1 Basic Theories

1.1 Boolean Theory

Operators Some boolean operators are supported by \LaTeX, but they have names suggesting shape rather than content, e.g., \texttt{a \vee b}. It would be nice if they were given informative, short names (because you don’t want to enter \texttt{\&conjunction} all the time) without clashing with existing \LaTeX commands (e.g., \texttt{\&and} is already taken).

\[
\begin{align*}
  a \land b & \quad a \ \text{\&et} \ b \\
  a \lor b & \quad a \ \text{\&vel} \ b \\
  a \Rightarrow b & \quad a \ \text{\&imp} \ b \\
  a \Rightarrow b & \quad a \ \text{\&Imp} \ b \\
  a \Leftarrow b & \quad a \ \text{\&pmi} \ b \text{ or } a \ \text{\&Impby} \ b \\
  a \iff b & \quad a \ \text{\&Pmi} \ b \text{ or } a \ \text{\&Impby} \ b \\
  a \equiv b & \quad a \ \text{\&Eq} \ b \\
\end{align*}
\]

\[
\text{if } b \text{ then } x \text{ else } y \quad \text{\&cond\{b\}\{x\}\{y\}}
\]

The first two come from Latin. The rest should be obvious. Other boolean symbols ($\bot$, $=$, etc.) have reasonable names in \LaTeX, and I will not show them.

Proof format Using the \texttt{align*} environment provided by \texttt{AMS-LaTeX}, a calculational proof with hints can be typeset easily. The first proof in the textbook:

\[
\begin{align*}
  a \land b \Rightarrow c & \quad \text{Material Implication} \\
  \equiv \neg (a \land b) \lor c & \quad \text{Duality} \\
  \equiv \neg a \lor \neg b \lor c & \quad \text{Material Implication} \\
  \equiv a \Rightarrow \neg b \lor c & \quad \text{Material Implication} \\
  \equiv a \Rightarrow (b \Rightarrow c)
\end{align*}
\]

Its code:

\[
\begin{align*}
& \text{\&begin\{align*\}} \\
& \text{\&Blank a \&et b \&imp c} \quad \&\text{\&text\{Material Implication\} \&} \\
& \text{\&Eq \&neg\{a \&et b\} \&vel c} \quad \&\text{\&text\{Duality\} \&} \\
& \text{\&Eq \&neg a \&vel \&neg b \&vel c} \quad \&\text{\&text\{Material Implication\} \&} \\
& \text{\&Eq a \&imp \&neg b \&vel c} \quad \&\text{\&text\{Material Implication\} \&} \\
& \text{\&Eq a \&imp \{b \&imp c\}} \\
& \text{\&end\{align*\}}
\end{align*}
\]

The command \texttt{\&Blank} is a blank relation symbol I invented; it is necessary in that position to keep \texttt{align*} happy. Its definition is simply: \texttt{\mathrel\{\phantom\{\Eq\}\}}.
2 Basic Data Structures

Bunch Theory

\[ \begin{align*}
A, B & \quad A, B \\
A' B & \quad A' B \\
null & \quad \text{nul}
\end{align*} \]

\[ \begin{align*}
fA & \quad \text{card } A \\
0..10 & \quad 0 \text{ \text{lto} } 10 \\
nat, xnat & \quad \text{nat, } \text{xnat} \\
int, xint & \quad \text{int, } \text{xint} \\
grat, xrat & \quad \text{rat, } \text{xrat}
\end{align*} \]

String Theory

\[ \begin{align*}
nil & \quad \text{nil} \\
n^* S & \quad n^* S \\
^* S & \quad \{\}^* S \\
0..10 & \quad 0 \text{ \text{lto} } 10
\end{align*} \]

List Theory

\[ \begin{align*}
L^+ M & \quad L^+ M \\
n \to i \mid L & \quad n \text{ \text{lto} } i \text{ \text{ow} } L \\
L n & \quad L \text{ \text{ap} } n
\end{align*} \]

3 Function Theory

The \texttt{fun} and \texttt{fn} commands produce functions; \texttt{fun} requires a domain and \texttt{fn} omits the domain.

\[ \begin{align*}
\langle x : \text{nat} \to x + 1 \rangle & \quad \texttt{fun}\{x\}\{\text{nat}\}\{x+1\} \\
\langle x \to x + 1 \rangle & \quad \texttt{fn}\{x\}\{x+1\}
\end{align*} \]

The \texttt{bind} and \texttt{bnd} commands help you produce quantified expressions. They just have the quantifier missing, and you just put it back. \texttt{bind} requires a domain and \texttt{bnd} omits the domain. Some examples:

\[ \begin{align*}
\forall x : x = x & \quad \texttt{forall}\texttt{bnd}\{x\}\{x=x\} \\
\Sigma i : 0..10 \cdot i^2 & \quad \texttt{Sigma}\texttt{bnd}\{i\}\{0 \text{ \text{lto} } 10\}\{i^2\} \\
\exists x : \text{nat} \cdot x/2 : \text{nat} & \quad \texttt{S}\texttt{bnd}\{x\}\{\text{nat}\}\{x/2: \text{nat}\}
\end{align*} \]

Two quantifiers are not already available in \LaTeX: \texttt{MAX} and \texttt{MIN}. I have defined them as \texttt{MAX} and \texttt{MIN}, respectively.

Both application and composition are \texttt{ap}. You can think of it as standing for “apposition”. Selective union is \texttt{ow}, standing for “otherwise”. You have seen them in List Theory. More examples:
\begin{align*}
MAX v : x \cdot n &= \text{MAX}\bind{v}{x}{n} \\
MIN v : x \cdot n &= \text{MIN}\bind{v}{x}{n} \\
f \downarrow g &= f \downarrow ow g \\
h \downarrow f x g y &= h \downarrow ap f \downarrow ap x \downarrow ap g \downarrow ap y
\end{align*}

4 Program Theory

\begin{align*}
\texttt{ok} &= \texttt{ok} \\
S \cdot R &= S \downarrow dc R \\
x := e &= x \downarrow get e
\end{align*}