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:

\begin{verbatim}
documentclass[11pt]{article}
usepackage[hmargin=20mm,vmargin=15mm]{geometry} % Fill more of the paper
usepackage{CalcStyleV8} % Special macros for math in COMP SCI 1FC3
\end{verbatim}

2 Quantification
Quantification is written in the following way:

\begin{verbatim}
( \star x : t \ with R \ spot E )
( \star x : t \ withspot E )
\end{verbatim}

The same patterns are used for set comprehension:

\begin{verbatim}
\{ x : t \ with R \ spot E \}
\{ x : t \ withspot E \}
\{ x : t \ with R \}
\end{verbatim}

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

- \texttt{\declType} for type declarations (type annotations in other contexts just use “:”).
- \texttt{\declEquiv} for definition of propositions and predicates
- \texttt{\declEqu} for definition of other constants and functions
- \texttt{\remark} for remarks at the end of a line
- \texttt{\also} to separate multiple declarations
- \texttt{\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}

\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 \cdot p
\declEquiv
\lnot \exists q : P \BREAK \ strut:\BREAK \ withspot called(q,p)
\end{decls}

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

4 Symbols

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

Propositional logic:

\begin{center}
\begin{tabular}{|l|l|l|}
\hline
\LaTeX & Output & Description \\
\hline
\texttt{false} & \texttt{false} & \texttt{false} \ Boolean constant \texttt{false} \\
\texttt{true} & \texttt{true} & \texttt{true} \ Boolean constant \texttt{true} \\
\texttt{\land} & \texttt{\&} & \texttt{\&} \ conjunction \\
\texttt{\lor} & \texttt{\lor} & \texttt{\lor} \ disjunction \\
\texttt{implies} & \texttt{\Rightarrow} & \texttt{\Rightarrow} \ implication \\
\texttt{\equiv} & \texttt{\equiv} & \texttt{\equiv} \ equivalence \\
\texttt{\nequiv or \not\equiv} & \texttt{\nequiv} & \texttt{\nequiv} \ inequivalence \\
\texttt{\lnot} & \texttt{\neg} & \texttt{\neg} \ Boolean negation \\
\hline
\end{tabular}
\end{center}
Types:

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

For commonly used quantification operators, there are alternative symbols:

<table>
<thead>
<tr>
<th>\LaTeX</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>\forall</td>
<td>\forall quantification with \land</td>
</tr>
<tr>
<td>\exists</td>
<td>\exists quantification with \lor</td>
</tr>
<tr>
<td>\Sigma</td>
<td>\Sigma quantification with +</td>
</tr>
<tr>
<td>\Pi</td>
<td>\Pi quantification with \cdot</td>
</tr>
</tbody>
</table>

Set theory:

<table>
<thead>
<tr>
<th>\LaTeX</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>\in</td>
<td>\in element-of</td>
</tr>
<tr>
<td>\notin or \not\in</td>
<td>\notin not-element-of</td>
</tr>
<tr>
<td>\emptyset</td>
<td>\emptyset Alternative notation for the empty set {}</td>
</tr>
<tr>
<td>\Universe</td>
<td>\Universe the “universe” or domain of discourse (context-dependent!)</td>
</tr>
<tr>
<td>\intersection</td>
<td>\cap set intersection</td>
</tr>
<tr>
<td>\union</td>
<td>\cup set union</td>
</tr>
<tr>
<td>\setminus</td>
<td>\setminus set difference</td>
</tr>
<tr>
<td>\compl</td>
<td>\compl set complement</td>
</tr>
<tr>
<td>\SET{t}</td>
<td>set(t) the type of sets with elements of type \textit{t}</td>
</tr>
<tr>
<td>\power</td>
<td>\mathcal{P} the (unary) power set operator</td>
</tr>
<tr>
<td>#</td>
<td># size operator for finite sets: # : set(t) \rightarrow \mathbb{N}</td>
</tr>
<tr>
<td>\subseteq</td>
<td>\subseteq subset</td>
</tr>
<tr>
<td>\subset</td>
<td>\subset proper subset</td>
</tr>
<tr>
<td>\supseteq</td>
<td>\supseteq superset</td>
</tr>
<tr>
<td>\supset</td>
<td>\supset proper superset</td>
</tr>
<tr>
<td>\not\subseteq</td>
<td>\not\subseteq negation of subset relation</td>
</tr>
<tr>
<td>\not\subset</td>
<td>\not\subset negation of proper subset relation</td>
</tr>
<tr>
<td>\not\supseteq</td>
<td>\not\supseteq negation of superset relation</td>
</tr>
<tr>
<td>\not\supset</td>
<td>\not\supset negation of proper superset relation</td>
</tr>
</tbody>
</table>
Cartesian Products and Relations:

<table>
<thead>
<tr>
<th>\LaTeX</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>\times</td>
<td>Cartesian product of sets (and of types)</td>
</tr>
<tr>
<td>\langle x, y \rangle</td>
<td>pair with constituents x and y</td>
</tr>
<tr>
<td>\texttt{fst}</td>
<td>first pair projection. Typing: \texttt{fst : (t1 \times t2) \to t1}</td>
</tr>
<tr>
<td>\texttt{snd}</td>
<td>second pair projection. Typing: \texttt{snd : (t1 \times t2) \to t2}</td>
</tr>
<tr>
<td>\texttt{rel} \leftrightarrow</td>
<td>relation set (and type) constructor: ( A \leftrightarrow B = \mathbb{P}(A \times B) )</td>
</tr>
<tr>
<td>\texttt{relId . A} \mapsto \text{Id.A}</td>
<td>identity relation on set A. Typing: ( \text{Id} : \mathbb{P} t \to (t \leftrightarrow t) )</td>
</tr>
<tr>
<td>\texttt{relDom . R} \mapsto \text{Dom.R}</td>
<td>domain of relation R. Typing: ( \text{Dom} : (t \leftrightarrow u) \to \mathbb{P} t )</td>
</tr>
<tr>
<td>\texttt{relRan . R} \mapsto \text{Ran.R}</td>
<td>range of relation R. Typing: ( \text{Ran} : (t \leftrightarrow u) \to \mathbb{P} u )</td>
</tr>
<tr>
<td>\texttt{R \converse} \mapsto \text{R}^-</td>
<td>converse of relation R</td>
</tr>
<tr>
<td>\texttt{fcmp} \mapsto \text{if} ; (t \leftrightarrow u) \times (t \leftrightarrow v) \to (t \leftrightarrow u) \times (t \leftrightarrow v)</td>
<td>(forward) relation composition. ( \text{if} : (t \leftrightarrow t) \times (t \leftrightarrow u) \to \mathbb{P} t )</td>
</tr>
<tr>
<td>\texttt{R}^* \mapsto \text{R}^*</td>
<td>reflexive-transitive closure of relation R</td>
</tr>
</tbody>
</table>

Other functions and operators:

<table>
<thead>
<tr>
<th>\LaTeX</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{becomes} :=</td>
<td>in substitutions, and later for assignment</td>
</tr>
<tr>
<td>\texttt{id}</td>
<td>\text{id}</td>
</tr>
<tr>
<td>\texttt{max} \uparrow</td>
<td>binary infix maximum operator</td>
</tr>
<tr>
<td>\texttt{min} \downarrow</td>
<td>binary infix minimum operator</td>
</tr>
</tbody>
</table>

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

5.2 Substitution

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

We declare:
\begin{calc}
  h \equiv \text{Henry VIII had one son}
\end{calc}
\begin{calc}
  c \equiv \text{Cleopatra had two sons}
\end{calc}
Then the original sentence is formalised as:
\begin{calc}
  h \land c
\end{calc}
5.3 A Problem due to Wim Feijen [Gries 1991]

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

To solve the problem, calculate beginning with the LHS:
\begin{align*}
x + y &\geq x \uparrow y \\
&\equiv \ (\text{Definition of } \uparrow) \\
&\quad x + y \geq x \quad \land \quad x + y \geq y \\
&\equiv \ (\text{Arithmetic}) \\
&\quad y \geq 0 \quad \land \quad x \geq 0 \\
&\equiv \ (\text{Symmetry of } \land) \\
&\quad x \geq 0 \quad \land \quad y \geq 0
\end{align*}

5.4 Proving a Goal

\begin{calc}[(3.5) Reflexivity of $\equiv$, $p \equiv p$]
\begin{align*}
p &\equiv p \\
&\equiv \ (3.3) \text{ Identity of } \equiv \\
&\equiv \text{true} \\
&\equiv \ (3.4)
\end{align*}
\end{calc}

Proving \quad (3.5) \text{ Reflexivity of } \equiv, \ p \equiv p:\
\begin{align*}
p \equiv p \\
&\equiv \ (3.3) \text{ Identity of } \equiv \\
&\equiv \text{true} \quad \text{This is (3.4)}
\end{align*}
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 \{ \ldots \} inside the \$ \ldots \$.

\begin{align*}
( e = f ) \land ( E[z := e] \equiv E[z := f] )
\end{align*}

Proving \eqref{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 \eqref{3.83} Leibniz Axiom

\begin{align*}
( e = f ) \land ( E[z := e] \equiv E[z := f] ) & \equiv ( e = f ) \\
( e = f ) & \Rightarrow ( E[z := e] \equiv E[z := f] ) \\
( e = f ) & \land E[z := e] \equiv ( e = f ) \land E[z := f]
\end{align*}