LilyPond

Das Notensatzprogramm

Programmbenutzung

Das LilyPond-Entwicklerteam

Copyright © 1999–2008 bei den Autoren

The translation of the following copyright notice is provided for courtesy to non-English speakers, but only the notice in English legally counts.

Die Übersetzung der folgenden Lizenzanmerkung ist zur Orientierung für Leser, die nicht Englisch sprechen. Im rechtlichen Sinne ist aber nur die englische Version gültig.

> Es ist erlaubt, dieses Dokument unter den Bedingungen der GNU Free Documentation Lizenz (Version 1.1 oder spätere, von der Free Software Foundation publizierte Versionen, ohne Invariante Abschnitte), zu kopieren, verbreiten und/oder zu verändern. Eine Kopie der Lizenz ist im Abschnitt "GNU Free Documentation License" angefügt.
>
> Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections. A copy of the license is included in the section entitled "GNU Free Documentation License".

Für LilyPond Version 2.12.1

Inhaltsverzeichnis

1 Installieren ... 1

1.1 Vorkompilierte Binär-Pakete ... 1
Herunterladen ... 1
1.2 Aus den Quellen übersetzen ... 1

2 Setup ... 2

2.1 Setup für bestimmte Betriebssysteme ... 2
2.1.1 Einrichtung für MacOS X ... 2
2.2 Unterstützung von Texteditoren ... 3
2.2.1 Emacs-Modus ... 3
2.2.2 Vim-Modus ... 3
2.2.3 jEdit ... 3
2.2.4 TexShop ... 3
2.2.5 TextMate ... 4
2.2.6 LilyKDE ... 4
2.3 Point and click ... 4

3 LilyPond starten ... 6

3.1 Übliche Programmbenutzung ... 6
3.2 Benutzung auf der Kommandozeile ... 6
3.2.1 `lilypond` aufrufen ... 6
3.2.2 Optionen auf der Kommandozeile für `lilypond` ... 6
3.2.3 Umgebungsvariablen ... 10
3.3 Fehlermeldungen ... 10
3.4 Dateien mit `convert-ly` aktualisieren ... 11
3.4.1 Optionen von `convert-ly` auf der Kommandozeile ... 12
3.4.2 Probleme mit `convert-ly` ... 12
3.5 Fehler melden ... 14

4 `lilypond-book`: Noten in den Text integrieren ... 15

4.1 Ein musikwissenschaftlicher Text als Beispiel ... 15
4.2 Noten in Text integrieren ... 18
4.2.1 LaTeX ... 18
4.2.2 Texinfo ... 20
4.2.3 HTML ... 20
4.2.4 DocBook ... 21
4.3 Die Musikfragment-Optionen ... 22
4.4 `lilypond-book` aufrufen ... 25
4.5 Dateiendungen ... 27
4.6 Alternative Methoden, Text mit Musik zu kombinieren ... 28
4.6.1 Viele Zitate aus einer langen Partitur ... 28
4.6.2 LilyPond-Noten in OpenOffice.org integrieren ... 28
4.6.3 LilyPond-Noten in andere Programme integrieren ... 28

5 Von anderen Formaten konvertieren 29

5.1 `midi2ly` aufrufen 29
5.2 `muscxml2ly` aufrufen 30
5.3 `abc2ly` aufrufen 31
5.4 `etf2ly` aufrufen 32
5.5 LilyPond-Dateien erstellen 32

Anhang A GNU Free Documentation License 34

Anhang B LilyPond-Index 40

1 Installieren

Es gibt von Lilypond zwei verschiedene Typen von Versionen: stabile Versionen und instabile Entwicklerversionen. Stabile Versionen haben eine gerade Unter-Versionsnummer (z.B. 2.8, 2.10, 2.12, etc.). Entwicklerversionen haben hingegen ungerade Unter-Versionsnummern (z.B. 2.7, 2.9, 2.11, etc.).

LilyPond selbst zu kompilieren ist ein relativ komplizierter Vorgang, daher empfehlen wir **stark**, die vorkompilierten Binär-Pakete zu benutzen.

1.1 Vorkompilierte Binär-Pakete

Herunterladen

Aktuelle Informationen zu vorkompilierten Binär-Paketen für Ihre Plattform finden Sie unter `http://lilypond.org/web/install/`. Falls Ihr Betriebssystem auf dieser allgemeinen Seite nicht behandelt wird, sehen Sie bitte in der vollständigen Liste unter `http://download.linuxaudio.org/lilypond/binaries/` nach.

Momentan werden Binärpakete für folgende Plattformen erstellt:

```
darwin-ppc  - MacOS X powerpc
darwin-x86  - MacOS X intel
freebsd-64  - FreeBSD 6.x, x86_64
freebsd-x86 - FreeBSD 4.x, x86
linux-64    - Beliebige GNU/Linux Distribution, x86_64
linux-ppc   - Beliebige GNU/Linux Distribution, powerpc
linux-x86   - Beliebige GNU/Linux Distribution, x86
mingw       - Windows x86
```

Bekannte Probleme und Warnungen

Wenn Sie MacOS 10.3 oder 10.4 benutzen und Python-Skripte wie `convert-ly` und `lilypond-book` benutzen wollen, lesen Sie Abschnitt “Setup for MacOS X” in *Application Usage*.

1.2 Aus den Quellen übersetzen

LilyPond kann auch selbst direkt aus dem Quellcode des git-Depots kompiliert werden. Da jedoch für die Kompilierung definitiv Englisch-Kenntnisse vorhanden sein müssen, soll hier lediglich auf die englische Dokumentation verwiesen werden: Abschnitt “Compiling from source” in *Application Usage*.

2 Setup

In diesem Kapitel werden verschiedene Konfigurationsoptionen für LilyPond und andere Programme behandelt, die nach der Installation auszuführen sind. Dieses Kapitel kann als Referenz betrachtet werden, lesen Sie einen Abschnitt nur, wenn er auf Sie zutrifft.

2.1 Setup für bestimmte Betriebssysteme

Dieser Abschnitt erklärt zusätzliche Einstellungen für bestimmte Betriebssysteme.

2.1.1 Einrichtung für MacOS X

Benutzung von Python-Skripten unter MacOS 10.3 oder 10.4

Das LilyPond-Programmpaket für MacOS X stellt Python nicht zur Verfügung, aber Python 2.4 oder höher wird benötigt, um Skripte wie `convert-ly` zu benutzen. Wenn Sie MaxOS 10.3 oder 10.4 besitzen, müssen Sie eine neurere Python-Version von http://python.org/download/ installieren und dann die erste Zeile von `convert-ly` und `lilypond-book` auf folgende Weise ändern: Wenn sich das installierte Python-Programm in Ihrem *PFAD* befindet, muss die erste Zeile heißen:

```
#!/usr/bin/env python
```

andernfalls muss sie heißen:

```
#!/path/to/newly_installed/python
```

MacOS X auf der Kommandozeile

Die Skripte (etwa `lilypond-book`, `convert-ly`, `abc2ly` und sogar `lilypond`) befinden sich innerhalb der `.app`-Datei für MacOS X. Sie können auf der Kommandozeile direkt aufgerufen werden, etwa

```
Pfad/zu/LilyPond.app/Contents/Resources/bin/lilypond
```

Entsprechend funktionieren auch die anderen Skripte wie `lilypond-book`, `convert-ly`, `abc2ly` usw.

Sie können sich auch selber Skripte anlegen, die diesen Pfad automatisch hinzufügen. Erstellen Sie ein Verzeichnis, indem die Skripte gespeichert werden:

```
mkdir -p ~/bin
cd ~/bin
```

Erstellen Sie eine Datei mit dem Namen `lilypond` und dem Inhalt

```
exec path/to/LilyPond.app/Contents/Resources/bin/lilypond "$@"
```

Erstellen Sie entsprechende Dateien mit den Namen `lilypond-book`, `convert-ly` und den Namen der anderen Hilfsprogramme, die Sie benutzen (`abc2ly`, `midi2ly` usw.). Ersetzen Sie einfach den Teil `bin/lilypond` mit `bin/convert-ly` (oder einem anderen Programmnamen) in der entsprechenden Datei.

Machen Sie die Datei ausführbar:

```
chmod u+x lilypond
```

Jetzt müssen Sie dieses Verzeichnis noch zu Ihrem Pfad (PATH) hinzufügen. Verändern Sie die Datei `.profile` in Ihrem Benutzerverzeichnis (oder erstellen Sie sie), dass sie die Zeile

```
export PATH=$PATH:~/bin
```

beinhaltet. Die Datei muss mit einer Leerzeile enden.

Beachten Sie, dass *Pfad/zu* üblicherweise `/Applications/` ist.

2.2 Unterstützung von Texteditoren

Verschiedene Texteditoren haben Unterstützung für LilyPond.

2.2.1 Emacs-Modus

Emacs hat einen LilyPond-Modus (‘`lilypond-mode`’), eine Datei, die die Vervollständigung von Befehlen, Einrückungen, für LilyPond spezifische Klammerschließungen und die Markierung der Syntax beherrscht. Zusätzlich stehen noch praktische Tastaturkombinationen zum Programmaufruf und zum Nachschlagen in den Handbüchern zur Verfügung. Siehe unten, wenn die Datei ‘`lilypond-mode`’ sich nicht auf Ihrem Computer befindet.

Der Emacs-Modus zur Noteneingabe und zum Programmaufruf ist in den Emacs-Quellen im ‘`elisp`’-Verzeichnis enthalten. Mit dem Befehl `make install` können Sie es nach *elispdir* installieren. Die Datei ‘`lilypond-init.el`’ sollte in *load-path*‘`/site-start.d/`’ liegen oder Ihrem ‘`~/.emacs`’ or ‘`~/.emacs.el`’ hinzugefügt werden.

Als Benutzer können Sie Ihren Quellenpfad (etwa ‘`~/site-lisp/`’) ihrem *load-path* hinzufügen, indem Sie folgende Zeile zu Ihrer ‘`~/.emacs`’-Datei hinzufügen:

```
(setq load-path (append (list (expand-file-name "~/site-lisp")) load-path))
```

2.2.2 Vim-Modus

Für VIM wird ein ‘`vimrc`’ bereitgestellt, zusammen mit Werkzeugen zur Syntaxauszeichnung. Ein Vim-Modus zur Noteneingabe und zum Programmaufruf befindet sich im Quellarchiv im `$VIM`-Verzeichnis.

LilyPond-Dateien werden automatisch erkannt, wenn sich in der Datei ‘`~/.vim/filetype.vim`’ folgender Inhalt befindet:

```
if exists("did_load_filetypes")
  finish
endif
augroup filetypedetect
  au! BufNewFile,BufRead *.ly            setf lilypond
augroup END
```

Fügen Sie den LilyPond-Pfad in ihre ‘`~/.vimrc`’-Datei ein, indem Sie ihre folgende Zeile hinzufügen:

```
set runtimepath+=/usr/local/share/lilypond/${LILYPOND_VERSION}/vim/
```

Dabei wird ${LILYPOND_VERSION}durch Ihre LilyPond-Version ersetzt. Sollten Sie LilyPond nicht nach ‘`/usr/local/`’ installiert haben, müssen Sie den Pfad entsprechend anpassen.

2.2.3 jEdit

Das Plugin für den jEdit-Texteditor, LilyPondTool genannt, ist das vielfältigste textbasierte Werkzeug, um LilyPond-Notationsdateien zu editieren. Zu den besonderen Eigenschaften gehört ein Dokument-Ersteller mit Liedtextunterstützung, der die Neuerstellung von Dateien erleichtert, und ein integriertes PDF-Anzeigeprogramm mit „point-and-click“-Unterstützung. Demos, Bildschirmphotos und Installationsanweisungen finden sich unter `http://lilypondtool.organum.hu`

2.2.4 TexShop

Der TexShop-Editor für MacOS X kann erweitert werden, um LilyPond, lilypond-book und convert-ly aus dem Editor heraus zu starten. Die Erweiterung findet sich unter `http://www.dimi.uniud.it/vitacolo/freesoftware.html`.

2.2.5 TextMate

Es gibt ein LilyPond-Paket für TextMate. Es kann installiert werden, indem man

```
mkdir -p /Library/Application\ Support/TextMate/Bundles
cd /Library/Application\ Support/TextMate/Bundles
svn co http://macromates.com/svn/Bundles/trunk/Bundles/Lilypond.tmbundle/
```

ausführt.

2.2.6 LilyKDE

LilyKDE ist eine Erweiterung für den Texteditor Kate für KDE. LilyKDE beinhaltet einen mächtigen Assistenten, mit dem sehr schnell neue LilyPond-Dokumente erstellt werden könne, sowie einen eingebetteten PDF-Betrachter.

LilyKDE benutzt auch Rumor, um Musik direkt von einem MIDI-Keyboard einzulesen.

Weitere Fähigkeiten von LilyKDE sind Silbentrennung für Liedtexte oder die Möglichkeit, LilyPond gleichzeitig mit mehreren Eingabedateien vom KDE Dateimanager aus aufzurufen.

2.3 Point and click

Point and click erlaubt es, die Noten in der Quelldatei zu finden, indem man sie im PDF anklickt. Das erleichtert es, fehlerhafte Stellen zu finden.

Wenn diese Funktionalität aktiv ist, fügt LilyPond Hyperlinks zum PDF hinzu. Diese werden dann bei einem Klick zum Browser geschickt, der wiederum einen Texteditor mit dem Cursor an der richtigen Stelle öffnet.

Damit diese Kettenreaktion funktioniert, müssen Sie ihrem PDF-Programm mit dem Skript '`lilypond-invoke-editor`' beibringen, Hyperlinks zu folgen.

Im Falle von Xpdf auf einer UNIX-Maschine sollte folgende Zeile in der Datei '`xpdfrc`'[1] stehen.

```
urlCommand     "lilypond-invoke-editor %s"
```

Das Programm '`lilypond-invoke-editor`' ist ein kleines Hilfsprogramm. Es ruft einen Editor auf für bestimmte `textedit`-URIs und startet einen Browser für die anderen. Die Umgebungsvariable `EDITOR` wird für folgende Zeichenketten:

`emacs` das startet

```
emacsclient --no-wait +line:column file
```

`vim` das startet

```
gvim --remote +:line:normchar file
```

`nedit` das startet

```
nc -noask +line file'
```

Die Umgebungsvariable `LYEDITOR` wird verwendet, um dieses zu überschreiben. Sie enthält den Befehl, um den Editor aufzurufen, wobei die Variablen `%(file)s`, `%(column)s`, `%(line)s` mit der Datei, Spalte und Zeile ersetzt werden. Die Einstellung

```
emacsclient --no-wait +%(line)s:%(column)s %(file)s
```

für `LYEDITOR` etwa entspricht dem von vornherein eingestellten Emacs-Aufruf.

Die point and click-Links vergrößern die Notationsdatei erheblich. Um die Größe der PDF- und PS-Dateien zu verringern, kann point and click ausgeschaltet werden, indem die Zeile

[1] Unter UNIX befindet sich diese Datei entweder in '`/etc/xpdfrc`' oder als '`.xpdfrc`' in Ihrem Heim-Verzeichnis.

```
\pointAndClickOff
```

in der ‘`.ly`’-Datei gesetzt wird. Point and Click kann auch explizit eingeschaltet werden durch die Zeile:

```
\pointAndClickOn
```

Alternativ können Sie Point and Click auch mit einer Kommandozeilenoption ausschalten:

```
lilypond -dno-point-and-click file.ly
```

> **Achtung:** Sie sollten Point and Click immer für Dateien ausschalten, die Sie an andere Personen weitergeben möchten. Anderenfalls werden Pfadinformationen Ihres Computers in die PDF-Datei kopiert, was ein potentielles Sicherheitsrisiko darstellt.

3 LilyPond starten

Dieses Kapitel behandelt die technischen Details, wie Lilypond ausgeführt werden kann.

3.1 Übliche Programmbenutzung

Die meisten Benutzer führen LilyPond von einer graphischen Benutzeroberfläche aus. Siehe Abschnitt “First steps” in *Handbuch zum Lernen*, falls Sie dies nicht bereits getan haben.

3.2 Benutzung auf der Kommandozeile

Dieser Abschnitt enthält zusätzliche Informationen, wie Sie LilyPond von der Kommandozeile ausführen können. Dies kann erforderlich sein, um etwa zusätzliche Optionen an das Programm zu übergeben. Außerdem sind einige Zusatzprogramme (wie etwa `midi2ly`) nur von der Kommandozeile verfügbar.

Unter ‚Kommandozeile‘ verstehen wir die Kommandozeile des jeweiligen Betriebssystems. Windows Benutzern ist sie vielleicht eher unter den englischen Begriffen ‚DOS shell‘ oder ‚command shell‘ bekannt. MacOS X Benutzer kennen sie eher unter ‚Terminal‘ oder ‚Konsole‘. Sie sollten auch den Abschnitt Abschnitt 2.1.1 [Einrichtung für MacOS X], Seite 2 konsultieren.

Wie die Kommandozeile im jeweiligen Betriebssystem benutzt werden kann, soll in diesem Handbuch nicht näher beschrieben werden. Sehen Sie bitte im Handbuch Ihres Betriebssystems nach oder informieren Sie sich im Internet, wenn Sie mit der Kommandozeile nicht vertraut sind.

3.2.1 `lilypond` aufrufen

Das `lilypond` Programm kann folgendermaßen von der Kommandozeile aufgerufen werden.

```
lilypond [Option]... Dateiname...
```

Wird ein ‘`Dateiname`’ ohne Erweiterung angegeben, so wird ‘`.ly`’ als Standarderweiterung für LilyPond-Dateien benutzt. Um Daten von `stdin` einzulesen, benutzen Sie einfach einen Bindestrich (`-`) als *Dateiname*.

Wenn Lilypond die Datei ‘`Dateiname.ly`’ verarbeitet, werden daraus die Dateien ‘`Dateiname.ps`’ und ‘`Dateiname.pdf`’ erzeugt. Es können an `lilypond` auch mehrere ‘`.ly`’ Dateien übergeben werden, die dann einzeln und voneinander unabhängig abgearbeitet werden.[1]

Falls ‘`Dateiname.ly`’ mehr als einen `\score`-Block enthält, werden die weiteren Stücke in durchnummerierte Dateien der Form ‘`Dateiname-1.pdf`’ ausgegeben. Zusätzlich wird der Wert der Variable `output-suffix` zwischen den ursprünglichen Dateienamen und der Zahl eingefügt. Eine Lilypond-Datei *Dateiname.ly* mit dem Inhalt

```
#(define output-suffix "Geige")
\book { ... }
#(define output-suffix "Cello")
\book { ... }
```

erzeugt daher die Dateien *Dateiname*‘`-Geige.pdf`’ und *Dateiname*‘`-Cello-1.pdf`’.

3.2.2 Optionen auf der Kommandozeile für `lilypond`

Die folgenden Kommandozeilenoptionen werden von `lilypond` unterstützt:

`-e,--evaluate=`*expr*

> Wertet den Scheme-Ausdruck *expr* aus, bevor die ‘`.ly`’ Dateien gelesen und interpretiert werden. Die `-e` Option kann auch mehrfach angegeben werden, die Ausdrücke werden nacheinander ausgewertet.

[1] Der Zustand von GUILE wird allerdings nicht nach jeder Datei zurückgesetzt, sodass Achtung geboten ist, wenn in einer Datei globale Änderungen von Scheme aus durchgeführt werden.

Da der Ausdruck im `guile-user` Modul ausgewertet wird, ist bei der Definitionen innerhalb von *expr* folgendes Vorgehen nötig. An der Kommandozeile wird z.B. `a` im `guile-user` Modul definiert:

```
lilypond -e '(define-public a 42)'
```

Am Beginn der '`.ly`'-Datei muss dann das `guile-user` Modul noch geladen werden, bevor die Definition von *a* verfügbar ist:

```
#(use-modules (guile-user))
```

`-f,--format=`*Format*

Bestimmt das Ausgabeformat. Mögliche Werte von *Format* sind `svg`, `ps`, `pdf` und `png`.

Beispiel: `lilypond -fpng` *Dateiname*`.ly`

`-d,--define-default=`*Variable*`=`*Wert*

Damit wird die interne Programmoption *Variable* auf den Scheme-Wert *Wert* gesetzt. Wird kein *Wert* angegeben, so wird *#t* benutzt. Um eine Option auszuschalten, kann der Präfix `no-` dem Namen *Variable* der Variable vorangestellt werden. So ist etwa

```
-dno-point-and-click
```

dasselbe wie

```
-dpoint-and-click='#f'
```

Hier sind ein paar interessante Optionen:

'`help`' Die Ausführung von `lilypond -dhelp` zeigt alle verfügbaren `-d` Optionen.

'`paper-size`'

Setzt das Standard-Papierformat,

```
-dpaper-size=\"letter\"
```

Die Zeichenkette, die das Format angibt, muss in Anführungszeichen mit Backslash (`\"`) stehen.

'`safe`' Vertraut der `.ly` Datei nicht.

Wenn LilyPond über einen Webserver verfügbar gemacht wird, **MUSS** unbedingt eine die Optionen `--safe` oder `--jail` angegeben werden. Die `--safe` Option verhindert, dass in der .ly-Datei angegebener Scheme-Code das System gefährden kann, wie etwa in folgendem Beispiel:

```
#(system "rm -rf /")
{
  c4^#(ly:export (ly:gulp-file "/etc/passwd"))
}
```

Mit der `-dsafe` Option werden alle Scheme-Ausdrücke einem speziellen sicheren Modus ausgewertet. Dieser Modus ist vom GUILE '`safe-r5rs`' Modul abgeleitet und fügt noch zahlreiche weitere erlaubte Funktionen der LilyPond Programm-Schnittstelle hinzu. Diese Funktionen sind in '`scm/safe-lily.scm`' angegeben.

Zusätzliche verbietet der sichere Modus auch `\include` Befehle.

Im sicheren Modus ist es nicht möglich, LilyPond-Variablen nach Scheme zu exportieren.

`-dsafe` erkennt jedoch *KEINE* Überbeanspruchung der verfügbaren Ressourcen. In diesem Modus ist es also trotzdem möglich, dass LilyPond in einer Endlosschleife hängt, z.B. wenn zyklische Datenstrukturen an das Backend übergeben werden. Wenn LilyPond also auf einem öffentlich zugänglichen Webserver verfügbar gemacht wird, sollte der Prozess sowohl in der CPU- als auch in der Speichernutzung limitiert werden.

Der sichere Modus verhindert auch, dass zahlreiche nützliche Musikfragmente von LilyPond verarbeitet werden. Die `--jail` Option ist eine sicherere Alternative, benötigt allerdings auch mehr Aufwand zur Einrichtung.

'`backend`' Gibt an, welches Ausgabeformat das LilyPond Backend benutzt. Mögliche Werte für diese Option sind:

`ps` PostScript-Ausgabeformat.

Postscript-Dateien enthalten auch TTF-, Type1- und OTF-Schriften. Allerdings wird die gesamte Schriftart eingefügt und nicht nur die benötigten Zeichen. Vor allem wenn nicht-westliche Zeichensätze benutzt werden, kann dies zu sehr großen Dateien führen.

`eps` Erzeugt ‚encapsulated PostScript' (EPS). Jede Seite (oder jedes System) wird als eigene '`EPS`'-Datei ausgegeben, inklusive Schriftarten. Außerdem wird eine Datei mit allen Seiten (bzw. Systemen) und Schriftarten erzeugt.

Dies ist die Standardeinstellung von `lilypond-book`.

`svg` SVG-Ausgabe (Scalable Vector Graphics). Jede Seite wird als eigene '`SVG`'-Datei ausgegeben, inklusive eingebetteten Schriftarten. Sie benötigen einen SVG-Betrachter, der eingebettete Schriftarten unterstützt, oder einen SVG-Betrachter, der eingebettete Schriftarten durch OTF-Schriften ersetzen kann. In UNIX und Linux kann z.B. Inkscape (ab Version 0.42) benutzt werden, nachdem die OTF-Schriften aus dem LilyPond-Verzeichnis (typischerweise '`/usr/share/lilypond/VERSION/fonts/otf/`') in das Verzeichnis '`~/.fonts/`' kopiert wurden.

`scm` gibt die rohen Scheme-basierenden Zeichenbefehle aus, wie sie intern von LilyPond benutzt werden.

`null` Keine Partitur wird ausgegeben, hat gleichen Effekt wie `-dno-print-pages`.

Beispiel: `lilypond -dbackend=svg` *`Dateiname`*`.ly`

'`preview`' Erzeugt eine Ausgabedatei, die nur die Titelzeilen und das erste System enthält.

'`print-pages`'
Erzeugt vollständige Seiten (Standardeinstellung). `-dno-print-pages` ist in Verbindung mit `-dpreview` nützlich.

`-h,--help`
Zeigt eine Zusammenfassung der Programmbenutzung und der Optionen.

`-H,--header=`*FELD*
: Gibt den Inhalt eines Feldes aus dem `\header`-Block in die Datei '`Dateiname.`*FELD*' aus.

`--include, -I=`*Verzeichnis*
: Fügt *Verzeichnis* zur Liste der Suchpfade hinzu.

`-i,--init=`*Initialisierungsdatei*
: Benutzt *Initialisierungsdatei* zur gesamten Programminitialisierung. Der Standardwert ist '`init.ly`'.

`-o,--output=`*DATEI*
: Schreibt das Ergebnis der Verarbeitung mit LilyPond in die Ausgabedatei *DATEI*. Die entsprechende Dateinamenserweiterung wird angehängt (z.B. `.pdf` für pdf).

`--ps`
: Erzeugt PostScript.

`--png`
: Erzeugt eine Grafik-Datei im PNG-Format von jeder Seite. Diese Option impliziert auch `--ps`. Die Auflösung in DPI der Grafik kann festgelegt werden durch

```
-dresolution=110
```

`--pdf`
: Erzeugt PDF-Dateien. Dies impliziert `--ps`.

`-j,--jail=`*Benutzer*`,`*Gruppe*`,`*Jail-Verzeichnis*`,`*Arbeitsverzeichnis*
: Führt `lilypond` in einem chroot-Jail aus.

  Die `--jail` Option ist eine flexiblere Alternative zu `--safe`, wenn LilyPond über das Internet verfügbar gemacht wird oder LilyPond-Quelldateien von Dritten automatisch vararbeitet werden.

  Sie funktioniert dergestalt, dass das Wurzelverzeichnis von `lilypond` auf *Jail-Verzeichnis* gesetzt wird, bevor die tatsächliche Kompilierung der .ly-Datei beginnt. Der Benutzer und die Gruppe werden auf die angegebenen Werte gesetzt und das aktuelle Arbeitsverzeichnis wird ebenfalls auf den angegebenen Wert *Arbeitsverzeichnis* gesetzt. Diese Einstellungen garantieren (zumindest in der Theorie), dass es nicht möglich ist, aus dem Jail auszubrechen. Damit `--jail` funktioniert, muss `lilypond` allerdings als root ausgeführt werden, was normalerweise auf sichere Art mit dem Kommando `sudo` erreicht werden kann.

  Das Jail-Verzeichnis zu erstellen ist etwas heikel, da LilyPond alle zur Ausführung nötigen Bibliotheken und Dateien *innerhalb des Jail-Verzeichnisses* finden muss. Ein typisches Setup besteht aus folgenden Punkten:

  Erstellung eines getrennten Dateisystems
  : Ein eigenes Dateisystem muss für LilyPond erstellt werden, sodass es mit sicheren Einstellungen wie `noexec`, `nodev` und `nosuid` eingebunden werden kann. Damit ist es unmöglich, Programme von diesem Dateisystem auszuführen oder direkt auf eine Hardware-Schnittstelle zuzugreifen. Wenn Sie keine eigene Partition erstellen möchten, können Sie auch eine Datei der entsprechenden Größe erstellen und sie als ‚loop'-Gerät einbinden. Ein getrenntes Dateisystem garantiert auch, dass LilyPond nicht mehr Festplattenspeicher benutzt als erlaubt.

  Erstellung eines eigenen Benutzerkontos
  : Es sollte ein eigener Benutzer und eine eigene Gruppe (z. B. `lily/lily`) mit geringen Rechten für die Ausführung von LilyPond innerhalb des Jails benutzt werden. Nur ein einziges Verzeichnis des Jails sollte für den Benutzer schreibbar sein und als *Arbeitsverzeichnis* an `lilypond` übergeben werden.

Einrichtung des Jails

LilyPond muss zahlreiche Dateien für die Ausführung einlesen. All diese Dateien müssen in das Jail-Verzeichnis kopiert werden (mit denselben Pfaden wie im tatsächlichen Wurzel-Dateisystem). Die gesamte LilyPond-Installation (typischerweise '`/usr/share/lilypond`') sollte kopiert werden.

Falls Probleme auftreten, ist es am einfachsten, Lilypond mittels `strace` zu starten, wodurch Sie relativ leicht feststellen können, welche Dateien im Jail noch fehlen.

Ausführung von LilyPond

In einem mit `noexec` eingebundenen Jail ist es nicht möglich, externe Programme auszuführen. Daher muss LilyPond auf eine Art gestartet werden, die keine weitere Ausführung von Programmen benötigt. Wie bereits erwähnt muss LilyPond mit Administrator-Rechten gestartet werden (die es allerdings sofort wieder abgibt), beispielsweise mittels `sudo`. Außerdem ist es eine gute Idee, die LilyPond zur Verfügung stehende CPU-Zeit zu limitieren (z. B. mit `ulimit -t`) und – falls das Betriebssystem dies unterstützt – auch den zur Verfügung stehenden Hauptspeicher.

`-v,--version`

Gibt die Versionsnummer aus.

`-V,--verbose`

Gibt ausführliche informative Meldungen aus: Zeigt die vollen Dateipfade aller gelesenen Dateien sowie Informationen über die Laufzeit.

`-w,--warranty`

Zeigt die Garantiebedingungen an, unter denen GNU LilyPond steht. (Es besteht **KEINERLEI GARANTIE**!)

3.2.3 Umgebungsvariablen

`lilypond` erkennt und benützt die folgenden Umgebungsvariablen:

`LILYPOND_DATADIR`

Diese Variable gibt das Verzeichnis an, wo Lilypond seine eigenen Dateien, Meldungen und Übersetzungen finden kann. Dieses Verzeichnis sollte Unterverzeichnisse '`ly/`', '`ps/`', '`tex/`', etc. beinhalten.

`LANG` Gibt die Sprache an, in der Warnungen und Fehlermeldungen ausgegeben werden.

`LILYPOND_GC_YIELD`

Mit dieser Variable (mit Werten zwischen 0 und 100) kann die Feinabstimmung zwischen dem Bedarf an Hauptspeicher und Rechenleistung bei der Ausführung von LilyPond durchgeführt werden. Bei höheren Werten benutzt LilyPond mehr Hauptspeicher, benötigt aber weniger Prozessor-Leistung. Bei niedrigeren Werten wird mehr Prozessor-Zeit, dafür weniger Hauptspeicher benötigt. Voreinstellung ist ein Wert von `70`.

3.3 Fehlermeldungen

Während der Verarbeitung einer Dateien können diverse Meldungen an der Kommandozeile auftreten:

Warnung Irgendetwas ist verdächtig. Wenn Sie etwas Ungewöhnliches in Ihrer Datei durchführen, dann werden Sie die Meldung verstehen und können sie gegebenenfalls

ignorieren. Im Normalfall jedoch bedeutet eine Warnung, dass mit Ihrer Datei etwas nicht stimmt, LilyPond jedoch trotzdem versucht, die Datei soweit wie möglich korrekt zu übersetzen.

Fehler Irgendetwas stimmt definitiv nicht. Der aktuelle Bearbeitungsschritt (Einlesen, Interpretieren oder Formatieren der Datei) wird noch fertig ausgeführt, danach bricht die Bearbeitung aber ab.

Fataler Fehler
Irgendetwas stimmt definitiv nicht und LilyPond kann nicht weiter ausgeführt werden. Dies ist nur sehr selten der Fall, meist sind die Schriftarten nicht korrekt installiert.

Scheme Fehler
Fehler, die während der Ausführung von Scheme-Code auftreten, werden vom Scheme-Interpreter aufgefangen und an der Kommandozeile ausgegeben. Wenn Sie LilyPond mit der `--verbose` Option (auch `-V`) ausführen, wird der sogennante ‚Call trace' ausgegeben, der die aufgerufenen Funktionen zur Zeit des Fehlers angibt.

Programmierfehler
Eine interne Inkonsistenz ist aufgetreten. Diese Fehlermeldungen sollen den Programmierern die Fehlersuche erleichtern und können meistens einfach ignoriert werden. In manchen Fällen werden so viele Meldungen ausgegeben, dass die Lesbarkeit der restliche Ausgabe davon beeinträchtigt wird.

Abgebrochen (core dumped)
Dies bezeichnet einen ernsten Programmierfehler, der das Programm zum Absturz gebracht hat. Solche Fehler werden als kritisch angesehen. Falls daher einer auftritt, senden Sie bitte einen Bug-Report!

Wenn Warnungen oder Fehlermeldungen mit einer konkreten Stelle in der Eingabedatei verknüpft werden können, dann hat die Meldung die folgende Form:

```
Dateiname:Zeile:Spalte: Meldung
Fehlerhafte Eingabezeile
```

Ein Zeilenumbruch wird in der fehlerhaften Zeile an jener Stelle eingefügt, wo der Fehler aufgetreten ist. Zum Beispiel

```
test.ly:2:19: Fehler: keine gültige Dauer: 5
  { c'4 e'
             5 g' }
```

Diese Stellen sind LilyPonds Vermutung, wo die Warnung oder der Fehler aufgetreten ist, allerdings treten Warnungen und Fehler ja gerade in unerwarteten Fällen auf. Manchmal kann Lilypond auch eine fehlerhafte Stelle zwar noch problemlos verarbeiten, ein paar Zeilen später wirkt sich der Fehler aber dann doch noch aus. In solchen Fällen, wo Sie in der angegebenen Zeile keinen Fehler erkennen, sollten Sie auch die Zeilen oberhalb der angegebenen Stelle genauer unter die Lupe nehmen.

3.4 Dateien mit `convert-ly` aktualisieren

Die Eingabesyntax von LilyPond ändert sich graduell um etwa die Eingabe zu erleichtern oder neue Funktionen zu ermöglichen. Ein Nebeneffekt hiervon ist, dass das LilyPond-Übersetzerprogramm nicht mehr mit älteren Eingabedateien kompatibel sein kann. Um dies zu umgehen, kann `convert-ly` benutzt werden, welches die meisten der Syntaxänderungen korrigieren kann.

Das Programm benötigt eine `\version`-Information in der Eingabedatei, um zu wissen, von welcher Version konvertiert werden soll. In den meisten Fällen genügt es, den Befehl

```
convert-ly -e meineDatei.ly
```

auszuführen.

Wenn keine Veränderungen in `meineDatei.ly` vorgenommen wurden und eine Datei `meineDatei.ly.NEW` erstellt wird, it `meineDatei.ly` schon aktuell.

3.4.1 Optionen von `convert-ly` auf der Kommandozeile

`convert-ly` konvertiert immer bis zur letzten Syntaxänderung, die es beherrscht. Das heißt, dass die neue `version`-Nummer in der Datei überlicherweise etwas niedriger ist als die Version von `convert-ly`.

Um LilyPond-Fragmente in texinfo-Dateien zu aktualisieren, gilt der Befehl:

```
convert-ly --from=... --to=... --no-version *.itely
```

Um die Änderungen der LilyPond-Syntax zwischen bestimmten Versionen zu sehen, kann der Befehl

```
convert-ly --from=... --to=... -s
```

benutzt werden.

Viele Dateien können auf einmal aktualisiert werden, indem `convert-ly` mit den üblichen UNIX-Befehlen kombiniert wird. Das folgende Beispiel aktualisiert alle `.ly`-Dateien im aktuellen Verzeichnis:

```
for f in *.ly; do convert-ly -e $f; done;
```

Die allgemeine Syntax des Programms lautet:

`convert-ly [`*`option`*`]... `*`file`*`...`

Folgende Optionen sind möglich:

`-e,--edit`
: Verändere die entsprechende Zeile der Eingabedatei. Überschreibt `--output`.

`-f,--from=`*`von-Versionsnummer`*
: Setze die Version, ab welcher konvertiert werden soll. Wenn diese Option nicht gesetzt ist, versucht `convert-ly`, die Version auf Grundlage von `\version` zu raten.

`-n,--no-version`
: Normalerweise fügt `convert-ly` eine `\version`-Zeile zur Ausgabe hinzu. Mit dieser Option kann das unterdrückt werden.

`-s, --show-rules`
: Zeige alle bekannten Konversionsregeln und beende.

`--to=`*`bis-Versionsnummer`*
: Hiermit kann das Ziel der Konversion gesetzt werden. Standard ist die neueste mögliche Version.

`-h, --help`
: Benutzerhilfe ausgeben.

3.4.2 Probleme mit `convert-ly`

Nicht alle Änderungen der Syntax können konvertiert werden. Nur eine Ausgabeoption kann angegeben werden. Scheme- und LilyPond-Scheme-Code wird sehr unwahrscheinlich korrekt aktualisiert, hier ist meistens Nacharbeit erforderlich.

```
Hier eine Liste einiger Befehle, die convert-ly nicht konvertieren kann.

Diese Liste ist in Form von Fehlermeldungen, denn convert-ly
ist so strukturiert, dass nicht alle benötigten Änderungen
```

```
implementiert werden können.  Es handelt sich also eher um eine
Wunschliste zur Referenz.

1.6->2.0:
 Bezifferter Bass wird nicht immer korrekt konvertiert, besonders
 Befehle wie {< >}.  Mats Kommentar hierzu:
    Um convert-ly korrekt ausführen zu können, müssen alle Vorkommen
    von '{<' in etwas beliebiges wie '{#' und genauso '>}' in '&}'
    geändert werden.  Nach der Konversion können sie dann zurück
    in '{ <' und '> }' verwandelt werden
 Nicht alle Textbeschriftungen werden richtig konvertiert.  In der
 alten Syntax  konnten Beschriftungsbefehle mit Klammern gruppiert
 werden, etwa
   -#'((bold italic) "string")
   This will incorrectly be converted into
   -\markup{{\bold italic} "string"}
   instead of the correct
   -\markup{\bold \italic "string"}
2.0->2.2:
 \partcombine wird nicht konvertiert
 \addlyrics => \lyricsto wird nicht konvertiert, dadurch kompilieren
 manche Dateien mit mehreren Strophen nicht.
2.0->2.4:
 \magnify wird nicht nach \fontsize konvertiert.
    - \magnify #m => \fontsize #f, where f = 6ln(m)/ln(2)
 remove-tag wird nicht verändert.
    - \applyMusic #(remove-tag '. . .) => \keepWithTag #'. . .
 first-page-number wird nicht geändert.
    - first-page-number no => print-first-page-number = ##f
 Zeilenumbrüche im \header-Feld werde nicht konvertiert.
    - \\\\  als Zeilenumbruch in \header{...} => \markup \center-align <
      "First Line" "Second Line" >
 Crescendo- und Decrescendo-Enden werden nicht konvertiert.
    - \rced => \!
    - \rc => \!
2.2->2.4:
 \turnOff (in \set Staff.VoltaBracket = \turnOff eingesetzt) wird nicht korrekt behandelt.
2.4.2->2.5.9
 \markup{ \center-align <{ ... }> } sollte umgewandelt werden in:
 \markup{ \center-align {\line { ... }} }
 aber im Moment fehlt \line.
2.4->2.6
 Besondere LaTeX-Zeichen wie $~$ im Text werden nicht nach UTF(
 konvertiert.
2.8
 \score{} muss jetzt mit einem musikalischen Ausdruck beginnen.
 Alles andere (insbesondere \header{}) darf erst nach den Noten
 kommen.
```

3.5 Fehler melden

Wenn Sie eine Datei haben, die zu einem Absturz von LilyPond oder zu einer fehlerhaften Ausgabe führt, so ist dies ein ‚Bug'. Die List der aktuell bekannten derartigen Fehler findet sich in unserem ‚Google Bug Tracker':

http://code.google.com/p/lilypond/issues/list

Wenn Sie einen Fehler gefunden haben, der noch nicht aufgelistet ist, melden Sie dies bitte anhand der Anweisungen auf der Seite

http://lilypond.org/web/devel/participating/bugs

Wenn Sie Beispieldateien für den Fehler erstellen, versuchen Sie bitte, die Datei möglichst minimal zu halten und nur jenen LilyPond Code aufzunehmen, der auch wirklich für den Fehler verantwortlich ist. Meldungen mit Beispieldateien, die nicht minimal sind, können wir meist aus Zeitgründen nicht effektiv bearbeiten.

4 lilypond-book: Noten in den Text integrieren

Wenn Sie in ein Dokument Grafiken Ihres Musiksatzes einfügen möchten, so können Sie genauso machen, wie Sie andere Grafiken einfügen würden: Die Bilder werden getrennt vom Dokument im PostScript- oder PNG-Format erstellt und können dann in LaTeX oder HTML inkludiert werden.

`lilypond-book` automatisiert diesen Prozess: Dieses Programm extrahiert Musik-Schnipsel aus Ihrem Dokument, ruft `lilypond` auf und fügt die resultierenden Bilder in Ihr Dokument ein. Die Länge der Zeilen und die Schriftgroße werden dabei Ihrem Dokument angepasst.

`lilypond-book` ist ein eigenständiges Programm und wird üblicherweise von der Kommandozeile aufgerufen. Nähere Informationen hierzu finden sich in Abschnitt 3.2 [Benutzung auf der Kommandozeile], Seite 6. Wenn Sie MacOS 10.3 oder 10.4 benutzen und Probleme mit `lilypond-book` haben, lesen Sie Abschnitt 2.1.1 [Einrichtung für MacOS X], Seite 2.

Dieses Vorgehen kann bei LaTeX, HTML, Texinfo oder DocBook Dokumenten angewendet werden.

4.1 Ein musikwissenschaftlicher Text als Beispiel

Zahlreiche Texte enthalten Musikbeispiele: musikwissenschaftliche Abhandlungen, Liederbücher oder Handbücher wie dieses. Solche Texte können händisch erzeugt werden, indem einfach die Musikbeispiele als Grafik (PostScript, PNG, GIF, etc.) im Textverarbeitungsprogramm eingefügt werden. Für HTML, LaTeX, Texinfo und DocBook Dokumente existiert jedoch ein Weg, dies automatisiert durchzuführen.

Das Programm `lilypond-book` extrahiert die Musikfragmente aus dem Dokument, formatiert sie automatisiert in eine Grafik und fügt die resultierenden Notenbeispiele dann wieder in das Dokument ein. Dies soll hier an einem einfachen LaTeX-Beispiel verdeutlicht werden. Das Beispiel selbst enthält schon Erklärungen, sodass wir es hier nicht weiter diskutieren müssen.

Eingabe

```
\documentclass[a4paper]{article}

\begin{document}

Dokumente für \verb+lilypond-book+ können Musik und Text nach Belieben
kombinieren. Zum Beispiel:

\begin{lilypond}
\relative c' {
  c2 g'2 \times 2/3 { f8 e d } c'2 g4
}
\end{lilypond}

Optionen für \verb+lilypond+ werden dabei in eckige Klammern gesetzt.

\begin[fragment,quote,staffsize=26,verbatim]{lilypond}
  c'4 f16
\end{lilypond}

Größere Beispiele können auch in einer eigenständigen Datei gespeichert
und dann mit \verb+\lilypondfile+ eingebunden werden.
```

```
\lilypondfile[quote,noindent]{screech-boink.ly}

(Falls nötig kann screech-boink.ly durch eine beliebige andere .ly
Datei im selben Verzeichnis wie diese Datei ersetzt werden.)

\end{document}
```

Verarbeitung

Speichern Sie den obigen LaTeX Quellcode in eine Datei '`lilybook.lytex`' und führen Sie dann in der Kommandozeile folgende Befehle aus:

```
lilypond-book --output=out --pdf lilybook.lytex
lilypond-book (GNU LilyPond) 2.12.1

Reading lilybook.lytex...
..(viele Ausgabezeilen entfernt)..
Compiling lilybook.tex...
cd out
pdflatex lilybook
..(viele Ausgabezeilen entfernt)..
xpdf lilybook
(Ersetzen Sie xpdf durch Ihren PDF-Betrachter)
```
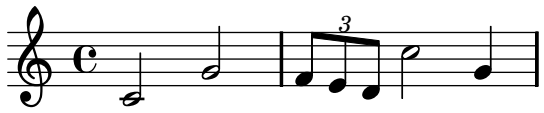
Die Ausführung von `lilypond-book` und `latex` erzeugt zahlreiche temporäre Dateien, die das Arbeitsverzeichnis unnötig vollstopfen würden. Daher empfiehlt sich die Benutzung der `--output=dir` Option, wodurch die Dateien im Unterverzeichnis '`dir`' erzeugt werden.

Das Endresultat des obigen LaTeX Beispiels ist im nächsten Abschnitt zu sehen.[1]

[1] Da dieses Handbuch mit Texinfo erzeugt wurde, kann sich das Aussehen des Beispiels leicht von dem mit LaTeX erzeugten unterscheiden.

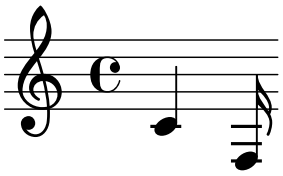
Ausgabe

Dokumente für `lilypond-book` können Musik und Text nach Belieben kombinieren. Zum Beispiel:



Optionen für `lilypond` werden dabei in eckige Klammern gesetzt.

```
c'4 f16
```

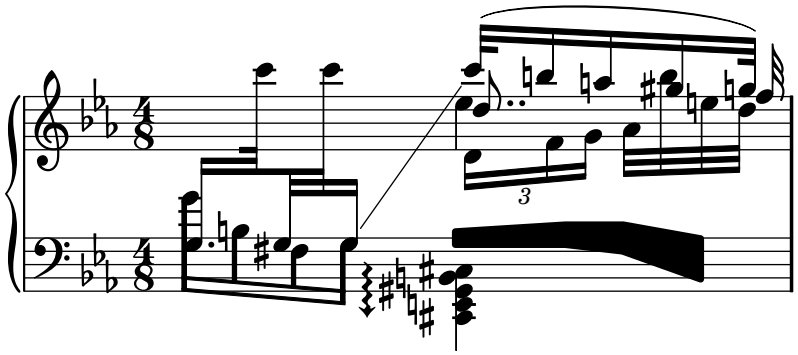


Größere Beispiele können auch in einer eigenständigen Datei gespeichert und dann mit `\lilypondfile` eingebunden werden.

Screech and boink

Random complex notation

Han-Wen Nienhuys



4.2 Noten in Text integrieren

In diesem Abschnitt soll die Integration von LilyPond mit den verschiedenen Dateiformaten detailliert erläutert werden.

4.2.1 LaTeX

LaTeX ist der de-facto Standard zur Publikation von wissenschaftlichen Texten in Naturwissenschaft und Technik. Es basiert auf dem Schriftsetzer TeX, der die bestmögliche Typographie erzeugt.

Siehe die *LaTeX2e-Kurzbeschreibung* für eine Einführung in die Benutzung von LaTeX.

Musikbeispiele können eingegeben werden als

```
\begin[Optionen,kommen,]{lilypond}
  IHR LILYPOND QUELLCODE
\end{lilypond}
```

oder

```
\lilypondfile[Optionen,kommen,hier]{Dateiname}
```

oder
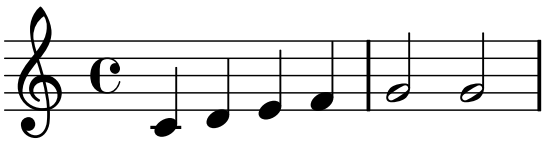
```
\lilypond{ IHR LILYPOND QUELLCODE }
```

Zusätzlich kann mit `\lilypondversion` die benutzte Versionsnummer von LilyPond angezeigt werden. Der Aufruf von `lilypond-book` liefert eine Datei, die dann mit LaTeX weiter verarbeitet werden kann.

Dies soll hier an einigen Beispielen gezeigt werden. Die `lilypond`-Umgebung

```
\begin[quote,fragment,staffsize=26]{lilypond}
  c' d' e' f' g'2 g'2
\end{lilypond}
```

erzeugt



Die Kurzversion

```
\lilypond[quote,fragment,staffsize=11]{<c' e' g'>}
```

erzeugt



Innerhalb des `\lilypond{}` Befehls dürfen keine geschwungenen Klammern `{` oder `}` vorkommen, weshalb dieser Befehl nur mit der `fragment` Option Sinn macht.

Die Standardzeilenlänge wird bestimmt, indem die Angaben in der Dokumentpräambel, also dem Teil der LaTeX Datei vor dem `\begin{document}`, analysiert werden. Der `lilypond-book` Befehl sendet diese Angaben an LaTeX, um herauszufinden, wie breit der Text tatsächlich ist. Die Breite der Notenzeilen wird dann an diese Textbreite angepasst. Ein derartig heuristischer Algorithmus kann natürlich auch versagen, wobei man in diesem Fall die Breite auch explizit durch die `line-width` Option des `\lilypond{}` oder `\begin{lilypond}` Befehls angeben kann.

Jedes Musikbeispiele ruft die folgenden Makros auf, wenn sie vom Benutzer definiert wurden:

- `\preLilyPondExample` – wird vor der Musik aufgerufen,

- \postLilyPondExample – wird nach der Musik aufgerufen,
- \betweenLilyPondSystem[1] – wird zwischen den einzelnen Systemen aufgerufen, wenn lilypond-book das Beispiel in verschiedene PostScript Dateien getrennt hat. Dieser LaTeX-Befehl muss so definiert werden, dass er genau ein Argument erhält, nämlich die Zahl der bereits in LaTeX eingefügten Dateien dieses Beispiels. Als Standard wird einfach ein \linebreak eingefügt.

Ausgewählte Schnipsel

Manchmal ist es nötig, Musikelemente wie Halte- oder Bindebögen so darzustellen, als ob sie am Ende des Musikausschnitten noch weitergehen würden. Eine solche Ausgabe kann erreicht werden, indem ein Zeilenumbruch in die Notenzeile eingefügt wird und die Ausgabe der folgenden Notenzeile unterdrückt wird.

In LaTeX wird dazu der Befehl \betweenLilyPondSystem einfach derartig programmiert, dass die Ausgabe der einzelnen Notensysteme abgebrochen wird, sobald die gewünschte Anzahl an Systemen erreicht ist. Da \betweenLilyPondSystem zum ersten Mal nach dem ersten System aufgerufen wird, ist die Ausgabe nur eines Systems trivial.

```
\def\betweenLilyPondSystem#1{\endinput}

\begin[fragment]{lilypond}
  c'1\( e'( c'~ \break c' d) e f\)
\end{lilypond}
```

Um eine größere Zahl an System nötig, dann muss dementsprechend eine TeX-Bedingung vor dem \endinput benutzt werden:

```
\def\betweenLilyPondSystem#1{
    \ifnum##1<2\else\endinput\fi
}
```

Dieses Beispiel bricht nach genau zwei ausgegebenen Notenzeilen ab. Für eine andere Anzahl braucht nur ‚2‘ durch die entsprechende Anzahl ersetzt werden.

Die Definition von \betweenLilyPondSystem bleibt gültig, bis TeX die aktuelle Umgebung in LaTeX verlässt oder der Befehl durch eine neue Definition überschrieben wird. Dies kann etwa folgendermaßen in der LaTeX-Datei geschehen:

```
\let\betweenLilyPondSystem\undefined
```

Obige Definition von \betweenLilyPondSystem kann durch die Definition eines TeX-Makros auch verallgemeinert werden,

```
\def\onlyFirstNSystems#1{
    \def\betweenLilyPondSystem##1{\ifnum##1<#1\else\endinput\fi}
}
```

wobei diesem Makro \onlyFirstNSystems einfach die Anzahl der gewünschten Systeme übergeben wird:

```
\onlyFirstNSystems{3}
\begin{lilypond}...\end{lilypond}
\onlyFirstNSystems{1}
\begin{lilypond}...\end{lilypond}
```

Siehe auch

lilypond-book stellt auch zahlreiche Kommandozeilen-Optionen zur Verfügung. Für eine Liste dieser Optionen und andere hilfreiche Details zur Verarbeitung von LaTeX-Dokumenten, siehe Abschnitt 4.4 [lilypond-book aufrufen], Seite 25.

4.2.2 Texinfo

Texinfo ist das Standard-Dokumentationsformat des GNU Projekts. Ein Beispiel für ein Dokument im Texinfo Format ist dieses Handbuch, wobei die HTML-, PDF- und Info-Versionen alle aus demselben Texinfo Dokument erzeugt werden.

In der Eingabedatei wir Musik eingegeben als

```
@lilypond[Optionen,kommen,hier]
  IHR LILYPOND QUELLCODE
@end lilypond
```

oder

```
@lilypond[Optionen,kommen,hier]{ IHR LILYPOND QUELLCODE }
```

oder
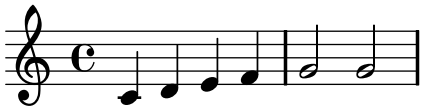
```
@lilypondfile[Optionen,kommen,hier]{Dateiname}
```

Zusätzlich kann mit `@lilypondversion` die aktuelle Versionsnummer von LilyPond angezeigt werden. Wenn `lilypond-book` eine derartige Datei verarbeitet, wird eine Texinfo-Datei mit der Erweiterung '`.texi`' erzeugt, die `@image` Befehle für die Ausgabe nach HTML, Info und PDF enthält. `lilypond-book` erzeugt die entsprechenden Grafiken der Musikbeispiele im EPS- und PDF-Format für die Ausgabe nach PDF und im PNG-Format für die Ausgabe nach HTML und Info.

Hier sollen zwei einfache Beispiele gezeigt werden. Eine `lilypond` Umgebung

```
@lilypond[fragment]
c' d' e' f' g'2 g'
@end lilypond
```

erzeugt



Die Kurzversion

```
@lilypond[fragment,staffsize=11]{<c' e' g'>}
```

erzeugt
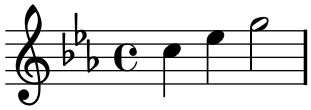


Im Gegensatz zu LaTeX erzeugt `@lilypond{...}` allerdings keine Grafik im Fließtext, sondern setzt sie immer in einen eigenen Absatz.

4.2.3 HTML

Musik wird eingegeben als

```
<lilypond fragment relative=2>
\key c \minor c4 es g2
</lilypond>
```

`lilypond-book` erzeugt dann daraus eine HTML-Datei mit den entsprechenden `<image>` Tags für die Musikbeispiele in jeweils einem eigenen Absatz.



Für Grafiken im Fließtext kann `<lilypond ... />` benutzt werden, wobei die Optionen durch einen Doppelpunkt von der Musik getrennt angegeben werden.

```
Musik <lilypond relative=2: a b c/> in derselben Zeile.
```

Um Dateien mit Musik einzubinden, kann folgendermaßen vorgegangen werden:

```
<lilypondfile Option1 Option2 ...>Dateiname</lilypondfile>
```

Zusätzlich gibt `<lilypondversion/>` die aktuelle Versionsnummer von LilyPond aus.

4.2.4 DocBook

Bei der Einbindung von Musik im LilyPond-Format in DocBook soll die Konformität unseres DocBook Dokuments erhalten bleiben und damit die Bearbeiten mit DocBook-Editoren sowie die Validierung weiter möglich bleiben. Aus diesem Grund werden in DocBook keine eigenen Tags wie in HTML benutzt, sondern die von den vorhandenen DocBook-Elementen vorgegebenen Konventionen entsprechend benützt.

Definitionen

Für die Einbindung von LilyPond Code werden in allen Fällen die `mediaobject` und `inlinemediaobject` Elemente benutzt, die unsere Beispiele in einem eigenen Absatz oder im Fließtext einfügen. Die Optionen zur Formatierung mit LilyPond werden dabei in der `role` Eigenschaft des innersten Elements angegeben, wie im nächsten Abschnitt gezeigt wird. Die DocBook Datei, die dann von `lilypond-book` verarbeitet wird, sollte der Klarheit halber die Dateierweiterung '`.lyxml`' (jedenfalls nicht '`.xml`') besitzen.

Eine LilyPond-Datei einfügen

Dies ist der einfachste Fall: Die LilyPond-Datei besitzt die Erweiterung '`.ly`' und wird einfach als `imageobject` eingebettet:

```
<mediaobject>
  <imageobject>
    <imagedata fileref="music1.ly" role="printfilename" />
  </imageobject>
</mediaobject>
```

Für das äußerste Element kann je nach Bedarf `mediaobject` oder `inlinemediaobject` benutzt werden.

LilyPond-Code einfügen

Die Einbindung von LilyPond-Code direkt in der DocBook-Datei ist durch die Benutzung von `programlisting` möglich, wobei die Sprache auf `lilypond` gesetzt wird:

```
<inlinemediaobject>
  <textobject>
    <programlisting language="lilypond" role="fragment verbatim staffsize=16 ragged-rig
\context Staff \with {
  \remove Time_signature_engraver
  \remove Clef_engraver}
  { c4( fis) }
    </programlisting>
  </textobject>
</inlinemediaobject>
```

Das äußerste Element ist also `mediaobject` oder `inlinemediaobject`, welches ein `textobject` mit dem `programlisting` enthält.

Ein DocBook-Dokument übersetzen

Der Aufruf von `lilypond-book` mit der '`.lyxml`'-Datei erzeugt ein gültiges DocBook-Dokument mit der Erweiterung '`.xml`', welches normal weiterverarbeitet werden kann. Bei Benutzung von dblatex wird daraus automatisch eine PDF-Datei erzeugt. Für die Erzeugung von HTML (HTML Hilfe, JavaHelp, etc.) können die offiziellen DocBook XSL-Stylesheets benutzt werden. Eventuell müssen dafür allerdings kleinere Anpassungen vorgenommen werden.

4.3 Die Musikfragment-Optionen

Im Folgenden meint ‚LilyPond-Kommando' ein beliebiges in den vorgehenden Abschnitten beschriebenes Kommando, welches ein LilyPond-Fragment in eine Datei einfügt und von `lilypond-book` verarbeitet wird. Der Einfachheit halber werden hier alle LilyPond-Kommandos in der Syntax von LaTeX dargestellt.

Zu beachten ist, dass die Optionen eines LilyPond-Kommandos von links nach rechts verarbeitet werden. Wenn eine Option also mehrfach angegeben wird, wird nur die letzte benutzt.

Die folgenden Optionen können für LilyPond-Kommandos benutzt werden:

`staffsize=`*ht*
: Setzt die Höhe einer Notenzeile auf *ht*, angegeben in Punkten.

`ragged-right`
: Erzeugt Notenzeilen im Flattersatz mit natürlichem Abstand der Noten. In anderen Worten: `ragged-right = ##t` wird in das Musikfragment eingefügt. Dies ist die Standardeinstellung für das `\lilypond{}` Kommando, wenn die Option `line-width` nicht angegeben wird. Ebenso ist dies die Standardeinstellung für die `lilypond`-Umgebung, wenn die Option `fragment`, aber keine Zeilenlänge explizit angegeben ist.

`noragged-right`
: Streckt Musikfragmente mit nur einer Notenzeile auf die volle Breite, es wird also `ragged-right = ##f` in das Musikfragment eingefügt.

`line-width`
`line-width=`*Breite*`\`*Einheit*
: Setzt die Breite der Notenzeilen auf *Breite*, gemessen in Vielfachen der *Einheit*. Als Einheit können die folgenden Zeichenfolgen angegeben werden: `cm`, `mm`, `in` oder `pt`. Diese Option hat nur Einfluss auf die Breite von Notenzeilen und Text im Musikfragment, nicht jedoch auf den restlichen Text des Dokuments.

  Wird diese Option ohne einen Wert angegeben, wird die Zeilenbreite auf einen Standardwert gesetzt, der durch einen heuristischen Algorithmus bestimmt wird.

  Wenn die `line-width` Option nicht angegeben wird, versucht `lilypond-book` einen geeigneten Standardwert für alle `lilypond`-Umgebungen zu finden, die die `ragged-right` Option nicht benutzen.

`notime`
: Verhindert die Ausgabe der Taktangabe am Anfang des Fragments und schaltet Taktstriche und alle Taktangaben im Fragment ab.

`fragment`
: Bewirkt, dass `lilypond-book` Standardcode um das Fragment herum einfügt, sodass z. B.

```
c'4
```

  ohne `\layout`, `\score`, etc. eingegeben werden kann.

`nofragment`
: Verhindert das Einfügen von Standardcode um das Fragment herum. Dies ist die Standardeinstellung, insofern ist diese Option im Normalfall unnötig.

`indent=`*`Einzug`*`\`*`Einheit`

Setzt den Einzug des ersten Notensystems auf *Einzug*, gemessen in Vielfachen der *Einheit*. Als Einheit können die folgenden Zeichenfolgen angegeben werden: `cm`, `mm`, `in` oder `pt`. Diese Option hat nur Einfluss auf den Einzug von Notenzeilen und Text im Musikfragment, nicht jedoch auf den restlichen Text des Dokuments.

`noindent` Setzt den Einzug des ersten Notensystems auf 0. Diese Option hat nur Einfluss auf den Einzug von Notenzeilen und Text im Musikfragment, nicht jedoch auf den restlichen Text des Dokuments. Dies ist die Standardeinstellung, insofern ist diese Option im Normalfall unnötig.

`quote` Verringert die Zeilenlänge des Musikfragments um $2 * 0.4$ in und setzt das Fragment in einen Zitat-Block. Der Wert von ‚0.4 in‘ kann durch die `exampleindent` Option angepasst werden.

`exampleindent`

Setzt den Betrag, um den das Fragment bei Benutzung der `quote` Option eingerückt wird.

`relative`
`relative=`*`n`*

Benutzt relative Oktavenbezeichnungen. Standardmäßig werden Noten relativ zum mittleren C angegeben. Das optionale ganzzahlige Argument gibt die Oktave der ersten Note an, wobei die Standardeinstellung von `1` das mittlere C bedeutet. Die `relative` Option macht nur Sinn in Verbindung mit der `fragment` Option, weshalb `fragment` automatisch durch die Angabe der `relative` Option impliziert wird. Eine explizite Angabe der `(no)fragment` Option hat keinen Effekt.

LilyPond benutzt zur Erzeugung seiner eigenen Dokumentation ebenfalls `lilypond-book`. Zu diesem Zweck stehen noch zahlreiche spezialisierte Optionen zur Verfügung:

`verbatim` Der LilyPond-Code im LilyPond-Kommando wird zum einen benutzt, um das Musikfragment in eine Grafik mit schönem Notensatz zu konvertieren, andererseits aber auch wörtlich in das Dokument eingefügt. Dies geschieht in einem ‚verbatim‘-Block, gefolgt vom Text einer möglicherweise angegebenen `intertext` Option[1] und der Grafik des tatsächlichen Notensatzes. Diese Option funktioniert nur fehlerhaft, wenn `\lilypond{}` im Fließtext benutzt wird.

Wird `verbatim` in Verbindung mit einem `lilypondfile`-Kommando benutzt, so ist es auch möglich, nur ein Teil der Datei wörtlich einfügen zu lassen: Wenn die eingebundene LilyPond-Datei ein Kommentar mit dem Inhalt ‘`begin verbatim`’ (ohne Anführungszeichen) enthält, wird nur der Dateiinhalt ab dieser Position eingefügt. Enthält die Datei mehrere solche Kommentare, wirkt nur das letzte. Analog wird nur der Dateiinhalt bis zu einem etwaigen Kommentar mit dem Inhalt ‘`end verbatim`’ eingefügt. Im folgenden Beispiel wird das gesamte Musik für die Erzeugung der Grafik im relativen Oktavenmodus interpretiert, der wörtlich in das Dokument kopierte LilyPond-Code zeigt den `relative`-Befehl jedoch nicht.

```
\relative c' { % begin verbatim
  c4 e2 g4
  f2 e % end verbatim
}
```

erzeugt ein Zitat der Form

```
  c4 e2 g4
  f2 e
```

[1] Die `intertext` Option ist noch nicht implementiert.

Wenn Kommentare und Variablen im Zitat, aber nicht im Quelltext übersetzt werden sollen, kann die Umgebungsvariable LYDOC_LOCALEDIR auf einen Verzeichnispfad gesetzt werden. Das Verzeichnis sollte einen Baum an '.mo'-Nachrichtenkatalogen beinhalten mit lilypond-doc als Domain.

addversion

(Nur innerhalb von Texinfo-Dateien.) Stellt \version @w{"@version{}"} an den Beginn des Fragments der Ausgabe mit verbatim.

texidoc (Nur innerhalb von Texinfo-Dateien.) Wird lilypond mit der Kommandozeilenoption '--header=texidoc' für eine Datei 'foo.ly' und enthält die Datei ein texidoc-Feld im \header-Block, so wird dessen Inhalt in die Datei 'foo.texidoc' ausgegeben. Die texidoc Option veranlasst lilypond-book, den Inhalt dieser '.texidoc' Dateien innerhalb eines Dokumentationsblocks direkt vor dem Musikfragment in das aktuelle Dokument einzufügen.

Enthält also die Datei 'foo.ly' etwa den LilyPond-Code

```
\header {
  texidoc = "Dieses Beispiel zeigt eine einzelne Note."
}
{ c'4 }
```

und das Texinfo-Dokument 'text.texinfo'

```
@lilypondfile[texidoc]{foo.ly}
```

so liefert der folgende Aufruf von lilypond-book das gewünschte Ergebnis:

```
lilypond-book --pdf --process="lilypond \
  -dbackend=eps --header=texidoc" test.texinfo
```

Die meisten Test-Dateien (im 'input/' Verzeichnis von LilyPond) sind kleine '.ly' Dateien von genau dieser Form.

Auch die Übersetzung dieser zusätzlichen Kommentare ist möglich: Dazu muss das Texinfo-Dokument den Befehl @documentlanguage *LANG* und der \header Block in der Datei 'foo.ly' die Übersetzung im Feld texidoc*LANG* enthalten. Wird nun lilypond mit der Option '--header=texidoc*LANG*' aufgerufen, so wird der Inhalt der Datei 'foo.texidoc*LANG*' anstelle von 'foo.texidoc' eingefügt.

lilyquote

(Nur innerhalb von Texinfo-Dateien.) Diese Option wirkt wie die quote Option, außer dass nur das Musikfragment (und natürlich optional der gesamte LilyPond-Code bei Benutzung von verbatim) in den Zitatblock eingefügt wird. Dies ist nützlich, wenn Sie ein Musikfragment zitieren möchten, nicht aber dessen texidoc Dokumentationsblock.

doctitle (Nur innerhalb von Texinfo-Dateien.) Diese Option wirkt ähnlich wie die texidoc Option: Wenn lilypond mit der Option '--header=doctitle' aufgerufen wird und die Eingabedatei 'foo.ly' ein Feld doctitle im \header-Block enthält, wird dessen Wert in die Datei 'foo.doctitle' geschrieben. Wird die doctitle Option für ein Musikfragment benutzt, so wird der Inhalt dieser Datei, der eine einzelne Textzeile sein sollte, im Texinfo-Dokument als @lydoctitle *Text* eingefügt. @lydoctitle muss allerdings in Ihrem Texinfo-Dokument als Makro selbst definiert werden. Die Übersetzung funktioniert völlig analog zu texidoc.

nogettext

Nur für Texinfo-Ausgabe: Kommentare und Variablenbezeichnungen im zitierten Quelltext des Schnipsel werden nicht übersetzt.

`printfilename`
: Wenn eine LilyPond-Datei mittels `\lilypondfile` und dieser Option eingebunden wird, wird der Dateiname (ohne die Pfadangabe) unmittelbar vor dem Musikfragment ausgegeben. In HTML-Dateien ist er außerdem ein Link auf die LilyPond-Datei.

`fontload`
: Diese Option inkludiert die Schriftarten in allen EPS-Dateien, die von diesem Fragment erzeugt werden. Dies ist nötig, wenn das Fragment Schriftarten benutzt, die LaTeX nicht selbst findet.

4.4 `lilypond-book` aufrufen

`lilypond-book` erzeugt abhängig vom Ausgabeformat eine Datei mit einer der folgenden Dateierweiterungen: '`.tex`', '`.texi`', '`.html`' oder '`.xml`'. Alle '`.tex`', '`.texi`' und '`.xml`' Dateien müssen noch mit den entsprechenden Programmen (LaTeX, DocBook, etc.) weiter verarbeitet werden, um druckfähige Dateien zu erhalten.

Formatabhängige Anweisungen

LaTeX

Es existieren zwei Methoden, Ihr LaTeX-Dokument weiter zu verarbeiten, um zu einer druck- oder publikationsfähigen Datei zu gelangen: Zum einen die direkte Erzeugung einer PDF-Datei mit PDFLaTeX, zum anderen die Erzeugung einer DVI daraus einer PostScript-Datei mittels LaTeX und einem DVI-nach-PostScript Konverters wie `dvips`. Die erste Methode ist einfacher und daher empfehlenswert.[2] Welche Methode auch immer benutzt wird, die Konvertierung zwischen PostScript und PDF kann leicht mit Hilfsprogrammen wie `ps2pdf` und `pdf2ps` (aus dem Ghostscript Paket) erfolgen.

Um eine PDF-Datei mittels PDFLaTeX zu erzeugen, kann folgendermaßen vorgegangen werden:

```
lilypond-book --pdf Ihre_Datei.pdftex
pdflatex Ihre_Datei.tex
```

Um eine PDF-Datei mittels LaTeX/`dvips`/`ps2pdf` zu erhalten, sind folgende Befehle nötig:

```
lilypond-book Ihre_Datei.lytex
latex Ihre_Datei.tex
dvips -Ppdf Ihre_Datei.dvi
ps2pdf Ihre_Datei.ps
```

Die '`.dvi`'-Datei, die beim Aufruf von `latex` erzeugt wird, scheint keine Notenköpfe zu enthalten, was in Ordnung ist. Wenn Sie die Datei wie beschrieben weiter verarbeiten, erscheinen die Notenköpfe korrekt in den '`.ps`' und '`.pdf`' Dateien.

Der Aufruf von `dvips` kann einige Warnungen über fehlende Schriftarten ausgeben. Auch dies ist in Ordnung und kann ignoriert werden.

Wenn Sie in der LaTeX-Datei das Papierformat auf Querformat eingestellt haben, vergessen Sie nicht auf die `-t landscape` Option beim Aufruf von `dvips`.

Texinfo

Um ein von `lilypond-book` erzeugtes Texinfo-Dokument zu verarbeiten, gehen Sie wie für alle anderen Texinfo-Dokumente vor: Rufen Sie – abhängig vom gewünschten Ausgabeformat – eines der Programme `texi2pdf`, `texi2dvi`, `makeinfo` oder `texi2html` auf.

Die Dokumentation von Texinfo liefert dazu nähere Informationen.

[2] Manchmal kann eine Datei entweder von PDFLaTeX oder von LaTeX nicht korrekt verarbeitet werden, weshalb hier beide Methoden beschrieben werden.

Optionen auf der Kommandozeile

lilypond-book unterstützt die folgenden Kommandozeilenoptionen:

`-f` *Format*
`--format=`*Format*
: Gibt das Format des Eingabedokuments an: `html`, `latex`, `texi` (Standardeinstellung), `texi-html` oder `docbook`. Ist diese Option nicht angegeben, versucht `lilypond-book` das Format anhand des Dateinamens zu bestimmen. Im Moment bedeutet `texi` praktisch dasselbe wie `texi-html`.

`-F` *Filter*
`--filter=`*Filter*
: Leitet die Musikfragmente durch das Programm *filter* anstatt sie mit Lilypond zu einer Grafik zu verarbeiten. '`--filter`' und '`--process`' kann nicht gleichzeigt benutzt werden. Beispielaufruf:

```
lilypond-book --filter='convert-ly --from=2.0.0 -' Mein-Buch.tely
```

`-h`
`--help`
: Gibt eine kurze Hilfemeldung aus.

`-I` *Pfad*
`--include=`*Pfad*
: Fügt *Pfad* zu den Include-Pfaden hinzu. `lilypond-book` sucht auch in allen Include-Pfaden nach bereits erstellten Grafiken für die Musikfragmente. Wird das aktuelle Fragment gefunden und hat sich seit der letzten Erstellung nicht geändert, wird es nicht erneut erzeugt. Bei manchen der Programme zur Weiterverarbeitung wie etwa `makeinfo` oder `latex` muss dieselbe `-I` *`Pfad`* Option angegeben werden, damit das entsprechende Programm die Grafiken ebenso findet.

`-o` *Verzeichnis*
`--output=`*Verzeichnis*
: Erzeugt die Ausgabedateien in *Verzeichnis*. Der Aufruf von `lilypond-book` erzeugt zahlreiche kleine Dateien, die von LilyPond, `latex`, `makeinfo` etc. dann weiter benützt werden. Um zu vermeiden, dass das Quellenverzeichnis durch diese Dateien unübersichtlich wird, kann die '`--output`' Option benutzt werden. Vor dem Aufruf von `latex` oder `makeinfo` sollten Sie in dieses Verzeichnis wechseln.

```
lilypond-book --output=out IhreDatei.lytex
cd out
...
```

`--skip-lily-check`
: Nicht mit einer Fehlermeldung abbrechen, wenn keine Ausgabe von LilyPond gefunden wird. Dies wird benutzt für Dokumentation ohne Grafiken.

`--skip-png-check`
: Nicht mit einer Fehlermeldung abbrechen, wenn für die EPS-Dateien keine PNG-Grafiken gefunden werden. Dies wird benutzt für Dokumentation ohne Grafiken.

`--lily-output-dir=`*Verzeichnis*
: Schreibt '`lily-XXX`' Dateien nach *Verzeichnis* und erzeugt im mit `--output` angegebenen Verzeichnis Verknüpfungen darauf. Diese Option ist nützlich, um

Zeit zu sparen, wenn Dokumente in verschiedenen Verzeichnissen viele identische Musikfragmente enthalten.

`--info-images-dir=`*Verzeichnis*
: Formatiert die Texinfo-Ausgabe dergestalt, dass Info in *Verzeichnis* nach den Grafiken zu den Musikfragmenten sucht.

`--latex-program=`*Programm*
: Führt `Programm` anstelle von `latex` aus. Dies ist nützlichen, wenn das Dokument mit einer anderen LaTeX-Variante wie etwa `xelatex` verarbeitet werden soll.

`--left-padding=`*Einrückung*
: Fügt *Einrückung* als zusätzlichen Einzug in die EPS-Box ein. *Einrückung* wird in Millimetern angegeben, die Standardeinstellung is 3.0 Millimeter. Diese Option kann benutzt werden, wenn die Notenzeilen über den Rand des Dokuments hinausstehen.

  Die Breite eines eng ausgeschnittenen Notensystems kann variieren aufgrund von Notationselementen, die über den linken Rand hinausstehen, wie etwa Taktzahlen und Bezeichnungen der Instrumente. Diese Option verkürzt die Notenzeile und verschiebt sie um denselben Betrag nach rechts.

`-P` *Kommando*
`--process=`*Kommando*
: Verarbeitet LilyPond-Fragmente mit *Kommando* anstelle des Standardbefehls `lilypond`. `--filter` und `--process` können nicht gleichzeitig angegeben werden.

`--pdf` Erzeugt PDF-Dateien mit PDFLaTeX.

`-V`
`--verbose`
: Gibt ausführliche informative Meldungen aus.

`-v`
`--version`
: Gibt die Versionsnummer aus.

Bekannte Probleme und Warnungen

Der Texinfo-Befehl `@pagesizes` wird ignoriert. Ebenso werden LaTeX-Befehle ignoriert, die den Seitenrand oder die Zeilenlänge nach der Dokumentpräambel verändern.

Nur der erste `\score`-Block eines LilyPond-Fragments wird verarbeitet.

4.5 Dateiendungen

Für die Eingabedatei kann zwar jede beliebige Dateinamenserweiterung benutzt werden, allerdings muss bei Verwendung einer nicht bekannten Erweiterung das Ausgabeformat explizit an `lilypond-book` angegeben werden. Details dazu finden sich im Abschnitt Abschnitt 4.4 [lilypond-book aufrufen], Seite 25. Wird eine bekannte Erweiterung benutzt, wählt `lilypond-book` automatisch das richtige Ausgabeformat basierend auf der Erweiterung der Eingabedatei:

Erweiterung	**Ausgabeformat**
‘`.html`’	HTML
‘`.itely`’	Texinfo
‘`.latex`’	LaTeX
‘`.lytex`’	LaTeX
‘`.lyxml`’	DocBook
‘`.tely`’	Texinfo
‘`.tex`’	LaTeX
‘`.texi`’	Texinfo

'.texinfo'	Texinfo
'.xml'	HTML

Wird dieselbe Erweiterung für die Eingabedatei wie für die Ausgabedatei benutzt und befindet sich die Eingabedatei im aktuellen Arbeitsverzeichnis von lilypond-book, muss die --output Option für lilypond-book benutzt werden. Anderenfalls würde lilypond-book ja die Eingabedatei überschreiben, weshalb das Programm in diesem Fall mit einer Fehlermeldung wie „Fehler: Ausgabe würde Eingabedatei überschreiben; verwenden Sie –output.“ abbricht.

4.6 Alternative Methoden, Text mit Musik zu kombinieren

Dieser Abschnitt stellt Methoden vor, wie Text und Musik auf andere Weise kombiniert werden können als dies durch lilypond-book automatisiert geschieht.

4.6.1 Viele Zitate aus einer langen Partitur

Wenn aus einer großen Partitur viele kleine Fragmente eingefügt werden sollen, kann dazu das ‚clip systems‘ Feature benutzt werden. Siehe Abschnitt “Notationsfragmente extrahieren” in *Notationsreferenz*.

4.6.2 LilyPond-Noten in OpenOffice.org integrieren

Musik im LilyPond-Format kann in OpenOffice.org eingefügt werden mittels OOoLilyPond.

4.6.3 LilyPond-Noten in andere Programme integrieren

Im die Ausgabe von LilyPond in anderen Programmen einzufügen, sollte lilypond benutzt werden. Jedes Beispiel muss getrennt manuell erzeugt und ins Dokument eingefügt werden; für letzteres schlagen Sie bitte im Handbuch Ihrer Textverarbeitungs-Software nach. Die meisten Programme unterstützen das Einfügen von Grafiken im 'PNG'-, 'EPS'- oder 'PDF'-Format.

Um den leeren Rand um die Notenzeilen zu verringern, können folgende Einstellungen benutzt werden:

```
\paper{
  indent=0\mm
  line-width=120\mm
  oddFooterMarkup=##f
  oddHeaderMarkup=##f
  bookTitleMarkup = ##f
  scoreTitleMarkup = ##f
}

{ c1 }
```

Eine 'EPS'-Datei kann mit folgendem Befehl erzeugt werden:

```
lilypond -dbackend=eps -dno-gs-load-fonts -dinclude-eps-fonts   Dateiname.ly
```

eine 'PNG'-Datei mittels:

```
lilypond -dbackend=eps -dno-gs-load-fonts -dinclude-eps-fonts --png Dateiname.ly
```

5 Von anderen Formaten konvertieren

LilyPond kann auch Musik aus diversen anderen Formaten importieren. Dieses Kapitel beschreibt die dazu mit LilyPond mitgelieferten Hilfsprogramme. Daneben existieren natürlich auch noch weitere Programme, die Dateien für LilyPond erstellen können, wie etwa graphische Sequenzierprogramme und XML-Konverter. Näheres dazu findet sich auf der Homepage von LilyPond.

Die im Folgenden beschriebenen Programme sind eigenständige Hilfsprogramme und werden üblicherweise von der Kommandozeile aufgerufen. Siehe Abschnitt 3.2 [Benutzung auf der Kommandozeile], Seite 6 für weitere Informationen. Wenn Sie MacOS 10.3 oder 10.4 benutzen und Probleme mit diesen Skripten (z. B. `convert-ly`) haben, lesen Sie Abschnitt 2.1.1 [Einrichtung für MacOS X], Seite 2.

Bekannte Probleme und Warnungen

Leider haben wir nicht ausreichend viele Entwickler, um all die folgenden Hilfsprogramme ständig zu warten. Wir stellen den aktuellen Stand zur Verfügung, können aber leider Fehlerberichte nur selten bearbeiten. Selbstverständlich sind Patches von Ihnen sehr willkommen!

5.1 `midi2ly` aufrufen

`midi2ly` übersetzt eine Typ 1 MIDI-Datei in eine Eingabedatei für LilyPond.

MIDI (Music Instrument Digital Interface) ist ein internationaler Standard für digitale Instrumente: Es spezifiziert die Verkabelung, ein serielles Protokoll und ein Dateiformat. Das MIDI-Dateiformat ist der de-facto Standard um Musik von vielen Programmen zu exportieren. Allerdings fehlen in den MIDI-Dateien viele Ausdrucks- und Artikulationszeichen. Dennoch kann MIDI vielfach nützlich sein, um Musik von einem Programm zu importieren, für das kein spezielles Hilfsprogramm den direkten Import nach LilyPond unterstützt.

`midi2ly` konvertiert die MIDI-Spuren nach Abschnitt “Staff” in *Referenz der Interna* und MIDI-Kanäle in Abschnitt “Voice” in *Referenz der Interna* Kontexte. Tonhöhen werden relativ angegeben, Tondauern nur wenn nötig.

MIDI-Dateien können auch direkt von einem digitalen Keyboard aufgenommen und dann in eine ‘`.ly`’-Datei konvertiert werden. Allerdings sind Musikinterpretationen von Menschen (aus gutem Grund!) rhythmisch nicht exakt genug um die Konvertierung von MIDI nach LY trivial zu gestalten. Wenn `midi2ly` mit Quantisierung (`-s` und `-d` Kommandozeilenoptionen) aufgerufen wird, versucht es diese Unschärfen im Zeitablauf zu korrigieren, ist allerdings nicht sonderlich gut darin. Daher können wir diese Technik leider nicht für die Konvertierung von MIDI-Aufnahmen empfehlen.

`midi2ly` wird von der Kommandozeile folgendermaßen aufgerufen:

```
midi2ly [Optionen]... MIDI-Datei
```

Unter ‚Kommandozeile‘ verstehen wir dabei die Kommandozeile des jeweiligen Betriebssystems. Für nähere Informationen hierzu siehe Kapitel 5 [Von anderen Formaten konvertieren], Seite 29.

Die folgenden Kommandozeilenoptionen werden von `midi2ly` unterstützt:

`-a, --absolute-pitches`
: Gibt absolute Tonhöhen aus.

`-d, --duration-quant=`*LÄNGE*
: Quantisiert Tondauern zu Vielfachen von *LÄNGE*.

`-e, --explicit-durations`
: Gibt alle Tondauern explizit an.

`-h,--help`
: Zeigt eine Zusammenfassung der Programmbenutzung und der Optionen.

`-k, --key=`*acc*`[:`*Moll*`]`
: Setzt die Standard-Tonart. *acc* > 0 gibt die Anzahl der Kreuze an, *acc* < 0 gibt die Anzahl der Bs der Tonart an. Eine Moll-Tonart wird durch `:1` angegeben.

`-o, --output=`*Datei*
: Die Ausgabe wird in die Datei *Datei*‘`.ly`’ geschrieben.

`-s, --start-quant=`*LÄNGE*
: Quantisiert den Beginn der Noten zu Vielfachen von *LÄNGE*.

`-t, --allow-tuplet=`*DUR*`*`*NUM*`/`*DEN*
: Erlaubt Tuplet-Dauern der Form *DUR***NUM*/*DEN*.

`-V, --verbose`
: Gibt ausführliche informative Meldungen während der Konvertierung aus.

`-v, --version`
: Gibt die Versionsnummer aus.

`-w, --warranty`
: Zeigt die Lizenzbedingungen und Urheberrechtshinweise.

`-x, --text-lyrics`
: Interpretiert alle Texte als Liedtexte.

Bekannte Probleme und Warnungen

Überlappende Noten in einem Arpeggio werden nicht korrekt dargestellt. Nur die erste Note wird eingelesen und konvertiert, die restlichen werden ignoriert. Als Abhilfe können Sie alle Noten auf dieselbe Tonlänge setzen und Phrasierungszeichen oder Pedalindikatoren hinzufügen.

5.2 `musicxml2ly` aufrufen

MusicXML ist ein XML-Dialekt zur Darstellung von Musiknotation.

`musicxml2ly` wandelt eine MusicXML-Datei nach LilyPond um, wobei sowohl die Noten, Artikulationszeichen, Struktur der Partitur, Liedtexte etc. einer MusicXML-Datei (im ‚partwise‘-Format) in eine .ly-Datei um.

`musicxml2ly` wird von der Kommandozeile folgendermaßen aufgerufen:

```
musicxml2ly [Optionen]... XML-Datei
```

Unter ‚Kommandozeile‘ verstehen wir dabei die Kommandozeile des jeweiligen Betriebssystems. Für nähere Informationen hierzu siehe Kapitel 5 [Von anderen Formaten konvertieren], Seite 29.

Wenn als Dateiname ‘`-`’ angegeben wird, liest `musicxml2ly` Daten direkt von der Kommandozeile ein.

Die folgenden Kommandozeilenoptionen werden von `musicxml2ly` unterstützt:

`-a, --absolute`
: Konvertiert in absolute Tonhöhen.

`-h,--help`
: Zeigt eine Zusammenfassung der Programmbenutzung und der Optionen.

`-l, --language=LANG`
: Inkludiert die Sprachdatei *LANG*‘`.ly`’ in der Ausgabe. Wird z.B. ’deutsch’ angegeben, wird ‘`deutsch.ly`’ in die .ly-Datei eingebunden und deutsche Notenbezeichnungen benutzt.

`--lxml` Benutzt das lxml.etree Python-Paket für die Verarbeitung von XML (benötigt weniger Speicher und Rechenleistung)

`--nd --no-articulation-directions`
Konvertiert keine Richtungsangaben (`^`, `_` oder `-`) von Artikulations- und Lautstärkebezeichnungen.

`--no-beaming`
Konvertiert keine Informationen über die Balkensetzung aus der MusicXML-Datei. Stattdessen wird dies LilyPond überlassen.

`-o,--output=`*Dateiname*
Die Ausgabe wird in die Datei *Dateiname*‘`.ly`’ geschrieben. Wird als *Dateiname* nur ‘`-`’ angegeben, wird das Ergebnis der Konvertierung an der Kommandozeile ausgegeben. Wird diese Option nicht angegeben, so erfolgt die Ausgabe in die Datei *XML-Datei*‘`.ly`’.

`-r,--relative`
Konvertiert in relative Tonhöhen. (Standardeinstellung)

`-v, --verbose`
Gibt ausführliche informative Meldungen während der Konvertierung aus.

`--version`
Gibt die Versionsnummer aus.

`-z,--compressed`
Die Eingabedatei wird als komprimierte MusicXML-Datei eingelesen. Dies ist die Standardeinstellung für Dateien mit der Erweiterung ‘`.mxl`’.

5.3 `abc2ly` aufrufen

ABC ist ein relativ einfaches ASCII-basierendes Musikformat und ist dokumentiert auf der ABC-Homepage:

http://www.walshaw.plus.com/abc/learn.html.

`abc2ly` konvertiert ABC-Dateien nach LilyPond und wird von der Kommandozeile folgendermaßen aufgerufen:

`abc2ly [`*Optionen*`]...` *ABC-Datei*

Unter ‚Kommandozeile‘ verstehen wir dabei die Kommandozeile des jeweiligen Betriebssystems. Für nähere Informationen hierzu siehe Kapitel 5 [Von anderen Formaten konvertieren], Seite 29.

Die folgenden Kommandozeilenoptionen werden von `abc2ly` unterstützt:

`-b,--beams=None`
Die Balkensetzung aus der ABC-Datei erhalten.

`-h,--help`
Zeigt eine Zusammenfassung der Programmbenutzung und der Optionen.

`-o,--output=`*Dateiname*
Die Ausgabe wird in die Datei *Dateiname*‘`.ly`’ geschrieben.

`-s,--strict`
Strenge Auslegung einer erfolgreichen Konvertierung.

`-v,--version`
Gibt die Versionsnummer aus.

Es existiert außerdem die Möglichkeit, LilyPond-Befehle für die Konvertierung mit `abc2ly` gleich in der ABC-Datei anzugeben. Wenn sich etwa in der ABC-Datei das Kommentar

```
%%LY voices \set autoBeaming = ##f
```

befindet, so wird der Text nach dem Schlüsselwort ‚voices' direkt in die aktuelle Stimme in der LilyPond-Datei eingefügt.

Ebenso bewirkt

```
%%LY slyrics more words
```

dass alles nach dem ‚slyrics' Schlüsselwort an der aktuellen Stelle im Liedtext eingefügt wird.

Bekannte Probleme und Warnungen

Der ABC-Standard ist eigentlich kein wirklich vollständiger Standard. Für komplexere Notation wie etwa Polyphonie existieren verschiedene Konventionen.

Mehrere Lieder in einer Datei können nicht konvertiert werden.

ABC synchronisiert den Liedtext am Anfang jeder Zeile mit den Noten, `abc2ly` macht dies nicht.

`abc2ly` ignoriert die Balkensetzung in der ABC-Datei.

5.4 `etf2ly` aufrufen

ETF (Enigma Transport Format) ist ein Dateiformat, das Coda Music Technology in älteren Versionen des Programms Finale benutzt hat.

`etf2ly` konvertiert Teile einer ETF-Datei nach LilyPond und wird von der Kommandozeile folgendermaßen aufgerufen:

```
etf2ly [Optionen]... ETF-Datei
```

Unter ‚Kommandozeile' verstehen wir dabei die Kommandozeile des jeweiligen Betriebssystems. Für nähere Informationen hierzu siehe Kapitel 5 [Von anderen Formaten konvertieren], Seite 29.

Die folgenden Kommandozeilenoptionen werden von `etf2ly` unterstützt:

`-h,--help`
: Zeigt eine Zusammenfassung der Programmbenutzung und der Optionen.

`-o,--output=`*Dateiname*
: Die Ausgabe wird in die Datei *Dateiname*`.ly` geschrieben.

`--version`
: Gibt die Versionsnummer aus.

Bekannte Probleme und Warnungen

Die Liste der Artikulationszeichen is unvollständig. Leere Takte verwirren `etf2ly`. Mehrfache Vorschlagnoten werden falsch beendet.

5.5 LilyPond-Dateien erstellen

LilyPond selbst wird nur mit den oben genannten Konvertierprogrammen ausgeliefert und unterstützt keine weiteren Formate. Es existieren allerdings externe Hilfsprogramme, die auch LilyPond-Dateien erzeugen können. Diese umfassen unter anderem:

- Denemo, ein grafisches Programm zum Musiksatz.
- Rumor, ein MIDI-nach-LilyPond Konverter, der monophone MIDI-Eigaben in Echtzeit konvertiert.
- lyqi, ein Emacs-Modus.

- xml2ly, ein Konverter für MusicXML, der auf XML-Transformationen mittels XSLT basiert.
- NoteEdit, ein grafisches Programm zum Notensatz, das auch MusicXML importieren kann.
- Rosegarden, das MIDI importieren kann.
- FOMUS, eine LISP-Bibliothek, um Musiknotation zu erzeugen.
- `http://vsr.informatik.tu-chemnitz.de/staff/jan/nted/nted.xhtml`, ein Notensatzprogramm mit experimentellem Export nach LilyPond.
- TuxGuitar, ein Programm zur Erzeugung von Gitarren-Tabulaturen, unterstützt den Export nach LilyPond.
- MuseScore unterstützt ebenfalls den Export nach LilyPond.

Anhang A GNU Free Documentation License

Version 1.1, March 2000

Copyright © 2000 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of ‚copyleft‘, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The ‚Document‘, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as ‚you‘.

A ‚Modified Version‘ of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A ‚Secondary Section‘ is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The ‚Invariant Sections‘ are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The ‚Cover Texts‘ are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A ‚Transparent‘ copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file

format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not „Transparent“ is called „Opaque“.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The „Title Page“ means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, „Title Page“ means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

   You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

   You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

   If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

   If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

   If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

   It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section entitled ‚History‘, and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled ‚History‘ in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the ‚History‘ section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. In any section entitled ‚Acknowledgments‘ or ‚Dedications‘, preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgments and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled ‚Endorsements‘. Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as ‚Endorsements‘ or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to

the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled ‚Endorsements', provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled ‚History' in the various original documents, forming one section entitled ‚History'; likewise combine any sections entitled ‚Acknowledgments', and any sections entitled ‚Dedications'. You must delete all sections entitled ‚Endorsements.'

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an ‚aggregate', and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

    The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

    Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License ‚or any later version‘ applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

Anhang: Wie kann die Lizenz für eigene Dokumente verwendet werden

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
  Copyright (C)  year  your name.
  Permission is granted to copy, distribute and/or modify this document
  under the terms of the GNU Free Documentation License, Version 1.1
  or any later version published by the Free Software Foundation;
  with the Invariant Sections being list their titles, with the
  Front-Cover Texts being list, and with the Back-Cover Texts being list.
  A copy of the license is included in the section entitled ,GNU
Free Documentation License‘
.
```

If you have no Invariant Sections, write ,with no Invariant Sections‘ instead of saying which ones are invariant. If you have no Front-Cover Texts, write ,no Front-Cover Texts‘ instead of ,Front-Cover Texts being *list*‘; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Anhang B LilyPond-Index

\

\header in LaTeX documents 18

A

ABC 31
Aktualisierung mit convert-ly 11
Aktualisierung von LilyPond-Dateien 11
Aufruf von dvips 25
Ausgabeformat 8

C

Coda Technology 32
convert-ly: Aktualisierung 11

D

Dateigröße, Ausgabedatei 4
Dateisuche 9
docbook 15
DocBook, Musik in 15
Dokument, Musik einfügen in 15
dvips 25

E

Editoren 3
emacs 3
enigma 32
error messages 10
ETF 32
Externe Programme, LilyPond-Dateien erzeugen .. 32

F

Fataler Fehler 11
FDL, GNU Free Documentation License 34
Fehler 11, 14
Fehler melden 14
Fehlermeldung, Format 11
Fehlerprotokoll, Scheme 11
Finale 32
`First steps` 6
Form der Fehlermeldungen 11

H

Hervorhebung der Syntax 3
html 15
HTML, Musik in 15

K

Kommandozeilen-Optionen für `lilypond` 6
Konturschriften 25

L

LANG 10
latex 15
LaTeX, Musik in 15
`lilypond` aufrufen 6
LILYPOND_DATADIR 10

M

MIDI 29
Modus, Editoren 3
MusicXML 30
Musikwissenschaft 15

N

`Notationsfragmente extrahieren` 28

O

Optionen an der Kommandozeile 6

P

Point and click 4
Point and Click, Kommandozeile 7
PostScript Ausgabeformat 8
Programmierfehler 11

S

Scheme dump 8
Scheme Fehler 11
`Staff` 29
Suchpfad 9
SVG (Scalable Vector Graphics) 8
Syntax-Hervorhebung 3

T

texi 15
texinfo 15
Texinfo, Musik in 15
titling and lilypond-book 18
Type1 Schriften 25

V

vim 3
`Voice` 29

W

Warnung 10