

Beachten Sie die Verwendung von `nonumberplain` als Stil für die beweis-Umgebung sowie das `\theoremsymbol`, das das Ende des Beweises angibt (hier ein schwarzes Quadrat). Diese Symbole erhalten Sie, wenn Sie die Paketoption `thmmarks` für `ntheorem` angeben:

```
\usepackage[thmmarks]{ntheorem}
```

Das `\theoremsymbol` ist mitunter schwierig zu platzieren (die Dokumentation von `ntheorem` behandelt die entsprechenden Details). `ntheorem` verfolgt hier einen zweistufigen Ansatz und lagert Informationen in der `.aux`-Datei zwischen; Sie müssen also mindestens zwei \LaTeX -Läufe durchführen, bis Sie korrekt positionierte Endmarken haben.



HACK #22 Matrizen und ähnliche Konstruktionen

Die Matrix – reloaded: Hier zeigen wir Ihnen einige Methoden für den Umgang mit rechteckigen Zahlenanordnungen aller Art.

Matrizen sind (unter anderem) aus Algebra- oder Numerikvorlesungen nicht wegzudenken. \LaTeX erlaubt mit der `array`-Umgebung einfache Konstruktionen wie

$\left[\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array} \right]$	<pre>\[\left[\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array} \end{array} \right]</pre>
---	---

aber bei Komplizierterem muss es bald passen, was die »Hausmittel« angeht. Bequemer ist zum Beispiel das `amsmath`-Paket, das diverse mathematische Erweiterungen bietet, etwa Matrizen mit »mitgelieferten« Klammern, unter anderem

$$\begin{array}{cc} a & -b \\ c^2 & d \end{array} \quad \left(\begin{array}{cc} a & -b \\ c^2 & d \end{array} \right) \quad \left[\begin{array}{cc} a & -b \\ c^2 & d \end{array} \right] \quad \left| \begin{array}{cc} a & -b \\ c^2 & d \end{array} \right|$$

mit der dazugehörigen Eingabe

```
\[
\begin{matrix} a & -b \\ c^2 & d \end{matrix} \quad \begin{pmatrix} a & -b \\ c^2 & d \end{pmatrix} \quad \begin{bmatrix} a & -b \\ c^2 & d \end{bmatrix} \quad \begin{vmatrix} a & -b \\ c^2 & d \end{vmatrix}
```

Dies ist ein Abzug aus dem `amsmath`-Paket, Copyright © 1995-1996 by Leslie Lamport, 78-3-89721-477-4
<http://www.oreilly.de/catalog/latex/hksgger/>
Dieses Abzug unterliegt dem Urheberrecht. © O'Reilly Verlag 2007

```

c^2 & d \end{bmatrix} \quad
\begin{vmatrix} a & -b \\
c^2 & d \end{vmatrix}
\]

```

Diese Umgebungen haben im Gegensatz zu array keine tabellenartige Kopfzeile, sondern verwenden eine Standardvorgabe von bis zu zehn zentrierten Spalten (änderbar, falls nötig, über den Zähler MaxMatrixCols).

Für noch kompliziertere Matrizen ist das easymat-Paket nützlich, das Teil des easy-Bündels von Enrico Bertolazzi ist: Die folgende Hilbert-Matrix

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

bekommen Sie mit der Eingabe

```

\[ \left[
\begin{MAT}(b){c;ccc}
1 & \frac{1}{2} & \frac{1}{3} & \cdots \\
\frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots \\
\frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots \\
\vdots & \vdots & \vdots & \ddots \\
\end{MAT}
\right] \]

```

Beachten Sie insbesondere die Spezifikation der gepunkteten Linien über die an strategischen Stellen platzierten Semikolons.

Ebenfalls eine Art Matrix (im weitesten Sinne) definiert die cases-Umgebung von amsmath:

$$A(m,n) = \begin{cases} n + 1 & \text{falls } m = 0 \\ A(m - 1, 1) & \text{falls } m > 0 \text{ und } n = 0 \\ A(m - 1, A(m, n - 1)) & \text{falls } m > 0 \text{ und } n > 0 \end{cases} \quad (2.1)$$

Die Eingabe dazu ist

```

\begin{equation}
A(m,n) =

```

Dies ist ein Auszug aus dem Buch „*Latex Hacks*“, ISBN 978-3-89721-477-4
<http://www.oreilly.de/catalog/latexhksger/>
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2007

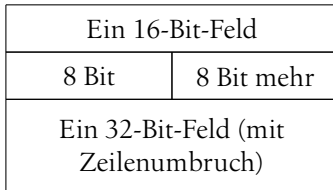
```
\begin{cases}
n + 1 & \text{falls } m=0 \\
A(m-1, 1) & \text{falls } m>0 \text{ und } n=0 \\
A(m-1, A(m, n-1)) & \text{falls } m>0 \text{ und } n>0
\end{cases}
```



Bytefelder und Protokolldateneinheiten

Setzen Sie bequem Schaubilder für Datenstrukturen und Netzwerkprotokolle.

Warum gleich ein Grafikprogramm anwerfen, wenn es darum geht, ein Datenformat für ein Netzwerkprotokoll darzustellen? Es gibt doch das `bytefield`-Paket von Scott Pakin, mit dem das in \LaTeX ein pures Kinderspiel ist. Betrachten Sie das folgende einfache Beispiel (aus der `bytefield`-Dokumentation gehliehen):



```
\begin{bytefield}{16}
\wordbox{1}{Ein 16-Bit-Feld}\
\bitbox{8}{8 Bit}
& \bitbox{8}{8 Bit mehr} \
\wordbox{2}{Ein 32-Bit-Feld
(mit Zeilenumbruch)}
\end{bytefield}
```

Sie müssen nur angeben, wie viele Bit Ihr Diagramm breit sein soll, und können dann `\bitbox`- und `\wordbox`-Kommandos verwenden, um die verschiedenen Felder einzutragen. `\bitbox`-Felder teilen sich zu mehreren nebeneinander eine Zeile, während `\wordbox`-Felder immer eine ganze Zeile einnehmen, aber mehrere Zeilen hoch sein dürfen. Das erste Argument ist bei `\bitbox` die Breite in Bit, bei `\wordbox` die Höhe in Byte; das zweite ist in beiden Fällen der Text, der im betreffenden Feld erscheinen soll.

Beide Arten von Boxen haben ein optionales Argument, das angibt, welche Seiten tatsächlich gezeichnet werden sollen – eine Teilmenge von »tblr«, nämlich *top* (oben), *bottom* (unten), links und rechts. Damit sind auch krause Formate möglich wie



```
\begin{bytefield}{16}
\bitbox{4}{4 Bit}
& \bitbox[lrt]{12}{} \
\wordbox[lrb]{1}{28-Bit-Feld}
\end{bytefield}
```

Sie können auch eine Kopie der `bytefield`-Bibliographie hinzufügen:

Dies ist ein Auszug aus dem Buch „*Latex Hacks*“, ISBN 978-3-89721-477-4
Copyright © Scott Pakin, 2005.
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2007