CalcCheck Manual

(Covering CalcCheck-0.2.24 and CalcStyleV9)

Wolfram Kahl

17 January 2012

Abstract

This document describes the use of the prototype proof checker CalcCheck and the accompanying LaTeX package CalcStyle for checking and typesetting the calculational proofs of Gries and Schneider's "Logical Approach to Discrete Math".

1 Preamble

A LaTeX preamble like the following is recommended:

```
\documentclass[11pt]{article}

\usepackage[hmargin=20mm,vmargin=15mm]{geometry} % Fill more of the paper

\usepackage{CalcStyleV8}             % Special macros for math in COMP SCI 1FC3
```

2 Quantification

Quantification is written in the following way:

```
( \star\ x : t \with R \spot E )
( \star\ x : t \withspot E )
```

$$(\star\ x : t \mid R \bullet E)$$

$$(\star\ x : t \mid\!\bullet E)$$

The same patterns are used for **set comprehension**:

```
\{ x : t \with R \spot E \}
\{ x : t \withspot E \}
\{ x : t \with R \}
```

$$\{x : t \mid R \bullet E\}$$

$$\{x : t \mid\!\bullet E\}$$

$$\{x : t \mid R\}$$

3 Declarations

For declarations, inside the `decls` environment the following special macros are available:

- `\declType` for type declarations (type annotations in other contexts just use ":").
- `\declEquiv` for definition of propositions and predicates
- `\declEqu` for definition of other constants and functions
- `\remark` for remarks at the end of a line
- `\also` to separate multiple declarations
- `\BREAK` for line breaks in long right-hand sides

```
\begin{decls}
    P \declEqu \mbox{set of persons}
\also
    A \declType P  \remark{Alex}
\also
    J \declType P
\also
    J \declEqu \mbox{Jane}
\end{decls}
```

P := set of persons
A : P — Alex
J : P
J := Jane

```
\begin{decls}
    called \declType P \times P \tfun \BB
\also
    called(p,q)
  \declEquiv
    \mbox{$p$ called $q$}
\also
    lonely \declType P \tfun \BB
\also
    lonely . p
  \declEquiv
    \lnot (\exists \ q : P
           \BREAK \strut\;
           \withspot called(q,p) )
\end{decls}
```

$\mathit{called} : P \times P \to \mathbb{B}$
$\mathit{called}(p, q) :\equiv p$ called q
$\mathit{lonely} : P \to \mathbb{B}$
$\mathit{lonely}.p :\equiv \neg(\exists\, q : P$
$\quad \bullet\ \mathit{called}(q, p))$

```
\begin{decls}
    father \declType P \tfun P
\also
    father . p
  \declEqu
    \mbox{the father of $p$}
\also
    grandfather \declType P \tfun P
\also
    grandfather . p
  \declEqu
    father(father . p)
\end{decls}
```

$\mathit{father} : P \to P$
$\mathit{father}.p :=$ the father of p
$\mathit{grandfather} : P \to P$
$\mathit{grandfather}.p := \mathit{father}(\mathit{father}.p)$

4 Symbols

For the symbols listed here, always use the LaTeX macros indicated:

Propositional logic:

LaTeX	Output	
`\false`	*false*	Boolean constant *false*
`\true`	*true*	Boolean constant *true*
`\land`	$\land$	conjunction
`\lor`	$\lor$	disjunction
`\implies`	$\Rightarrow$	implication
`\equiv`	$\equiv$	equivalence
`\nequiv` *or* `\not\equiv`	$\not\equiv$	inequivalence
`\lnot`	$\neg$	Boolean negation

Types:

LaTeX	Output	
`\BB`	$\mathbb{B}$	type/set of Boolean values; $\mathbb{B} = \{\mathit{false}, \mathit{true}\}$
`\NN`	$\mathbb{N}$	type/set of natural numbers
`\ZZ`	$\mathbb{Z}$	type/set of integers
`\QQ`	$\mathbb{Q}$	type/set of rational numbers
`\RR`	$\mathbb{R}$	type/set of real numbers
`\CC`	$\mathbb{C}$	type/set of complex numbers
`\times`	$\times$	Cartesian product of sets/types
`\tfun`	$\rightarrow$	type/set of total functions
`\SET{t}`	$\mathit{set}(t)$	type of sets with elements of type t

For commonly used quantification operators, there are alternative symbols:

LaTeX	Output	
`\forall`	$\forall$	quantification with $\wedge$
`\exists`	$\exists$	quantification with $\vee$
`\Sigma`	Σ	quantification with $+$
`\Pi`	Π	quantification with $\cdot$

Set theory:

LaTeX	Output	
`\in`	$\in$	element-of
`\notin` *or* `\not\in`	$\notin$	not-element-of
`\emptyset`	$\emptyset$	Alternative notation for the empty set $\{\}$
`\Universe`	$\mathbf{U}$	the “universe” or domain of discourse (context-dependent!)
`\intersection`	$\cap$	set intersection
`\union`	$\cup$	set union
`-`	$-$	set difference
`\compl`	$\sim$	set complement
`\SET{t}`	$\mathit{set}(t)$	the type of sets with elements of type t
`\power`	$\mathbb{P}$	the (unary) power set operator
`\#`	$\#$	size operator for finite sets: $\# : \mathit{set}(t) \rightarrow \mathbb{N}$
`\subseteq`	$\subseteq$	subset
`\subset`	$\subset$	proper subset
`\supseteq`	$\supseteq$	superset
`\supset`	$\supset$	proper superset
`\not\subseteq`	$\not\subseteq$	negation of subset relation
`\not\subset`	$\not\subset$	negation of proper subset relation
`\not\supseteq`	$\not\supseteq$	negation of superset relation
`\not\supset`	$\not\supset$	negation of proper superset relation

Cartesian Products and Relations:

LaTeX	Output	
`\times`	$\times$	Cartesian product of sets (and of types)
`\langle` x, y `\rangle`	$\langle x, y \rangle$	pair with constituents x and y
`\fst`	fst	first pair projection. Typing: $\mathsf{fst} : (t_1 \times t_2) \to t_1$
`\snd`	snd	second pair projection. Typing: $\mathsf{fst} : (t_1 \times t_2) \to t_2$
`\rel`	$\leftrightarrow$	relation set (and type) constructor: $A \leftrightarrow B = \mathbb{P}(A \times B)$
`\relId .` A	$\mathbb{I}.A$	identity relation on set A. Typing: $\mathbb{I} : \mathbb{P}\ t \to (t \leftrightarrow t)$
`\relDom .` R	$\mathsf{Dom}.R$	domain of relation R. Typing: $\mathsf{Dom} : (t \leftrightarrow u) \to \mathbb{P}\ t$
`\relRan .` R	$\mathsf{Ran}.R$	range of relation R. Typing: $\mathsf{Ran} : (t \leftrightarrow u) \to \mathbb{P}\ u$
R `\converse`	$R^{\smile}$	converse of relation R
`\fcmp`	$\fatsemi$	(forward) relation composition. $\fatsemi : (t_1 \leftrightarrow t_2) \times (t_2 \leftrightarrow t_3) \to (t_1 \leftrightarrow t_3)$
R`^+`	R^+	transitive closure of relation R
R`^*`	R^*	reflexive-transitive closure of relation R

Other functions and operators:

LaTeX	Output	
`\becomes`	$:=$	in substitutions, and later for assignment
`\id`	id	identity function
`\max`	$\uparrow$	binary infix maximum operator
`\min`	$\downarrow$	binary infix minimum operator

5 Examples

In the following examples, we show LaTeX source to the left, and the resulting output to the right.

5.1 Henry VIII had one son and Cleopatra had two.

```
We declare:
\begin{decls}
  h \declEquiv \mbox{Henry VIII had one son}
\also
  c \declEquiv \mbox{Cleopatra had two sons}
\end{decls}
Then the original sentence is formalised as:
\begin{calc}
  h \land c
\end{calc}
```

We declare:

$$h \quad :\equiv \quad \text{Henry VIII had one son}$$

$$c \quad :\equiv \quad \text{Cleopatra had two sons}$$

Then the original sentence is formalised as:

$$h \land c$$

5.2 Substitution

```
\begin{calc}
    (x + y)[x,y \becomes y - 3, z + 2]
  \CalcStep{=}{performing substitution}
    ((y - 3) + (z + 2))
  \CalcStep{=}{removing
                unnecessary parentheses}
    y - 3 + z + 2
\end{calc}
```

$$
\begin{array}{ll}
 & (x + y)[x, y := y - 3, z + 2] \\
= & \quad \langle \text{performing substitution} \rangle \\
 & ((y - 3) + (z + 2)) \\
= & \quad \langle \text{removing unnecessary parentheses} \rangle \\
 & y - 3 + z + 2
\end{array}
$$

5.3 A Problem due to Wim Feijen [Gries 1991]

```
Is the following true or false,
and how do you prove it?
\begin{calc}
    x + y \;\geq\; x \max y
  \qquad\equiv\qquad
    x \geq 0 \;\land\; y \geq 0
\end{calc}

\noindent
To solve the problem,
calculate beginning with the LHS:
\begin{calc}
    x + y \;\geq\; x \max y
  \CalcStep{=}{Definition of $\max$}
    x + y \geq x
    \quad\land\quad
    x + y \geq y
  \CalcStep{=}{Arithmetic}
    y \geq 0
    \quad\land\quad
    x \geq 0
  \CalcStep{=}{Symmetry of $\land$}
    x \geq 0
    \quad\land\quad
    y \geq 0
\end{calc}
```

Is the following true or false, and how do you prove it?

$$x + y \;\geq\; x \uparrow y \qquad\equiv\qquad x \geq 0 \land y \geq 0$$

To solve the problem, calculate beginning with the LHS:

$$\begin{array}{ll} & x + y \;\geq\; x \uparrow y \\ = & \quad \langle \text{Definition of } \uparrow \rangle \\ & x + y \geq x \quad\land\quad x + y \geq y \\ = & \quad \langle \text{Arithmetic} \rangle \\ & y \geq 0 \quad\land\quad x \geq 0 \\ = & \quad \langle \text{Symmetry of } \land \rangle \\ & x \geq 0 \quad\land\quad y \geq 0 \end{array}$$

5.4 Proving a Goal

```
\begin{calc}[(3.5) Reflexivity of $\equiv$,
  $p \equiv p
  $]
    p \equiv p
  \CalcStep{=}{(3.3) Identity of $\equiv$}
    \true
  \ThisIs{(3.4)}
\end{calc}
```

Proving (3.5) Reflexivity of $\equiv$, $p \equiv p$**:**

$$\begin{array}{ll} & p \equiv p \\ = & \quad \langle (3.3) \text{ Identity of } \equiv \rangle \\ & \mathit{true} \quad \text{— This is (3.4)} \end{array}$$

5.5 Substitution Theorem

To avoid having LaTeX misinterpret the closing `]` of substitution as part of a goal as end of the goal, enclose the goal theorem in braces {...} inside the `$...$`.

```
\begin{calc}[(3.84a) ${(e = f) \land E[z \becomes e] \equiv
                     (e = f) \land E[z \becomes f]}$]
    (e = f) \implies (E[z \becomes e] \equiv E[z \becomes f])
  \ThisIs{(3.83) Leibniz Axiom}
  \CalcStep{=}{Definition of $\implies$ (3.60)}
    (e = f) \land (E[z \becomes e] \equiv E[z \becomes f]) \equiv (e = f)
  \CalcStep{=}{(3.49)}
    (e = f) \land E[z \becomes e] \equiv (e = f) \land E[z \becomes f]
\end{calc}
```

Proving (3.84a) $(e = f) \land E[z := e] \equiv (e = f) \land E[z := f]$**:**

$(e = f) \Rightarrow (E[z := e] \equiv E[z := f])$ — This is (3.83) Leibniz Axiom

$=$ ⟨Definition of $\Rightarrow$ (3.60)⟩

$(e = f) \land (E[z := e] \equiv E[z := f]) \equiv (e = f)$

$=$ ⟨(3.49)⟩

$(e = f) \land E[z := e] \equiv (e = f) \land E[z := f]$