explicitly only to instructors, with the proviso “that answers to selected exercises may be used in lectures or distributed to students as answers to homeworks or tests”. 

© Springer-Verlag Berlin Heidelberg 2011
For example, the following proof can be found on p. 46 of LADM (without the "Proving" line):

**Proving** (3.16) \((p \not\equiv q) \equiv (q \not\equiv p)\):

\[
p \not\equiv q
\]

\[
= \langle \text{Def. of } \not\equiv (3.10) \rangle
\]

\[
\neg (p \equiv q)
\]

\[
= \langle \text{Symmetry of } \equiv (3.2) \rangle
\]

\[
\neg (q \equiv p)
\]

\[
= \langle \text{Def. of } \not\equiv (3.10), \text{ with } p, q := q, p \rangle
\]

\[
q \not\equiv p
\]

Using the \LaTeX macro package accompanying CalcCheck, this proof rendering has been generated from the following \LaTeX source:

\begin{calc}[(3.16) \((p \not\equiv q) \equiv (q \not\equiv p)\)]
\begin{align*}
p \not\equiv q
&\CalcStep{=}{\text{Def. of } \not\equiv (3.10)} \\
\neg (p \equiv q)
&\CalcStep{=}{\text{Symmetry of } \equiv (3.2)} \\
\neg (q \equiv p)
&\CalcStep{=}{\text{Def. of } \not\equiv (3.10), \text{ with } p, q \ becomes q, p} \\
q \not\equiv p
\end{align*}
\end{calc}

The \LaTeX macros have been kept as unobtrusive as possible, with the aim of letting the skill of producing CalcCheck-checked proofs directly improve the skill of producing hand-written proofs in the exams.

Running CalcCheck on an input file containing the above \LaTeX fragment produces the following output to an HTML file, and also in Unicode to the terminal:\footnote{CalcCheck output included in this paper has been rendered by a WWW browser from the CalcCheck-generated HTML files.}

**Proving** (3.16) \((p \not\equiv q) \equiv (q \not\equiv p)\):

\[
p \not\equiv q
\]

\[
= \langle \text{Def. } \not\equiv (3.10) \rangle
\]

--- CalcCheck-0.2.12: (3.10) Definition of \(\not\equiv\) \(-\) OK

\[
\neg (p \equiv q)
\]

\[
= \langle \text{Symmetry of } \equiv (3.2) \rangle
\]

--- CalcCheck-0.2.12: (3.2) Symmetry of \(\equiv\) \(-\) OK (no change)

\[
\neg (q \equiv p)
\]

\[
= \langle \text{Def. } \not\equiv (3.10), \text{ with } p, q \ becomes q, p\rangle
\]

--- CalcCheck-0.2.12: (3.10) Definition of \(\not\equiv\) \(-\) OK

\[
q \not\equiv p
\]

--- CalcCheck-0.2.12: Proof matches goal \(-\) OK