LilyPond

Usage

Das LilyPond-Entwicklerteam


Zu mehr Information, wie dieses Handbuch unter den anderen Handbüchern positioniert, oder um dieses Handbuch in einem anderen Format zu lesen, besuchen Sie bitte Abschnitt “Manuals” in Allgemeine Information.

Wenn Ihnen Handbücher fehlen, finden Sie die gesamte Dokumentation unter http://lilypond.org/.

Copyright © 1999–2015 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.


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 2.19.84
# Inhaltsverzeichnis

## 1 lilypond starten .................................................. 1

1.1 Übliche Programmbenutzung ............................................ 1
1.2 Benutzung auf der Kommandozeile ...................................... 1
   lilypond aufrufen .......................................................... 1
   Häufige Kommandozeilenbefehle ........................................ 2
   Grundlegende Optionen auf der Kommandozeile für LilyPond ........ 2
   Fortgeschrittene Optionen auf der Kommandozeile für LilyPond .... 5
   Umgebungsvariablen ...................................................... 10
   LilyPond in chroot-Kerker ............................................... 10
1.3 Fehlermeldungen ........................................................... 12
1.4 Häufige Fehlermeldungen ................................................ 13
   Noten laufen aus der Seite heraus ..................................... 13
   Ein zusätzliches System erscheint ..................................... 13
   Offensichtlicher Fehler in ../ly/init.ly .............................. 14
   Fehlermeldung Ungebundene Variable % ................................ 15
   Fehlermeldung FT_Get_Glyph_Name .................................... 15
   Warnung über absteigende staff affinities ............................ 15

## 2 Dateien mit convert-ly aktualisieren ............................... 16

2.1 Warum verändert sich die Syntax? .................................... 16
2.2 convert-ly aufrufen .................................................... 16
2.3 Optionen auf der Kommandozeile für convert-ly .................... 17
2.4 Probleme mit convert-ly ............................................... 17
2.5 Manuelle Konversion .................................................. 18

## 3 lilypond-book aufrufen ................................................ 20

3.1 Ein musikwissenschaftlicher Text als Beispiel ...................... 20
3.2 Noten in Text integrieren .............................................. 24
   3.2.1 \LaTeX .......................................................... 24
   3.2.2 Texinfo ......................................................... 26
   3.2.3 HTML .......................................................... 27
   3.2.4 DocBook ....................................................... 28
3.3 Die Musikfragment-Optionen ............................................ 29
3.4 lilypond-book aufrufen ............................................... 32
3.5 Dateiendungen .......................................................... 35
3.6 lilypond-book-Vorlagen ............................................... 35
   3.6.1 \LaTeX .......................................................... 35
   3.6.2 Texinfo ......................................................... 36
   3.6.3 html ........................................................... 36
   3.6.4 xelatex ........................................................ 37
3.7 Das Inhaltsverzeichnis flexibel einsetzen ............................ 38
3.8 Alternative Methoden Text und Musik zu kombinieren ............... 39

## 4 Externe Programme ..................................................... 40

4.1 Point and click ....................................................... 40
   Point and click aktivieren ............................................. 40
   Selektives point-and-click ............................................ 41
4.2 Unterstützung von Texteditoren .................................................. 41
   Emacs-Modus ........................................................................ 41
   Vim-Modus ........................................................................ 42
   Andere Editoren .................................................................. 42
4.3 Von anderen Formaten konvertieren ........................................ 42
   4.3.1 midi2ly aufrufen .......................................................... 42
   4.3.2 musicxml2ly aufrufen .................................................... 44
   4.3.3 abc2ly aufrufen .......................................................... 45
   4.3.4 etf2ly aufrufen .......................................................... 46
   4.3.5 Andere Formate .......................................................... 46
4.4 LilyPond-Ausgabe in anderen Programmen ................................ 46
   Viele Zitate aus einer langen Partitur ...................................... 46
   LilyPond-Noten in OpenOffice.org integrieren ......................... 46
   LilyPond-Noten in andere Programme integrieren ..................... 46
4.5 Unabhängige include-Abschnitte ............................................. 47
   4.5.1 MIDI-Artikulation ....................................................... 47

5 Vorschläge zum Schreiben von
   LilyPond-Eingabe-Dateien ..................................................... 48
   5.1 Allgemeine Vorschläge ....................................................... 48
   5.2 Das Kopieren von bereits vorhandener Musik ....................... 49
   5.3 Große Projekte ............................................................... 49
   5.4 Fehlersuche .................................................................... 50
   5.5 Make und Makefiles ....................................................... 50

Anhang A  GNU Free Documentation License ............................... 57

Anhang B  LilyPond-Index ......................................................... 64
1 lilypond starten

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

1.1 Übliche Programmbenutzung

Die meisten Benutzer führen LilyPond von einer graphischen Benutzeroberfläche aus. Siehe Abschnitt “Tutorium” in Handbuch zum Lernen, falls Sie dies nicht bereits getan haben. Wenn Sie einen alternativen Editor für Ihre LilyPond-Dateien verwenden, lesen Sie bitte die Dokumentation dieses Editors.

1.2 Benutzung auf der Kommandozeile


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.

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


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

erzeugt daher die Dateien Dateiname-Geige.pdf und Dateiname-Cello-1.pdf.

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.
Häufige Kommandozeilenbefehle
Wenn Ihre Kommandozeile normale Weiterleitungen unterstützt, können Sie es nützlich finden, mit folgenden Befehlen die Ausgabe der Kommandozeile in eine Datei zu leiten:

- `lilypond file.ly 1>stdout.log` um normale Ausgabe zu erhalten
- `lilypond file.ly 2>stderr.log` um Fehlermeldungen zu erhalten
- `lilypond file.ly &>all.log` um alle Meldungen zu erhalten

Wenden Sie sich an die Dokumentation für Ihre Kommandozeile, um zu sehen, ob derartige Optionen unterstützt werden oder die Syntax unterschiedlich ist. Beachten Sie, dass es sich hier um reine Verwaltungsprogramme handelt, die nichts mit LilyPond zu tun haben.

Grundlegende Optionen auf der Kommandozeile für LilyPond
Die folgenden Kommandozeilenoptionen werden von `lilypond` unterstützt:

- `-d, --define-default=Variable=Wert`
  Siehe [Fortgeschrittene Optionen auf der Kommandozeile für LilyPond], Seite 5.

- `-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.
  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:
  ```shell`
  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:
  ```scheme`
  #(use-modules (guile-user))
  ```
  **Achtung:** Windows-Benutzer müssen doppelte anstelle der einfachen Anführungsstriche einsetzen.

- `-E, --eps`
  Erzeugt EPS-Dateien.

- `-f, --format=Format`
  Beispiel: `lilypond -fpng Dateiname.ly`

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

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

- `-i, --init=Initialisierungsdatei`
  Benutzt `Initialisierungsdatei` zur gesamten Programm initialisierung. Der Standardwert ist `init.ly`.

- `-I, --include=Verzeichnis`
  Fügt `Verzeichnis` zur Liste der Suchpfade hinzu.
Kapitel 1: lilypond starten

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

Die --jail Option ist eine flexiblere Alternative zu -dsafe, wenn LilyPond über das Internet verfügbar gemacht wird oder LilyPond Befehle ausführt, die aus externen Quellen stammen (siehe [Fortgeschrittene Optionen auf der Kommandozeile für LilyPond], Seite 5).

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

Erstellung eines eigenen Benutzerkontos

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. Siehe auch [LilyPond in chroot-Kerker], Seite 10,

-1, --loglevel=Logstufe
Passt die Ausführlichkeit der Ausgabe auf der Kommandozeile entsprechend Logstufe an. Mögliche Werte sind:

NONE Keine Ausgabe, nicht einmal Fehlermeldungen.
ERROR Nur Fehlermeldungen, keine Warnungen oder Fortschrittssmeldungen.
WARN Warnungen und Fehlermeldungen, keine Fortschrittssmeldungen.
BASIC_PROGRESS Grundlegende Fortschrittssmeldungen (Erfolg), Warnungen und Fehler.
PROGRESS Alle Fortschrittssmeldungen, Warnungen und Fehler.
INFO (Standard) Fortschrittssmeldungen, Warnungen, Fehlermeldungen und weiter Information über die Ausführung.
DEBUG Alle möglichen Meldungen, die auch mit der Fehlersuche (Debug) zu tun haben können.

-o, --output=DATEI oder ORDNER

-0, --pspdfopt
Wird diese Option nicht oder mit dem Parameter size verwendet, wird eine Art der Erzeugung von Postscript- EPS- bzw. PDF-Dateien gewählt, die möglichst kleine Dateien erzeugt.
Der Parameter TeX führt zu Dateien, die besonders gut zur Einbindung in PdfTeX, LuaTeX oder XeTeX geeignet sind.
Sollen mehrere mit LilyPond erzeugte Dokumente in ein TeX-Dokument eingebunden werden, so bietet es sich an, TeX-GS zu verwenden und das von TeX erzeugte PDF noch einmal mit Ghostscript zu bearbeiten.

--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.

-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!)
Fortgeschrittene Optionen auf der Kommandozeile für LilyPond

\texttt{-d[Optionsbezeichnung]=[Wert],--define-default=[Optionsbezeichnung]=[Wert]}
Hiermit wird die entsprechende interne Scheme-Funktion auf den Wert gesetzt. Wenn kein \texttt{value} angegeben wird, wird der Standardwert eingesetzt. Die Vorsilbe \texttt{no-} kann zur \texttt{Optionsbezeichnung} hinzugefügt werden, um eine Funktion „auszuschalten“. Beispielsweise

\texttt{-dpoint-and-click=#f}

ist das gleiche wie

\texttt{-dno-point-and-click}

Folgende Optionen sind mit ihren entsprechenden Standardwerten unterstützt:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Wert</th>
<th>Erklärung/Optionen</th>
</tr>
</thead>
<tbody>
<tr>
<td>anti-alias-factor</td>
<td>1</td>
<td>Die Bilder in einer höheren Auflösung rendern (Faktor angegeben) und das Resultat herunterrechnen, um „Zacken“ in PNG-Bildern zu vermeiden.</td>
</tr>
<tr>
<td>aux-files</td>
<td>#t</td>
<td>Erstelle .tex, .texi, .count-Dateien im EPS-Backend.</td>
</tr>
<tr>
<td>backend</td>
<td>ps</td>
<td>Auswahl des Backend. Postscript-Dateien (Standart) enthalten TTF, Type1 und OTF-Schriftarten. Ihr Zeichenvorrat wird nicht reduziert (Subsetting). Die Benutzung von östlichen Schriftarten kann zu sehr großen Dateien führen.</td>
</tr>
<tr>
<td>null</td>
<td></td>
<td>Keine graphische Partitur ausgeben; hat den gleichen Effekt wie \texttt{-dno-print-pages}.</td>
</tr>
</tbody>
</table>
Kapitel 1: LilyPond starten

svg

scm
Ausgabe der rohen internen Scheme-basierten Zeichnungsbefehle.

check-internal-types #f
Überprüfe jede Eigenschaftszuweisung für Typen.

clip-systems #f
Erstelle ausgeschnittene Schnipsel einer Partitur.

datadir
Präfix für Datendateien (read-only).

debug-gc #f
Gebe Debugging-Statistik für Speicher aus.

debug-gc-assert-parsed-dead #f
Für Speicher-Debugging: Gehe sicher, dass alle Referenzen zu geparsten Objekten tot sind. Das ist eine interne Option und sie wird automatisch für `--debug-gc' angestellt.

debug-lexer #f
Debugging des Flex-lexer.

debug-page-breaking-scoring #f
Gebe viele unterschiedliche Seitenumbruchs situationen für Partituren aus.

debug-parser #f
Debugging des Bison-Parsers.

debug-property-callbacks #f
Debugging von zyklischen Callback-Ketten.

debug-skylines #f
Debugging von Skylines.

delete-intermediate-files #t
Entferne unbemutzbare, zwischenzeitliche .ps-Dateien, die während der Kompilations erstellt werden.

dump-cpu-profile #f
Gebe CPU-Zeitinformation aus (abhängig vom System).
Kapitel 1: lilypond starten

dump-profile #f
Gebe Speicher- und CPU-Zeitbenutzung für jede Datei aus.

dump-signatures #f
Gebe Ausgabesignaturen für jedes System aus. Wird für das Prüfen der Regressionsteste eingesetzt.

esp-box-padding #f
Verschiebe die linke Ecke der ausgegebenen EPS-Boundingbox um die angegebene Entfernung (in mm).

gs-load-fonts #f
Lade die Schriftarten durch Ghostscript.

gs-load-lily-fonts #f
Lade nur die LilyPond-Schriftarten durch Ghostscript.

gui #f
Gibt keine Ausgabe auf der Kommandozeile aus, sondern schreibt alles in die Log-Datei.

Anmerkung für Windows-Benutzer: Standardmäßig gibt lilypond.exe alle Fortschrittsinformation auf der Kommandozeile aus; lilypond-windows.exe gibt aber keine Fortschrittsinformation aus und zeigt sofort den Prompt an. Die Option -dgui kann hier benutzt werden, um die Ausgabe in eine Log-Datei umzuleiten.

help #f
Zeige die Hilfe.

include-book-title-preview #t
Füge Titel eines Buches (book) in die Vorschaubilder ein.

include-eps-fonts #t
Füge Schriftarten in EPS-Dateien von einzelnen Systemen ein.

include-settings #f
Füge eine Datei für globale Einstellungen ein, dieses wird gelesen, bevor die Partitur verarbeitet wird.

job-count #f
Bearbeite Dateien parallel, mit der angegebenen Anzahl von Prozessen.

log-file #f [file]
Wenn die Zeichenkette FOO als ein zweites Argument angegeben wird, wird die Ausgabe in die Log-Datei FOO.log umgeleitet.

max-markup-depth 1024
Maximale Tiefe eines Beschriftungs-(markup)-Baumes. Wenn eine Beschriftung mehr Ebenen hat, wird angenommen, dass die Beschriftung nicht von sich aus schließt, eine Warnung ausgegeben und eine leere Beschriftung gesetzt.

midi-extension "midi"
Schreibe als Standarddateierweiterung für MIDI die angegebene Zeichenkette.
Kapitel 1: lilypond starten

music-strings-to-paths #f
Konvertiere Textzeichenketten in Pfade, wenn
die Glyphen einer Musik-Schriftart gehören.

old-relative #f
Lässt den \relative-Modus für simultane
Musik ähnlich wie die Akkord-Syntax funktionieren.

paper-size "a4"
Stelle die Standardpapiergröße ein. Beachte-
en Sie, dass die Zeichenkette von doppelten
Anführungszeichen mit Backslash umgeben
werden muss.

pixmap-format png16m
Stellt das Ausgabeformat von GhostScript für
Pixel-Bilder ein.

point-and-click #f
Füge ,point & click'-Links in die PDF-
Ausgabedatei ein. Siehe auch Abschnitt 4.1
[Point and click], Seite 40.

preview #f
Erstelle Vorschaubilder zusätzlich zur nor-
malen Ausgabe.

Diese Option wird von allen Backends unterstützt: pdf, png, ps, eps und svg, allerdings nicht
cbm. Hiermit wird eine Ausgabedatei in der Form meineDatei.preview.Dateierweiterung
erstellt, die die Titel und das erste Notensystem enthält. Wenn \book- oder \bookpart-
Umgebungen ingesetzt werden, werden die Titel von \book, \bookpart oder \score in die
Ausgabe aufgenommen, sowie das erste System jeder \score-Umgebung, wenn die Variable
print-all-headers in der paper-Umgebung auf #t eingestellt ist.

Um die normale Ausgabe zu unterdrücken, können die Optionen -dprint-pages oder
-dno-print-pages eingesetzt werden.

print-pages #t
Erstelle vollständige Seiten, der Standard.
-dno-print-pages ist hilfreich im Zusammen-
hang mit -dpreview.

profile-property-accesses #f
Speichere Statistiken von get_property()-
Funktionsaufrufen.

protected-scheme-parsing #t
Fahre fort, wenn Fehler in eingefügtem
Scheme-Code im Parser bemerkt werden.
Wenn auf #f gesetzt, halte an Fehlern an und
glebe einen Stacktrace aus.

read-file-list #f [file]
Gibt den Dateinamen einer Datei an, die eine
Liste mit Eingabedateien enthält, die kom-
piliert werden sollen.

relative-includes #f
Wenn ein \include-Befehl bearbeitet wird,
suche nach der eingefügten Datei relativ zur
aktuellen Datei (und nicht relativ zur unter-
sten Ebene).
Kapitel 1: LilyPond starten

resolution 101 Setzt die Auflösung, mit der PNG-Bilder erstellt werden, auf einen bestimmten Wert (in dpi).

safe #f Der .ly-Eingabe nicht trauen.

Wenn LilyPond-Notensatz über einen Webserver zur Verfügung gestellt wird, müssen entweder die Option --safe oder die Option --jail mitgegeben werden. Die Option --safe verhindert eingefügten Scheme-Code daran, Schaden auszuüben, etwa

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

Die Option --dsafe interpretiert eingefügte Scheme-Ausdrücke in einem besonderen sicheren Modul. Das ist aus dem GUILE safe-r5rs-Modul abgeleitet, fügt aber einige Funktionen der LilyPond API hinzu, welche sich in scm/safe-lily.scm aufgelistet finden.

Zusätzliche verbietet der sichere Modus \include-Befehle und stellt die Benutzung von Backslash in \TeX Zeichenketten aus. Im sicheren Modus ist es auch nicht möglich, LilyPond-Variablen in Scheme zu importieren.


Der sichere Modus verhindert die Kompilierung von vielen nützlichen LilyPond-Schnipseln.

die Option --jail ist noch sicherer, erfordert aber mehr Arbeit beim Setup. Siehe auch [Grundlegende Optionen auf der Kommandozeile für LilyPond], Seite 2.

separate-log-files #f Für Eingabedateien Datei1.ly, Datei2.ly usw. die Log-Daten in die Dateien Datei1.log, Datei2.log ... schreiben.

show-available-fonts #f Eine Liste der verfügbaren Schriftarten.

strict-infinity-checking #f Erzwinge einen Programmabsturz wenn Inf und NaN Fließkommaausnahmen gefunden werden.

strip-output-dir #t Verzeichnisse von Eingabedateien nicht in die Konstruktion der Ausgabedateinamen einbeziehen.

svg-woff #f Woff-Schriftarten im SVG-Backend benützen.

trace-memory-frequency #f Zeichnet die Benutzung von Scheme so oft pro Sekunde auf. Das Resultat wird in die Dateien FILE.stacks und FILE.graph ausgegeben.

trace-scheme-coverage #f Abdeckung der Scheme-Dateien in Datei FILE.cov schreiben.
Kapitel 1: lilypond starten

verbose #f
Ausführliche Ausgabe, also Logstufe DEBUG (read-only).

warning-as-error #f
Alle Warnungen und 'Programmierfehler'-Nachrichten in Fehler ändern.

Umgebungsvariablen

lilypond erkennt und benutzt 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_LOGLEVEL**
Die standardmäßige Logstufe. Wenn LilyPond ohne eine explizite Logstufe außerufen wird (d. h. die Kommandozeilenoption --loglevel nicht eingesetzt wird), wird dieser Wert benutzt.

**LILYPOND_GC_YIELD**
Eine Variable (von 1 bis 100), die die Speicherverwaltung regelt. Bei niedrigeren Werten wird mehr Prozessor-Zeit, dafür weniger Hauptspeicher benötigt. Voreinstellung ist ein Wert von 70.

LilyPond in chroot-Kerker


- Installieren Sie die nötigen Pakete: LilyPond, GhostScript und ImageMagick.
- Erstellen Sie einen neuen Benutzer mit dem Namen lily:
  ```
  adduser lily
  ```
  Hierdurch wird auch eine Gruppe lily und ein Heimat-Ordner /home/lily für den neuen Benutzer erstellt.
- Im Heimat-Ordner des Benutzers lily erstellen Sie eine Datei, die als eigenes Dateisystem eingesetzt wird:
  ```
  dd if=/dev/zero of=/home/lily/loopfile bs=1k count= 200000
  ```
  In diesem Beispiel wird eine 200-MB-Datei als Kerker-Dateisystem erstellt.
- Erstellen Sie ein loop device, erstellen Sie ein Dateisystem und mounten Sie es, dann erstellen Sie dort einen Ordner, in dem der Benutzer lily Schreibrechte hat:
  ```
  mkdir /mnt/lilyloop
  losetup /dev/loop0 /home/lily/loopfile
  mkfs -t ext3 /dev/loop0 200000
  mount -t ext3 /dev/loop0 /mnt/lilyloop
  mkdir /mnt/lilyloop/lilyhome
  chown lily /mnt/lilyloop/lilyhome
  ```
- In der Konfiguration des Servers ist der Kerker (JAIL) /mnt/lilyloop und das Verzeichnis (DIR) /lilyhome.
- Erstellen Sie einen großen Verzeichnisbaum in dem Kerker, indem Sie die notwendigen Dateien dorthin kopiert, wie das Beispielskript unten zeigt.
Sie könne `sed` benutzen, um die notwendigen Kopierbefehle für ein bestimmtes Programm zu erstellen:

```bash
for i in "/usr/local/lilypond/usr/bin/lilypond" "/bin/sh" "/usr/bin/; 
   do ldd $i | sed 's/.*/\1\&\& cp -L \1\2/ | sed 's/\t/\1\2/ \1\2/' | sed '/.*=>.*/d'; done
```

Beispiel-Skript für 32-bit Ubuntu 8.04

```bash
#!/bin/sh
## defaults set here

username=lily
home=/home
loopdevice=/dev/loop0
jaildir=/mnt/lilyloop
# the prefix (without the leading slash!)
lilyprefix=usr/local
# the directory where lilypond is installed on the system
lilydir=/$lilyprefix/lilypond/

userhome=$home/$username
loopfile=$userhome/loopfile
adduser $username
dd if=/dev/zero of=$loopfile bs=1k count=200000
mkdir $jaildir
losetup $loopdevice $loopfile
mkfs -t ext3 $loopdevice 200000
mount -t ext3 $loopdevice $jaildir
chown $username $jaildir/lilyhome
chown $username $jaildir/lilyhome
chdir $jaildir

mkdir -p bin usr/bin usr/share usr/lib usr/share/fonts $lilyprefix tmp
chmod a+w tmp

cp -r -L $lilydir $lilyprefix

cp -L /bin/sh /bin/rm bin

cp -L /usr/bin/convert /usr/bin/gs usr/bin

cp -L /usr/share/fonts/truetype $lilyprefix

# Now the library copying magic
for i in "$lilyprefix/usr/bin/lilypond" "$lilyprefix/usr/bin/guile" "/bin/sh" "/bin/rm" "/usr/bin/gs" "/usr/bin/convert"; do ldd $i | sed 's/.*/\1\&\& cp -L \1\2/ | sed 's/\t/\1\2/ \1\2/' | sed '/.*=>.*/d'; done | sh -s

# The shared files for ghostscript...
cp -L -r /usr/share/ghostscript usr/share

# The shared files for ImageMagick

cp -L -r /usr/lib/ImageMagick* usr/lib
```
### Now, assuming that you have test.ly in /mnt/lilyloop/lilyhome, you should be able to run:

Note that \$/lilyprefix/bin/lilypond is a script, which sets the \$/lilyprefix/bin/lilypond -jlily,lily,/mnt/lilyloop,/lilyhome test.ly

### Kapitel 1: lilypond starten

#### 1.3 Fehlermeldungen

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

**Warnung (Warning)**

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 (Error)**

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 (Fatal error)**

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 (Scheme error)**

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 sogenannte Call trace ausgegeben, der die aufgerufenen Funktionen zur Zeit des Fehlers angibt.

**Programmierfehler (Programming error)**

Eine interne Inkonsistenz ist aufgetreten. Diese Fehlermeldungen sollen den Programmieren 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

```
  { 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.

Mehr Information darüber findet sich in Abschnitt 1.4 [Häufige Fehlermeldungen], Seite 13.

1.4 Häufige Fehlermeldungen

Die Fehlermeldungen, die unten beschrieben werden, treten oft auf, es ist jedoch nicht immer einfach, die Ursache zu finden. Wenn der Fehler einmal verstanden ist, kann er einfach behoben werden.

Noten laufen aus der Seite heraus

Noten, die rechts aus der Seite herauslaufen oder sehr komprimiert aussehen, liegen in fast allen Fällen an einer falschen Notendauer einer Note, die dazu führt, dass die letzte Note im Takt über die Taktgrenze hinwegdauert. Es ist nicht falsch, wenn die letzte Note eines Taktes über den Takt hinausdauert, weil einfach angenommen wird, dass sie im nächsten Takt fortgesetzt wird. Aber wenn eine längere Sequenz dieser überhängenden Noten auftritt, können die Noten sehr gedrängt aussehen oder über den Seitenrand fließen, weil die automatische Zeilenumbruchsfunktion einen Umbruch nur am Ende eines vollständigen Taktes einfügen kann, also wenn alle Noten zum Ende des Taktstriches auch wirklich aufhören.

Achtung: Eine falsche Dauer kann dazu führen, dass Zeilenumbrüche nicht möglich sein und die Zeile entweder sehr gedrängt dargestellt wird oder über den Seitenrand fließt.

Die falsche Dauer kann einfach gefunden werden, wenn Taktstrichüberprüfung eingesetzt wird, siehe Abschnitt “Takt- und Taktzahlüberprüfung” in Notationsreferenz.

If you actually intend to have a series of such carry-over measures you will need to insert an invisible bar line where you want the line to break. For details, see Abschnitt “Taktstriche” in Notationsreferenz.

Ein zusätzliches System erscheint


\override Staff.NoteHead.color = #red
\new Staff { a' }

\begin{verbatim}
\override Staff.NoteHead.color = #red
\new Staff { a' }
\end{verbatim}
Das liegt daran, dass kein \Staff-Kontext existiert, wenn der \override-Befehl verarbeitet wird, sodass ein System für diesen Befehl erstellt wird. Dann aber erstellt \new Staff noch ein zusätzliches System, wo die Noten gesetzt werden. Die richtige Schreibweise wäre:

\new Staff {
  \override Staff.NoteHead.color = #red
  a'
}

Ein zweites Beispiel zeigt, dass ein \relative-Befehl innerhalb von \repeat zwei Systeme erstellt, wobei der zweite etwas verschoben ist. Das liegt daran, dass \repeat zwei \relative-Umgebungen erstellt, die jede implizit einen Staff- und Voice-Kontext erstellen.

\repeat unfold 2 {
  \relative { c'4 d e f }
}

Indem man die Voice-Kontexte explizit erstellt, kann das Problem umgangen werden.

\new Voice {
  \repeat unfold 2 {
    \relative { c'4 d e f }
  }
}

Offensichtlicher Fehler in ../ly/init.ly

Verschiedene seltsame Fehlermeldungen können über Syntax-Fehler in ../ly/init.ly auf- tauchen, wenn die Eingabedatei nicht richtig formuliert ist, wenn sie etwa nicht richtig passende Klammerpaare oder Anführungszeichen enthält.


Diese Fehlermeldung kann auch mit einem fehlenden schließenden Anführungszeichen (") auftreten. In diesem Fall sollte die begleitende Fehlermeldung eine Zeilenzahl angeben, die dicht am Fehler liegt. Die nicht paarigen Anführungszeichen sind meistens ein oder zwei Zeilen darüber.
Kapitel 1: lilypond starten

Fehlermeldung Ungebundene Variable %
Diese Fehlermeldung erscheint am Ende der Kommandozeilenausgabe oder in der Log-Datei mit einer Meldung „GUILE signalled an error ...“ jedes Mal, wenn eine Scheme-Routine aufgerufen wird, die (falscherweise) ein LilyPond-Kommentar und kein Scheme-Kommentar enthält.

LilyPond-Kommentare beginnen mit dem Prozent-Zeichen (%) und dürfen nicht in Scheme-Routinen benutzt werden. Scheme-Kommentare beginnen mit einem Semikolon (;).

Fehlermeldung FT.GetGlyph.Name
Diese Fehlermeldung erscheint in der Kommandozeilenausgabe, wenn die Datei ein Zeichen enthält, das nicht zu ASCII gehört und die Datei nicht in UTF-8-Kodierung gespeichert wurde. Siehe auch Abschnitt “Zeichenkodierung” in Notationsreferenz.

Warnung über absteigende staff affinities
Diese Warnung erscheint, wenn keine Notensysteme in der Ausgabe vorhanden sind, wenn etwa nur ChordName-Kontext und Lyrics-Kontext in einem Liedblatt vorhanden sind. Die Warnungen können vermieden werden, indem man einen der Kontexte als System erscheinen lässt, indem man ihn zu Beginn hinzufügt:

\override VerticalAxisGroup.staff-affinity = ##f

2 Dateien mit convert-ly aktualisieren


2.1 Warum verändert sich die Syntax?


Beispielsweise alle \paper- und \layout-Eigenschaftsnamen sollen in der Form erstens-zweitens-drittens geschrieben werden. In der Version 2.11.60 bemerkten wir jedoch, dass die printallheaders-Eigenschaft sich nicht an diese Regel hielt. Sollten wir das jetzt lassen (womit neue Benutzer verwirrt werden, weil die Eingabe nicht logisch ist), oder sollten wir es ändern (womit wir alte Benutzer mit ihren schon geschriebenen Partituren ärgeren)? In diesem Fall haben wir uns entschieden, den Namen in print-all-headers zu ändern. Zum Glück kann diese Änderung mit dem convert-ly-Programm automatisch vorgenommen werden.


2.2 convert-ly aufrufen

convert-ly benutzt den Befehl \version mit Angabe der Versionsnummer der ursprünglichen LilyPond-Version. In den meisten Fällen genügt es, einfach auf der Kommandozeile

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

im Verzeichnis, in welchem die Datei liegt, aufzurufen. Hierdurch wird meineDatei.ly direkt aktualisiert und das Original nach meineDatei.ly~ gesichert.

**Achtung:** convert-ly konvertiert immer bis zur letzten Syntax-Änderung, die das Programm beherrscht. Das heißt, dass die \version-Nummer, die nach der Konversion in der Datei steht, normalerweise niedriger ist als die Version von convert-ly selbst.

Um alle Dateien in einem Verzeichnis zu konvertieren, schreibt man auf der Kommandozeile:

```
convert-ly -e *.ly
```

Man kann auch einen neuen Namen für die konvertierte Datei angeben, sodass die originale Datei unverändert bleibt. Dazu schreibt man auf der Kommandozeile:

```
convert-ly meineDatei.ly > meineneueDatei.ly
```

Das Programm gibt die Versionsnummern für alle Versionen aus, für die eine Konversion durchgeführt wurde. Wenn keine Versionsnummern ausgegeben werden, ist die Datei aktuell.

MacOS X-Benutzer können die Befehle unter dem Menü-Eintrag Compile > Update syntax ausführen.
Windows-Benutzer sollten diese Befehle auf der Kommandozeile (Eingabeaufforderung), die sich normalerweise unter Start > Zubehör > Eingabeaufforderung findet.

### 2.3 Optionen auf der Kommandozeile für `convert-ly`

Das Programm wird folgendermaßen aufgerufen:

```bash
class Name [Option]... Dateiname...
```

Folgende Optionen können benutzt werden:

- `-e, --edit`
  Die Konvertierung direkt am Original vornehmen, sodass es direkt verändert wird.

- `-f, --from=Von-Versionsnummer`
  Stellt die Versionsnummer ein, ab welcher die Konvertierung begonnen werden soll. Wenn die Option nicht benutzt wird, rät `convert-ly` die Versionsnummer anhand des `\version`-Eintrags in der Datei. Beispielsweise `--from=2.10.25`

- `-n, --no-version`
  Normalerweise fügt `convert-ly` einen `\version`-Eintrag zu der konvertierten Datei hinzu. Mit dieser Option wird das unterdrückt.

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

- `-to=bis-Versionsnummer`
  Die Zielversion der Konversion setzen. Standard ist die letzte mögliche Version, die das Programm beherrscht. Beispielsweise `--to=2.12.2`

- `-h, --help`
  Zeigt Hilfe zur Benutzung.

- `-l Logstufe, --loglevel=Logstufe`
  Passt die Ausführlichkeit der Ausgabe entsprechend `Logstufe` an. Mögliche Werte sind `NONE`, `ERROR`, `WARNING`, `PROGRESS` (Standard) und `DEBUG`.

Um LilyPond-Schnipsel in texinfo-Dateien zu aktualisieren, kann

```bash
class Name --from=... --to=... --no-version *.itely
```

benutzt werden.

Um sich die Änderungen der LilyPond-Syntax zwischen zwei Versionen anzeigen zu lassen, schreibt man

```bash
class Name --from=... --to=... -s
```

### 2.4 Probleme mit `convert-ly`

Wenn man `convert-ly` auf einer Eingabeaufforderung unter Windows mit einer Datei benutzt, die Leerzeichen im Dateinamen oder Pfad hat, muss der gesamte Dateiname mit drei (!) doppelten Anführungszeichen umgeben werden:

```bash
class Name ""D:/My Scores/Ode.ly"" > "D:/My Scores/new Ode.ly"
```

Wenn der einfache `convert-ly -e *.ly-Befehl` nicht funktioniert, weil die ausgeschriebene Kommandozeile zu lang wird, kann man `convert-ly` auch als Loop wiederholt laufen lassen. Dieses Beispiel für UNIX konvertiert alle -ly-Dateien im aktuellen Verzeichnis:

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

Für die Windows-Eingabeaufforderung lautet der entsprechende Befehl:

```bash
for %x in (*.ly) do convert-ly -e "%%%x%%"
```

2.5 Manuelle Konversion

Theoretisch könnte ein Programm wie convert-ly alle möglichen Syntax-Änderungen berücksichtigen. Schließlich ist es auch ein Computerprogramm, das die alte und die neue Version der Notationsdatei interpretiert, so dass ein anderes Computerprogramm eine Datei in die andere verwandeln könnte.\(^1\)

Das LilyPond-Team ist jedoch verhältnismäßig klein, sodass nicht alle Konversionen automatisch funktionieren. Unten eine Liste der bekannten Probleme:

1.6->2.0:
Bezifferter Bass wird nicht immer richtig konvertiert, besonders \(< >\). Mats' Kommentar zu einer Lösung:

To be able to run convert-ly on it, I first replaced all occurrences of '{<' to some dummy like '{#' and similarly I replaced '>' with '&}'. After the conversion, I could then change back from '{ #'} to '{ <' and from '& }' to '>' }'.

Nicht alle Textbeschriftung wird richtig konvertiert. In der alten Syntax konnte man mehrere Beschriftungen mit Klammern gruppieren, etwa:

-\#"{ (bold italic) "string"}"

Das wird falsch konvertiert zu:

-\markup{{\bold italic} "string"}

anstelle von:

-\markup{\bold \italic "string"}

2.0->2.2:
Versteht nicht \partcombine
Kann nicht \addlyrics => \lyricsto, sodass einige Dateien mit vielen Strophen nicht funktionieren

2.0->2.4:
\magnify wird nicht nach \fontsize verändert.
-\magnify #m => \fontsize #f, where \( f = 6 \ln(m)/\ln(2) \)
remove-tag wird nicht verändert.
-\applyMusic #(remove-tag '... ) => \keepWithTag '#... 
first-page-number wird nicht verändert.
-\first-page-number no => print-first-page-number = ##f
Zeilenumbrüche in Titelköpfen werden nicht umgewandelt.
-\\\ als Zeilenumbruch in \header-Feldern => \markup \center-align < "Erste Zeile" "Zweite Zeile" >
Crescendo und decrescendo-Endpunkte werden nicht umgewandelt.
-\rcode => \!
-\rc => \!

2.2->2.4:
\turnOff (benutzt in \set Staff.VoltaBracket = \turnOff) wird nicht richtig konvertiert

2.4.2->2.5.9
\markup{ \center-align \{ \line { ... }\} } sollte konvertiert werden in:
\markup{ \center-align {line { ... }} }

\(^1\) Das ist auf jeden Fall möglich für jede LilyPond-Datei, die kein Scheme beinhaltet. Wenn Scheme in der Datei verwendet wurde, enthält die Datei Turing-complete Sprache und es gibt Probleme mit dem „Halteproblem“ der Informatik.
jetzt fehlt aber \line.
2.4->2.6
Besondere LaTeX-Zeichen wie $\sim$ in Text werden nicht in UTF-8 umgewandelt.
2.8
\score{} muss jetzt immer mit einem musikalischen Ausdruck beginnen. Alles andere (insbesondere \header{}) darf erst nach den Noten kommen.
3 lilypond-book aufrufen

Wenn Sie in ein Dokument Grafiken Ihres Musiksatzes einfügen möchten, so können Sie genau so vorgehen, wie Sie andere Grafiken einfügen würden: Die Bilder werden getrennt vom Dokument im PostScript- oder PNG-Format erstellt und können dann in \texttt{\LaTeX} oder HTML eingefügt werden.

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


Dieses Vorgehen kann bei \texttt{\LaTeX}, HTML, Texinfo oder DocBook Dokumenten angewendet werden.

3.1 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, \texttt{\LaTeX}, Texinfo und DocBook Dokumente existiert jedoch ein Weg, dies automatisiert durchzuführen.

Das Programm \texttt{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 \texttt{\LaTeX}-Beispiel verdeutlicht werden. Das Beispiel selbst enthält schon Erklärungen, sodass wir es hier nicht weiter diskutieren müssen.

\textbf{Eingabe}

\begin{verbatim}
\documentclass[a4paper]{article}
\begin{document}

Dokumente für \texttt{lilypond-book} können Musik und Text nach Belieben kombinieren. Zum Beispiel:

\begin{lilypond}
\relative {
  c'2 e2 \tuplet 3/2 { f8 a b } a2 e4
}
\end{lilypond}

Optionen für \texttt{lilypond} werden dabei in eckige Klammern gesetzt.

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

Größere Beispiele können auch in einer eigenständigen Datei gespeichert und dann mit \texttt{lilypondfile} eingebunden werden.
\end{verbatim}
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) 2.19.84
Reading lilybook.lytex...
..(viele Ausgabezeilen entfernt)..
Compiling lilybook.tex...
cd out
pdflatex lilybook
..(viele Ausgabezeilen entfernt)..
xdpdf 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.\footnote{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:

\begin{music}
\relative { c' d e f g a b c}
\end{music}

Optionen für lilypond werden dabei in eckige Klammern gesetzt.
\begin{music}
c'4 f16\end{music}

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

\begin{music}
\flag{u5}\relative { \flags{u4} c'4 f16 \flags{u4} c' d e f g a b c \flags{u5}}
\end{music}

Wenn man eine tagline (die Zeile unten auf der Seite) braucht (in Standard oder angepasst), dann muss der ganze Schnipsel in eine \texttt{book {}}-Umgebung.

\begin{music}
\book{
\header{
   title = "A scale in LilyPond"
}
\relative {
   c' d e f g a b c
}}
\end{music}
A scale in LilyPond

Music engraving by LilyPond 2.19.84—www.lilypond.org
3.2 Noten in Text integrieren

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

3.2.1 \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 \LaTeX{}2e-Kurzbeschreibung (http://www.ctan.org/tex-archive/info/lshort/german/) für eine Einführung in die Benutzung von \LaTeX.\lilypond-book stellt folgende Befehle und Umgebungen zu Verfügung, um Noten in \LaTeX-Dateien einzufügen:

- den \lilypond{...}-Befehl, womit man direkt kurze LilyPond-Codeabschnitte schreiben kann
- die \begin{lilypond}...\end{lilypond}-Umgebung, mit der man längere LilyPond-Codeabschnitte direkt schreiben kann
- den \lilypondfile{...}-Befehl um eine LilyPond-Datei einzufügen
- den \musicxmlfile{...}-Befehl um eine MusicXML-Datei einzufügen, die dann von musicxml2ly und lilypond bearbeitet wird.

In der Eingabedatei werden Noten mit beliebigen der folgenden Befehle angegeben:

\begin{lilypond}[Optionen,kommen,hierhin] IHR LILYPOND-QUELLOCODE \end{lilypond}

\lilypond[Optionen,kommen,hier]{ IHR LILYPOND-QUELLOCODE }

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

\musicxmlfile[Optionen,kommen,hier]{Dateiname}

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

erzeugt

\begin{music}
\underline{\text{c\text{"o}}}
\end{music}

Die Kurzversion

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

erzeugt

\begin{music}
\underline{\text{c\text{"o}}}
\end{music}
Kapitel 3: lilypond-book aufrufen

Innerhalb des \lilypond{} Befehls dürfen keine geschwungenen Klammern { oder } vorkommen, weshalb dieser Befehl nur mit der \texttt{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 \texttt{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:

- \texttt{\preLilyPondExample} – wird vor der Musik aufgerufen,
- \texttt{\postLilyPondExample} – wird nach der Musik aufgerufen,
- \texttt{\betweenLilyPondSystem[1]} – wird zwischen den einzelnen Systemen aufgerufen, wenn \texttt{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.

\textbf{Ausgewählte Schnipsel}

Manchmal ist es nötig, Musikelemente wie Halte- oder Bindebögen so darzustellen, als ob sie am Ende des Musikausschnittes 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 \texttt{\betweenLilyPondSystem} einfach derartig programmiert, dass die Ausgabe der einzelnen Notensysteme abgebrochen wird, sobald die gewünschte Anzahl an Systemen erreicht ist. Da \texttt{\betweenLilyPondSystem} zum ersten Mal nach dem ersten System aufgerufen wird, ist die Ausgabe nur eines Systems trivial.

\begin{verbatim}
def\betweenLilyPondSystem#1{\endinput}

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

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

\begin{verbatim}
def\betweenLilyPondSystem#1{\endinput
  \ifnum##1<2\else\endinput\fi}
\end{verbatim}

Dieses Beispiel bricht nach genau zwei ausgegebenen Notenzeilen ab. Für eine andere Anzahl braucht nur \texttt{2} durch die entsprechende Anzahl ersetzt werden.

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

\begin{verbatim}
\let\betweenLilyPondSystem\undefined
\end{verbatim}

Obige Definition von \texttt{\betweenLilyPondSystem} kann durch die Definition eines \LaTeX{}-Makros auch verallgemeinert werden,

\begin{verbatim}
def\onlyFirstNSystems#1{
  \def\betweenLilyPondSystem#1{\ifnum##1<#1\else\endinput\fi}
}\end{verbatim}
wobei diesem Makro `\onlyFirstNSystems` einfach die Anzahl der gewünschten Systeme übergeben wird:

```latex
\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 3.4 [lilypond-book aufrufen], Seite 32.

### 3.2.2 Texinfo


`lilypond-book` stellt die folgenden Befehle und Umgebungen zur Verfügung, im Noten in Texinfo-Dateien einzufügen:

- den `\lilypond{...}`-Befehl, womit man direkt kurze LilyPond-Codeabschnitte schreiben kann
- die `\begin{lilypond}...\end{lilypond}`-Umgebung, mit der man längere LilyPond-Codeabschnitt direkt schreiben kann
- den `\lilypondfile{...}`-Befehl um eine LilyPond-Datei einzufügen
- den `\musicxmlfile{...}`-Befehl um eine MusicXML-Datei einzufügen, die dann von `musicxml2ly` und `lilypond` bearbeitet wird.

In der Eingabedatei werden Noten mit folgenden Befehlen eingegeben:

```latex
@lilypond[Optionen,kommen,hier]
IHR LILYPOND-QUELLODCE
@end lilypond

@lilypond[Optionen,kommen,hier]{ IHR LILYPOND-QUELLODCE }

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

@musicxmlfile[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

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

erzeugt
Die Kurzversion
\lilypond{<c' e' g'>}
erzeugt
\lilypond

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

### 3.2.3 HTML

\lilypond-book stellt folgende Befehle und Umgebungen zur Noteneingabe in HTML-Dateien zur Verfügung:

- den `<lilypond ... />`-Befehl, womit man direkt kurze LilyPond-Codeabschnitte schreiben kann
- die `<lilyond>...</lilyond>`-Umbgebung, mit der man längere LilyPond-Codeabschnitte direkt schreiben kann
- den `<lilypondfile>...</lilypondfile>`-Befehl um eine LilyPond-Datei einzufügen
- den `<musicxmlfile>...</musicxmlfile>`-Befehl um eine MusicXML-Datei einzufügen, die dann von \texttt{musicxml2ly} und \lilypond bearbeitet wird.

In der Eingabedatei werden Noten mit folgenden Befehlen eingegeben:

- `<lilypond Optionen hier> IHR LILYPOND-QUELLCODE </lilypond>`

Man kann beispielsweise schreiben

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

\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.

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

Um Dateien mit Musik einzubinden, kann folgendermaßen vorgegangen werden:

```
<lilypondfile Option1 Option2 ...>Dateiname</lilypondfile>
```
<musicxmlfile> hat die gleiche Syntax wie <lilypondfile>, aber fügt anstelle der LilyPond-Datei eine MusicXML-Datei ein.

Eine Liste der Optionen, die man mit lilypond oder lilypondfile benutzen kann, siehe Abschnitt 3.3 [Die Musikfragment-Optionen], Seite 29.

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

3.2.4 DocBook


Definitionen


Eine LilyPond-Datei einfügen

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

```xml
<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:

```xml
<inlinemediaobject>
  <textobject>
    <programlisting language="lilypond" role="fragment verbatim staffsize=16 ragged-right">
      \context Staff \\
      \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


3.3 Die Musikfragment-Optionen


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

Die folgenden Optionen können für LilyPond-Befehle benutzt werden:

staffsize=ht

Setzt die Höhe einer Notenzeile auf ht, angegeben in Punkten.

ragged-right

Erzeugt Notenzeilen im Flattersatz mit natürlichen Abstand der Noten. In anderen Worten: \texttt{ragged-right = \#t} wird in das Musikfragment eingefügt. Einzeilige Fragmentschnipsel werden standardmäßig immer im Flattersatz gesetzt, außer \texttt{noragged-right} wird explizit angegeben.

noragged-right

Streckt Musikfragmente mit nur einer Notenzeile auf die volle Breite, es wird also \texttt{ragged-right = \#f} in das Musikfragment eingefügt.

line-width

\texttt{line-width=\textit{Breite}\textbackslash Einheit}

Setzt die Breite der Notenzeilen auf \textit{Breite}, gemessen in Vielfachen der \textit{Einheit}. Als Einheit können die folgenden Zeichenfolgen angegeben werden: \texttt{cm, mm, in} oder \texttt{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 \texttt{line-width}-Option nicht angegeben wird, versucht lilypond-book einen geeigneten Standardwert für alle lilypond-Umgebungen zu finden, die die \texttt{ragged-right}-Option nicht benutzen.

papersize=Zeichenkette

Wobei \textit{Zeichenkette} eine Papiergröße wie definiert in \texttt{scm/paper.scm} ist, beispielsweise \texttt{a5, quarto, 11x17} usw.

Werte, die nicht in \texttt{scm/paper.scm} definiert sind, werden ignoriert, eine Warnung wird ausgegeben und das Schnipsel wird mit dem Standardwert \texttt{a4} ausgegeben.

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.

\texttt{c'4}

ohne \texttt{\layout, \score}, etc. eingegeben werden kann.
Kapitel 3: lilypond-book aufrufen

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

indent=Einzug\Einheit

noindent

quote
Verringert die Zeilenlänge des Musikfragments um \(2 \times 0.4\) in und setzt das Fragment in einen Zitat-Block. Der Wert von ',0.4\)in' kann durch die exampleindent Optionangepasst 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 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 einzufügen zu lassen: Wenn die eingebundene LilyPond-Datei ein Kommentar mit dem Inhalt ‘begin verbatim’ (ohne Anführungszeichen) enthält, wird nur der Dateinhalt ab dieser Position eingefügt. Enthält die Datei mehrere solche Kommentare, wirkt nur das letzte. Analog wird nur der Dateinhalt 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 { \% begin verbatim
  c'4 e2 g4
  f2 e \% end verbatim
}

1 Die intertext Option ist noch nicht implementiert.
erzeugt ein Zitat der Form
\begin{verbatim}
c4 e2 g4
f2 e
\end{verbatim}
Wenn Kommentare und Variablen im Zitat, aber nicht im Quelltext übersetzt werden sollen, kann die Umgebungsvariable \texttt{LYDOC_LOCALEDIR} auf einen Verzeichnispfad gesetzt werden. Das Verzeichnis sollte einen Baum an .mo-Nachrichtenkatalogen beinhalten mit \texttt{lilypond-doc} als Domain.

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

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

Enthält also die Datei \texttt{foo.ly} etwa den LilyPond-Code
\begin{verbatim}
\header {
    texidoc = "Dieses Beispiel zeigt eine einzelne Note."
}
{ c'4 }
\end{verbatim}
und das Texinfo-Dokument \texttt{text.texinfo}
\texttt{@lilypondfile[texidoc]{foo.ly}}
so liefert der folgende Aufruf von \texttt{lilypond-book} das gewünschte Ergebnis:
\begin{verbatim}
lilypond-book --pdf --process="lilypond \-dbackend=eps --header=texidoc" test.texinfo
\end{verbatim}
Die meisten Test-Dateien (im \texttt{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 \texttt{\documentlanguage LANG} und der \texttt{\header} Block in der Datei \texttt{foo.ly} die Übersetzung im Feld \texttt{texidocLANG} enthalten. Wird nun \texttt{lilypond} mit der Option \texttt{--header=texidocLANG} aufgerufen, so wird der Inhalt der Datei \texttt{foo.texidocLANG} anstelle von \texttt{foo.texidoc} eingefügt.

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

\texttt{nogettext}
(Nur für Texinfo-Ausgabe: Kommentare und Variablenbezeichnungen im zitierten Quelltext des Schnipsel werden nicht übersetzt.

\texttt{printfilename}
Wenn eine LilyPond-Datei mittels \texttt{lilypondfile} und dieser Option eingebunden wird, wird der Dateiname (ohne die Pfadangabe) unmittelbar vor dem Musikfrag-
ment ausgegeben. In HTML-Dateien ist er außerdem ein Link auf die LilyPond-Datei. Nur der eigentliche Name der Datei wird ausgegeben, der Pfad wird also nicht mit angezeigt.

### 3.4 lilypond-book aufrufen


#### Formatabhängige Anweisungen

\texttt{\LaTeX}

Es existieren zwei Methoden, Ihr \texttt{\LaTeX}-Dokument weiter zu verarbeiten, um zu einer druck- oder publikationsfähigen Datei zu gelangen: Zum einen die direkte Erzeugung einer PDF-Datei mit PDF\texttt{\LaTeX}, zum anderen die Erzeugung einer DVI daraus einer PostScript-Datei mittels \texttt{\LaTeX} und einem DVI-nach-PostScript Konverter wie \texttt{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 \texttt{ps2pdf} und \texttt{pdf2ps} (aus dem Ghostscript-Paket) erfolgen.

Um eine PDF-Datei mittels PDF\texttt{\LaTeX} zu erzeugen, kann folgendermaßen vorgegangen werden:

\begin{verbatim}
lilypond-book --pdf Ihre_Datei.pdftex
pdflatex Ihre_Datei.tex
\end{verbatim}

Um eine PDF-Datei mittels \texttt{\LaTeX}/\texttt{dvips}/\texttt{ps2pdf} zu erhalten, sind folgende Befehle nötig:

\begin{verbatim}
lilypond-book Ihre_Datei.lytex
latex Ihre_Datei.tex
dvips -Ppdf Ihre_Datei.dvi
ps2pdf Ihre_Datei.ps
\end{verbatim}

Die .dvi-Datei, die beim Aufruf von \texttt{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 \texttt{dvips} kann einige Warnungen über fehlende Schriftarten ausgeben. Auch dies ist in Ordnung und kann ignoriert werden.

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

#### Bekannte Probleme und Warnungen

Der Befehl \texttt{\pageBreak} funktioniert nicht innerhalb einer \texttt{\begin{lilypond} \ldots \end{lilypond}}-Umgebung.

Auch viele Variablen der \texttt{\paper} Umgebung funktionieren nicht innerhalb einer \texttt{\begin{lilypond} \ldots \end{lilypond}}-Umgebung. Benutzen Sie \texttt{\newcommand} mit \texttt{\betweenLilyPondSystem}:

\begin{verbatim}
\newcommand{\betweenLilyPondSystem}[1]{\vspace{36mm}\linebreak}
\end{verbatim}

\(^2\) Manchmal kann eine Datei entweder von PDF\texttt{\LaTeX} oder von \texttt{\LaTeX} nicht korrekt verarbeitet werden, weshalb hier beide Methoden beschrieben werden.
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.

Optionen auf der Kommandozeile
lilypond-book unterstützt die folgenden Kommandozeilenoptionen:

-\texttt{-f \textit{Format}}
\textbf{--format=\textit{Format}}


-\texttt{-F \textit{Filter}}
\textbf{--filter=\textit{Filter}}

Leitet die Musikfragmente durch das Programm filter anstatt sie mit LilyPond zu einer Grafik zu verarbeiten. --filter und --process kann nicht gleichzeitig benutzt werden. Beispielaufruf:

\begin{verbatim}
lilypond-book --filter='convert-ly --from=2.0.0 -' Mein-Buch.tely
\end{verbatim}

-\texttt{-h}
\textbf{--help}

Gibt eine kurze Hilfemeldung aus.

-\texttt{-I \textit{Pfad}}
\textbf{--include=\textit{Pfad}}


-\texttt{-l \textit{Logstufe}}
\textbf{--loglevel=\textit{Logstufe}}

Passt die Ausführlichkeit der Ausgabe entsprechend Logstufe an. Mögliche Werte sind NONE, ERROR, WARNING, PROGRESS (Standard) und DEBUG. Wenn diese Option nicht genutzt wird und die Umgebungsvariable LILYPOND_BOOK_LOGLEVEL definiert ist, wird ihr Wert als Logstufe eingesetzt.

-\texttt{-o \textit{Verzeichnis}}
\textbf{--output=\textit{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.

\begin{verbatim}
lilypond-book --output=out IhreDatei.lytex
cd out
\end{verbatim}
Kapitel 3: lilypond-book aufrufen

... 

--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

--lily-loglevel=Logstufe
Passt die Ausführlichkeit der Ausgabe entsprechend Logstufe an. Mögliche Werte sind NONE, ERROR, WARNING, PROGRESS (Standard) und DEBUG. Wenn diese Option nicht genutzt wird und die Umgebungsvariable LILYPOND_LOGLEVEL definiert ist, wird ihr Wert als Logstufe eingesetzt.

--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 nützlich, 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 ist 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 Taktenahlen und Bezeichnungen der Instrumente. Diese Option verkürzt die Notenzeile und verschiebt sie um denselben Betrag nach rechts.

-P Befehl
--process=Befehl
Verarbeitet LilyPond-Fragmente mit Befehl anstelle des Standardbefehls lilypond.
--filter und --process können mit lilypond-book nicht gleichzeitig benutzt werden.

--pdf
Erzeugt PDF-Dateien mit PDFLaTeX.

--redirect-lilypond-output
Standardmäßig wird die Ausgabe auf der Kommandozeile ausgegeben. Diese Option leitet die Ausgabe in eine Log-Datei im selben Verzeichnis wie die Quelldatei um.

--use-source-file-names

-V
--verbose
Gibt ausführliche informative Meldungen aus. Äquivalent zu --loglevel=DEBUG.
Kapitel 3: lilypond-book aufrufen

-v
--version

Gibt die Versionsnummer aus.

Bekannte Probleme und Warnungen

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

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

3.5 Dateiendungen


<table>
<thead>
<tr>
<th>Erweiterung</th>
<th>Ausgabeformat</th>
</tr>
</thead>
<tbody>
<tr>
<td>.html</td>
<td>HTML</td>
</tr>
<tr>
<td>.itely</td>
<td>Texinfo</td>
</tr>
<tr>
<td>.latex</td>
<td>\TeX</td>
</tr>
<tr>
<td>.lytex</td>
<td>\TeX</td>
</tr>
<tr>
<td>.lyxml</td>
<td>DocBook</td>
</tr>
<tr>
<td>.tely</td>
<td>Texinfo</td>
</tr>
<tr>
<td>.tex</td>
<td>\TeX</td>
</tr>
<tr>
<td>.texi</td>
<td>Texinfo</td>
</tr>
<tr>
<td>.texinfo</td>
<td>Texinfo</td>
</tr>
<tr>
<td>.xml</td>
<td>HTML</td>
</tr>
</tbody>
</table>


3.6 lilypond-book-Vorlagen


3.6.1 LaTeX

Sie können LilyPond-Fragmente in ein \TeX-Dokument einfügen:

\documentclass{article}

\begin{document}

Normaler \TeX-Text.

\begin{lilypond}
\relative {
   a'4 b c d
}

\end{lilypond}

\end{document}
\end{lilypond}

Weiterer LaTeX-Text und Optionen in eckigen Klammern.

\begin{lilypond}[fragment,relative=2,quote,staffsize=26,verbatim]
d4 c b a
\end{lilypond}
\end{document}

3.6.2 Texinfo

LilyPond-Fragmente können in Texinfo-Dokumente eingefügt werden: dieses gesamte Handbuch wurde in Texinfo geschrieben.

\input texinfo @node Top @top

Texinfo-Text

@lilypond
\relative {
  a'4 b c d
}
@end lilypond

Weiterer Texinfo-Text und Optionen in Klammern.

@lilypond[verbatim,fragment,ragged-right]
d4 c b a
@end lilypond

@bye

3.6.3 html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<!--[-- header_tag -->
<HTML>
<body>

<p>
Dokumente für lilypond-book können Noten und Text frei kombinieren. Zum Beispiel
</p>

<lilypond>
\relative {
  a'4 b c d
}
</lilypond>
</p>

<p>
Noch etwas LilyPond, mit Optionen:
</p>

<lilypond fragment quote staffsize=26 verbatim>
3.6.4 xelatex

```latex
\documentclass{article}
\usepackage{ifxetex}
\ifxetex
  \usepackage{xunicode,fontspec,xltxtra}
  \setmainfont[Numbers=OldStyle]{Times New Roman}
  \setsansfont{Arial}
\else
\fi
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage{mathptmx}%Times
\usepackage{helvet}%Helvetica
\begin{document}
\title{A short document with LilyPond and xelatex}
\maketitle

Normal \textbf{font} commands inside the \emph{text} work, because they \textsf{are supported by \LaTeX{} and XeTeX.} If you want to use specific commands like \verb+\XeTeX+, you should include them again in a \verb+\ifxetex+ environment. You can use this to print the \verb+\ifxetex+ \XeTeX{} command \verb+\fi+ which is not known to normal \LaTeX. 

In normal text you can easily use LilyPond commands, like this:

\begin{lilypond}
\{a2 b c'8 c' c' c'}
\end{lilypond}

\noindent
and so on.

The fonts of snippets set with LilyPond will have to be set from inside of the snippet. For this you should read the AU on how to use
3.7 Das Inhaltsverzeichnis flexibel einsetzen

Diese Funktionen existieren schon im OrchestraLily-Paket:

http://repo.or.cz/w/orchestrallily.git

Um den Text flexibler behandeln zu können, bevorzugen manche Benutzer es, das Inhaltsverzeichnis aus LilyPond zu exportieren und dann mit \LaTeX{} einzulesen.

Das Inhaltsverzeichnis (ToC) aus LilyPond exportieren

Hier wird angenommen, dass in Ihrer Partitur mehrere Sätze vorkommen, die sich alle in der selben LilyPond-Datei befinden.

```lilypond
#(define (oly:create-toc-file layout pages)
  (let* ((label-table (ly:output-def-lookup layout 'label-page-table)))
    (if (not (null? label-table))
      (let* ((format-line (lambda (toc-item)
        (let* ((label (car toc-item))
           (text (caddr toc-item))
           (label-page (and (list? label-table)
                             (assoc label label-table)))
           (page (and label-page (cdr label-page))))
           (format #f "\{\, section, 1, \(\"a\", \"a\" page text label\)\}"
                (formatted-toc-items (map format-line (toc-items))))
           (whole-string (string-join formatted-toc-items ",\n"))
           (output-name (ly:parser-output-name))
           (outfilename (format "\{\,toc\}" output-name))
           (outfile (open-output-file outfilename)))
        (if (output-port? outfile)
            (display whole-string outfile)
            (ly:warning (_ "Unable to open output file " a for the TOC information") outfilename))
            (close-output-port outfile))))

\paper {
  #(define (page-post-process layout pages) (oly:create-toc-file layout pages))
}
```

Das Inhaltsverzeichnis in \LaTeX{} importieren

In der \LaTeX{}-Datei sollte folgendes enthalten:

```latex
\usepackage{pdfpages}
\includescore{nameofthescore}
```

wobei \texttt{\includescore} wie folgt definiert ist:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% \includescore{PossibleExtension}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Read in the TOC entries for a PDF file from the corresponding .toc file.
% This requires some heavy latex tweaking, since reading in things from a file
% and inserting it into the arguments of a macro is not (easily) possible

% Solution by Patrick Fimm on \texttt{\LaTeX} on April 18, 2009:
% \readfile{filename}{\variable}
% \% reads in the contents of the file into \variable (undefined if file
% \% doesn't exist)
% \newread\readfile@f
% \def\readfile@line#1{%
\catcode\^=10\global\read\readfile@f to \readfile@tmp\%
\edef\do{\noexpand\edef\noexpand\tmpmacro{\noexpand\noexpand\tmpmacro}{\readfile@tmp}}\do%
\ifeof\readfile@f\else\endline%```
3.8 Alternative Methoden Text und Musik zu kombinieren

Andere Methoden, Text und Noten miteinander zu kombinieren (ohne lilypond-book zu benutzen) werden beschrieben in Abschnitt 4.4 [LilyPond-Ausgabe in anderen Programmen], Seite 46.
4 Externe Programme

LilyPond kann mit anderen Programmen auf verschiedene Weise interagieren.

4.1 Point and click

Point and click fügt Verlinkung für bestimmte Notationselemente in die PDF-Dokumente ein.

Point and click aktivieren

Point and click erlaubt es, die Noten in der Eingabedatei schnell zu finden, indem man auf sie im PDF-Programm klickt. Das erleichtert es, die Stellen zu finden, die Fehler in der Notation verursachen.

Wenn diese Funktionalität aktiv ist, fügt LilyPond Hyperlinks zur PDF-Datei hinzu. Diese Hyperlinks werden an den Webbrowser gesendet, der einen Texteditor mit dem Cursor an der richtigen Stelle öffnet.

Damit diese Kettenreaktion funktionieren kann, müssen Sie das PDF-Programm dazu anweisen, Hyperlinks zu folgen, indem das Skript `lilypond-invoke-editor`, welches mit LilyPond kommt, verwendet wird.

Für Xpdf unter UNIX sollte folgende Zeile in der Datei `xpdfrc`. Unter UNIX findet man diese Datei entweder in `/etc/xpdfrc` oder als `${HOME}/.xpdfrc`:

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

Das Programm `lilypond-invoke-editor` ist ein kleines Hilfsprogramm. Es ruft einen Editor für besondere `textedit-URIs` auf und einen Webbrowser für andere. Es testet die Umgebungsvariable `EDITOR` nach folgenden Mustern:

- `emacs` das ruft auf
  `emacsclient --no-wait +line:column file`

- `gvim` das ruft auf
  `gvim --remote :line:normcolumn file`

- `nedit` das ruft auf
  `nc -noask +line file`

Die Umgebungsvariable `LYEDITOR` wird benutzt, um dieses Verhalten zu umgehen. Sie enthält die Kommandozeile, mit der der Editor aufgerufen wird, wobei `%{file}s`, `%{column}s` und `%{line}s` mit der Datei, Spalte und Zeile ersetzt wird. Die Einstellung

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

für `LYEDITOR` entspricht dem normalen Aufruf von `emacsclient`.

Die point-and-click-Links vergrößern die Größe des PDFs sehr stark. Um die Größe von PDFs und auch PS-Dateien zu verkleinern, kann point and click ausgeschaltet werden, indem man in der Eingabedatei

```
\pointAndClickOff
```

schreibt. Point and click kann explizit aktiviert werden mit dem Befehl

```
\pointAndClickOn
```

Alternativ können Sie point and click auch mit einer Kommandozeilenoption anschalten:

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

**Achtung:** Sie sollten immer point and click ausschalten, wenn Sie LilyPond-Dateien verteilen wollen, damit keine Informationen über Ihre Dateistrukturen in den Dateien gespeichert werden, was ein Sicherheitsrisiko darstellen könnte.
Selektives point-and-click

Für einige interaktive Anwendungen kann es von Vorteil sein, nur einige Elemente mit Point and click zu aktivieren. Wenn man beispielsweise eine Anwendung erstellen will, die Audio oder Video beginnend von einer angeklickten Note abspielt, würde es unpraktisch sein, wenn die Point-and-click-Zeiger Information eines Bogens oder Versetzungszahlen, die gleichzeitig mit der Note erscheinen, darstellen würden.

Man kann dieses Verhalten erreichen, indem man angibt, welche Ereignisse aufgenommen werden sollen:

- Direkt in der .ly-Datei:
  \pointAndClickTypes #'note-event
  \relative {
    c'2\f( f)
  }
  oder
  #(ly:set-option 'point-and-click 'note-event)
  \relative {
    c'2\f( f)
  }
- Auf der Kommandozeile:
  lilypond -dpoint-and-click=note-event example.ly

Auch mehrere Ereignisse können eingebunden werden:

- Direkt in der .ly-Datei:
  \pointAndClickTypes #'(note-event dynamic-event)
  \relative {
    c'2\f( f)
  }
  oder
  #(ly:set-option 'point-and-click '(note-event dynamic-event))
  \relative {
    c'2\f( f)
  }
- Auf der Kommandozeile:
  lilypond \-e"(ly:set-option 'point-and-click '(note-event dynamic-event))" \example.ly

4.2 Unterstützung von Texteditoren

Verschiedene Editoren unterstützen LilyPond

Emacs-Modus


Als ein Benutzer können Sie Ihren Quellpfad (z. B. `~/site-lisp/) zum Ladepfad hinzufügen, indem Sie die (veränderte) folgende Zeile in Ihr `~/.emacs kopieren:
(setq load-path (append (list (expand-file-name "~/site-lisp")) load-path))

**Vim-Modus**

Für Vim ([http://www.vim.org](http://www.vim.org)) gibt es ein Dateiartplugin, einen Einzugsmodus und einen Syntaxhervorhebungsmodus für LilyPond. Um alle diese Eigenschaften zu aktivieren, erstellen (oder verändern) Sie die Datei `$HOME/.vimrc`, sodass sie folgende Zeilen enthält:
```
filetype off
set runtimepath+=/usr/local/share/lilypond/current/vim/
filetype on
```
Wenn LilyPond nicht in `/usr/local/` installiert ist, müssen Sie den Pfad anpassen. Das wird besprochen in Abschnitt “Mehr Information” in *Handbuch zum Lernen*.

**Andere Editoren**

Andere Editoren (sowohl Texteditoren als auch graphische Editoren) haben Unterstützung für LilyPond, aber ihre besonderen Konfigurationsdateien werden nicht mit LilyPond zusammen verteilt. Lesen Sie die entsprechende Dokumentation zu näheren Fragen. Derartige Editoren finden sich unter Abschnitt “Leichteres Editieren” in *Allgemeine Information*.

### 4.3 Von anderen Formaten konvertieren


Die im Folgenden beschriebenen Programme sind eigenständige Hilfsprogramme und werden üblicherweise von der Kommandozeile aufgerufen. Siehe Abschnitt 1.2 [Benutzung auf der Kommandozeile], Seite 1, 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 “MacOS X” in *Allgemeine Information*.

**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!

#### 4.3.1 midi2ly aufrufen

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


`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
Kapitel 4: Externe Programme

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 Betriebssys-
tems. Für nähere Informationen hierzu siehe Abschnitt 4.3 [Von anderen Formaten konvertieren],
Seite 42.

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.
4.3.2 musicxml2ly aufrufen


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 Abschnitt 4.3 [Von anderen Formaten konvertieren], Seite 42.

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

Die folgenden Kommandozeileoptionen werden von musicxml2ly unterstützt:

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

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

- `-l, --language=SPRACHE`
  Benutzt SPRACHE für die Notenbezeichnungen, etwa "deutsch" für deutsche Notenbezeichnungen.

- `--loglevel=Logstufe`
  Passt die Ausführlichkeit der Ausgabe entsprechend Logstufe an. Mögliche Werte sind NONE, ERROR, WARNING, PROGRESS (Standard) und DEBUG.

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

- `-m --midi`
  Aktiviert die MIDI-Umgebung

- `--no-articulation-directions`
  Konvertiert keine Richtungsangaben (`, `, oder `-`) von Artikulations- und Lautstärkerezeichen.

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

- `--output=Dateiname`

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

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

- `--version`
  Gibt die Versionsnummer aus.
-z, --compressed

4.3.3 abc2ly aufrufen

Achtung: Dieses Programm ist nicht unterstützt und kann aus kommenden LilyPond-Versionen entfernt werden.

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


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

abc2ly [Optionen]... ABC-Datei

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.

Gleichermaßen wird mit

%%LY slyrics more words
der Text, der auf das „slyrics“-Schlüsselwort folgt, in die aktuelle Gesangstextzeile eingefügt.

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.
4.3.4 **etf2ly aufrufen**

_Achtung:_ Dieses Programm ist nicht unterstützt und kann in kommenden LilyPond-Versionen entfernt werden.

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 Kommandozeilen des jeweiligen Betriebssystems. Für nähere Informationen hierzu siehe Abschnitt 4.3 [Von anderen Formaten konvertieren], Seite 42.

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.

4.3.5 **Andere Formate**

LilyPond kommt nicht mit der Unterstützung für andere Formate, aber einige externe Programme können auch LilyPond-Dateien erstellen. Diese finden sich unter Abschnitt „Leichteres Editieren“ in _Allgemeine Information_.

4.4 **LilyPond-Ausgabe in anderen Programmen**

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

**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_.

**LilyPond-Noten in OpenOffice.org integrieren**

Musik im LilyPond-Format kann in OpenOffice.org eingefügt werden mittels OOoLilyPond (http://ooolilypond.sourceforge.net).

**LilyPond-Noten in andere Programme integrieren**

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:

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

Benutzbare Bilddateien können mit folgendem Befehl erzeugt werden:

**EPS**

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

**PNG**

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

Ein transparentes PNG

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

```
lilypond -dbackend=eps -dno-gs-load-fonts -dinclude-eps-fonts \ -dpixmap-format=pngalpha --png myfile.ly
```

### 4.5 Unabhängige include-Abschnitte

Einige Leute haben große (und nützliche!) Code-Abschnitte geschrieben, die man in vielen Projekten verwenden kann. Dieser Code wird möglicherweise auch eines Tages in LilyPond selbst integriert werden, aber bis dahin müssen Sie ihn selber herunterladen und mit \include selber einfügen.

#### 4.5.1 MIDI-Artikulation

LilyPond kann benutzt werden, um MIDI-Ausgabe zu erzeugen, etwa um hörend zu korrigieren, was man notiert hat. Jedoch werden nur Dynamik, explizite Tempoänderungen und die Noten und Dauern selber in der MIDI-Datei aufgenommen.

Das articulate-Projekt ist ein Versuch, mehr Information in eine MIDI-Datei zu kopieren. Es funktioniert, indem man Noten, die sich nicht unter Bögen befinden, verkürzt, um die Noten zu „artikulieren“. Die Stärke der Kürzung hängt von den Artikulationszeichen ab, die an die Noten gehängt werden: Staccato halbiert den Notwert, Tenuto gibt der Note ihre gesamte Dauer usw. Das Skript kann auch Triller und Doppelschläge in MIDI umwandeln, und man könnte auch andere Ornamente implementieren.


#### Bekannte Probleme und Warnungen

Die größte Einschränkung ist, dass man nur die Sachen beeinflussen kann, die man weiß: alles, was nur Textbeschriftung ist (anstelle einer Noteneigenschaft) wird immernoch ignoriert.
5 Vorschläge zum Schreiben von LilyPond-Eingabe-Dateien

Jetzt sind Sie so weit, größere Stücke mit LilyPond zu schreiben – nicht nur die kleinen Beispiele aus der Übung, sondern ganze Stücke. Aber wie geht man das am besten an?

Solange LilyPond Ihre Dateien versteht und die Noten so setzt, wie Sie das wollen, spielt es eigentlich keine Rolle, wie Ihre Dateien aussehen. Es gibt aber trotzdem ein paar Dinge, die man beim Schreiben von LilyPond-Code berücksichtigen sollte.

- Was ist, wenn Sie einen Fehler machen? Die Struktur einer LilyPond-Datei kann es erleichtern (oder erschweren), bestimmte Fehler zu finden.
- Was ist, wenn Sie Ihre Dateien mit jemandem austauschen wollen? Oder Ihre Dateien nach einige Jahren noch einmal überarbeiten wollen? Manche LilyPond-Dateien versteht man auf den ersten Blick, über anderen muss man eine Stunde grübeln, um die Struktur zu ahnen.

5.1 Allgemeine Vorschläge

Hier einige Vorschläge, wie Sie Probleme vermeiden oder lösen können:

- **Schreiben Sie immer mit \version die Versionsnummer in jede Datei.** Beachten Sie, dass in allen Vorlagen die Versionsnummer \version "2.19.21" eingetragen ist. Es empfiehlt sich, in alle Dateien, unabhängig von ihrer Größe, den \version-Befehl einzufügen. Persönliche Erfahrung hat gezeigt, dass es ziemlich frustrierend sein kann zu erinnern, welche Programmversion man etwa vor einem Jahr verwendet hat. Auch convert-ly benötigt die Versionsnummer.


- **Ein Takt pro Textzeile.** Wenn irgendetwas kompliziertes vorkommt, entweder in der Musik selber oder in der Anpassung der Ausgabe, empfiehlt es sich oft, nur einen Takt pro Zeile zu schreiben. Bildschirmplatz zu sparen, indem Sie acht Takte in eine Zeile zwängen, hilft nicht weiter, wenn Sie ihre Datei „debuggen“ müssen.

- **Kommentieren Sie ihre Dateien.** Benutzen Sie entweder Taktmmern (in regelmäßigen Abständen) oder Verweise auf musikalische Themen („Zweites Thema in den Geigen“, „vierte Variation“ usw.). Sie brauchen diese Kommentare vielleicht noch nicht, wenn Sie das Stück notieren, aber spätestens wenn Sie nach ein paar Jahren etwas verändern wollen oder Sie den Quelltext an einen Freund weitergeben wollen, ist es weitaus komplizierter, die Datei struktur ohne Kommentare zu verstehen, als wenn Sie sie rechtzeitig eingefügt hätten.

- **Schreiben Sie Klammern mit Einrückung.** Viele Probleme entstehen durch ungerade Anzahl von { und }-Klammern.

- **Schreiben Sie Tondauerangaben am Anfang von Abschnitten und Bezeichnern.** Wenn Sie beispielsweise c4 d e am Anfang eines Abschnittes schreiben, ersparen Sie sich viele Probleme, wenn Sie ihre Musik eines Tages umarrangieren wollen.

5.2 Das Kopieren von bereits vorhandener Musik

Wenn Sie Musik aus einer fertigen Partitur kopieren (z. B. die LilyPond-Eingabe einer gedruckten Partitur):


- Definieren Sie `mBreak = { \break }` und schreiben Sie `\mBreak` in der Quelldatei immer dann, wenn im Manuskript ein Zeilenumber vorkommt. Das macht es einfacher, die gesetzte Zeile mit den ursprünglichen Noten zu vergleichen. Wenn Sie die Partitur fertig gestellt haben, könnten Sie `mBreak = { }`, also leer definieren, um diese manuellen Zeilenumber zu entfernen. Damit kann dann LilyPond selber entscheiden, wohin es passende Zeilenumber platziert.

- Wenn Sie eine Stimme für ein transponierendes Instrument als eine Variable notieren, wird empfohlen, dass die Noten von

  \transpose c klingende-Tonhöhe {...}

eingefasst werden (wobei `klingende-Tonhöhe` die klingende Tonhöhe des Instruments ist), sodass die Noten innerhalb der Variable für klingendes C geschrieben sind. Sie können die Variable zurücktransponieren, wenn es nötig ist, aber Sie müssen es nicht tun. Fehler in Transpositionen sind selten, wenn die Stimmen für die gleiche Ausgangstonhöhe geschrieben werden.

Denken Sie daran, dass Sie nur von/nach C transponieren. Das heißt, dass die einzigen anderen Tonhöhen, die Sie in Transpositionen benutzen, die Tonhöhen der Instrumente sind, für die Sie schreiben: `bes` für eine B-Trompete oder `aes` für eine As-Klarinette usw.

5.3 Große Projekte

Besonders wenn Sie an größeren Projekten arbeiten, ist es unumgänglich, dass Sie Ihre LilyPond-Dateien klar strukturieren.

- **Verwenden Sie Variablen für jede Stimme.** Innerhalb der Definition sollte so wenig Struktur wie möglich sein. Die Struktur des `\score-` Abschnittes verändert sich am ehesten, während die `violine-` Definition sich wahrscheinlich mit einer neuen Programmversion nicht ändern wird.

```lilypond
\transpose c klingende-Tonhöhe {...}
```

- **Trennen Sie Einstellungen von den Noten.** Diese Empfehlung wurde schon früher gegeben, aber für große Projekte ist es unumgänglich. Muss z. B. die Definition für `fdannp` verändert werden, so braucht man es nur einmal vorzunehmen und die Noten in der Geigenstimme, `violin`, bleiben unberührt.

```lilypond
\transpose c klingende-Tonhöhe {...}
```
Kapitel 5: Vorschläge zum Schreiben von LilyPond-Eingabe-Dateien

\dynamic f \italic \small { 2nd } \hspace #0.1 \dynamic p \\
violin = \relative { \\
g'4\fdannp c'8. e16"}

5.4 Fehlersuche

Früher oder später werden Sie in die Lage kommen, dass LilyPond Ihre Datei nicht kompilieren will. Die Information, die LilyPond während der Übersetzung gibt, können Ihnen helfen, den Fehler zu finden, aber in vielen Fällen müssen Sie nach der Fehlerquelle auf die Suche gehen.

Die besten Hilfsmittel sind in diesem Fall das Zeilen- und Blockkommentar (angezeigt durch `% bzw. `\%{ ... %}`). Wenn Sie nicht bestimmen können, wo sich das Problem befindet, beginnen Sie damit, große Teile des Quelltextes auszukommentieren. Nachdem Sie einen Teil auskommentiert haben, versuchen Sie, die Datei erneut zu übersetzen. Wenn es jetzt funktioniert, muss sich das Problem innerhalb der Kommentare befinden. Wenn es nicht funktioniert, müssen Sie weitere Teile auskommentieren bis sie eine Version haben, die funktioniert.

In Extremfällen bleibt nur noch solch ein Beispiel übrig:

\score { \\
<< \\
% \melody \\
% \harmony \\
% \bass \\
>> \\
\layout{}
}

(also eine Datei ohne Noten).

Geben Sie nicht auf, wenn das vorkommen sollte. Nehmen Sie das Kommentarzeichen von einem Teil wieder weg, sagen wir der Bassstimme, und schauen Sie, ob es funktioniert. Wenn nicht, dann kommentieren Sie die gesamte Bassstimme aus, aber nicht den \bass-Befehl in dem \score-Abschnitt:

\score { \\
\bass = \relative { \\
%{ \\
 c'4 c c c \\
 d d d d \\
%} \\
}
}

Jetzt beginnen Sie damit, langsam Stück für Stück der Bassstimme wieder hineinzunehmen, bis Sie die problematische Zeile finden.

Eine andere nützliche Technik zur Problemlösung ist es, Abschnitt “Minimalbeispiele” in Allgemeine Information zu konstruieren.

5.5 Make und Makefiles


Es gibt Fällen, wenn es sich sehr stark empfiehlt, ein Makefile für das aktuelle Projekt zu erstellen, entweder zur eigenen Bequemlichkeit, oder aber auch als Hilfe für andere, die vielleicht
icht einmal die Quelldateien lesen und verstehen wollen. Insbesondere bei großen Projekten mit vielen eingefügten Dateien und unterschiedlichen Ausgabeoptionen (etwa Partitur, einzelne Stimmen, Dirigierpartitur, Klavierauszug usw.), aber auch bei Projekten, die komplizierte Programmaufrufe zur Verarbeitung erfordern (wenn man etwa mit \texttt{lilypond-book} arbeitet), lohnt sich die Erstellung einer Make-Datei. Diese Dateien können sehr unterschiedliche ausfallen, und ihre Komplexität und Flexibilität kann den Bedürfnissen aber auch Kenntnissen des Schreibers angepasst werden. Das Programm GNU Make ist auf GNU/Linux-Distributionen und MacOS X installiert, aber es ist auch für Windows erhältlich.

Das \texttt{GNU Make Manual} gibt eine vollständige Anleitung, wie \texttt{make} benutzt werden kann. Hier sollen nur einige kleine Blicke auf die vielfältigen Möglichkeiten geworfen werden.


Das erste Beispiel ist für ein Orchesterstück in vier Stätzen und mit der folgenden Dateiarchitektur:

\begin{verbatim}
Symphony/
|--- MIDI/
|--- Makefile
|--- Notes/
|   |--- cello.ily
|   |--- figures.ily
|   |--- horn.ily
|   |--- oboe.ily
|   |--- trioString.ily
|   |--- viola.ily
|   |--- violinOne.ily
|   `-- violinTwo.ily
|--- PDF/
|--- Parts/
|   |--- symphony-cello.ly
|   |--- symphony-horn.ly
|   |--- symphony-oboes.ly
|   |--- symphony-viola.ly
|   |--- symphony-violinOne.ly
|   `-- symphony-violinTwo.ly
|--- Scores/
|   |--- symphony.ly
|   |--- symphonyI.ly
|   |--- symphonyII.ly
|   |--- symphonyIII.ly
|   `-- symphonyIV.ly
   `-- symphonyDefs.ily
\end{verbatim}

Die \texttt{.ly}-Dateien in den Verzeichnissen \texttt{Scores} und \texttt{Parts} erhalten ihre Noten aus \texttt{.ily}-Dateien, die sich im \texttt{Notes}-Verzeichnis befinden:

```
%%% Kopfzeile der Datei "symphony-cello.ly"
\include ../symphonyDefs.ily
\include ../Notes/cello.ily
```
Die Make-Datei hat die Ziele \textit{score} (das gesamte Stück als große Partitur), \textit{movements} (die einzelnen Sätze als große Partitur) und \textit{parts} (die einzelnen Stimmen für die Spieler). Es gibt auch das Ziel \textit{archive}, welches ein Tar-Archiv der Quelldateien erstellt, etwa um die Quellen über das Internet oder per E-Mail zu verteilen. Hier die Make-Datei für GNU/Linux oder MacOS X. Sie sollte unter dem Namen \textit{Makefile} im obersten Verzeichnis des Projektes gespeichert werden:

\begin{center}
\begin{tabular}{|l|}
\hline
\textbf{Achtung:} Wenn ein Ziel oder eine Musterregel definiert ist, müssen die folgenden Zeilen mit Tabulatoren, nicht mit Leerzeichen beginnen. \\
\hline
\end{tabular}
\end{center}

\begin{verbatim}
# Namensstamm der Ausgabedateien
piece = symphony
# finde heraus, wieviele Prozessoren vorhanden sind
CPU_CORES=`cat /proc/cpuinfo | grep -m1 "cpu cores" | sed s/".*": //`
# Der Befehl, um lilypond aufzurufen
LILY_CMD = lilypond -ddelete-intermediate-files \ 
           -dno-point-and-click -djob-count=$(CPU_CORES)

# Die Endungen, die im Makefile benutzt werden
.SUFFIXES: .ly .ily .pdf .midi

# Eingabe- und Ausgabedateien werden in den Verzeichnissen durchsucht,
# die sich in der VPATH-Variable befinden. Alle sind Unterverzeichnisse
# des aktuellen Verzeichnisses (angegeben durch die GNU make-Variable
# `CURDIR').
VPATH = \\
   $(CURDIR)/Scores \\
   $(CURDIR)/PDF \\
   $(CURDIR)/Parts \\
   $(CURDIR)/Notes

# Die Musterregel, um PDF und MIDI-Dateien aus der LY-Eingabedatei
# zu erstellen. Die .pdf-Ausgabedateien werden in das
# `PDF'-Unterverzeichnis abgelegt, die .midi-Dateien in das
# `MIDI'-Unterverzeichnis.
%.pdf %.midi: %.ly
   $(LILY_CMD) $<; \\ # this line begins with a tab
   if test -f "$*.pdf"; then \\
   mv "$*.pdf" PDF/; \\
   fi; \\
   if test -f "$*.midi"; then \\
   mv "$*.midi" MIDI/; \\
   fi

notes = \\
   cello.ily \\
   horn.ily \\
   oboe.ily \\
   viola.ily \\
   violinOne.ily \\
   violinTwo.ily
\end{verbatim}
# Abhängigkeiten der einzelnen Sätze.
$(piece)I.pdf: $(piece)I.ly $(notes)
$(piece)II.pdf: $(piece)II.ly $(notes)
$(piece)III.pdf: $(piece)III.ly $(notes)
$(piece)IV.pdf: $(piece)IV.ly $(notes)

# Abhängigkeiten der großen Partitur.
$(piece).pdf: $(piece).ly $(notes)

# Abhängigkeiten der Stimmen.
$(piece)-cello.pdf: $(piece)-cello.ly cello.ily
$(piece)-horn.pdf: $(piece)-horn.ly horn.ily
$(piece)-oboes.pdf: $(piece)-oboes.ly oboe.ily
$(piece)-viola.pdf: $(piece)-viola.ly viola.ily
$(piece)-violinOne.pdf: $(piece)-violinOne.ly violinOne.ily
$(piece)-violinTwo.pdf: $(piece)-violinTwo.ly violinTwo.ily

# `make score' eintippen, um die große Partitur mit allen vier
# Sätzen als eine Datei zu erstellen.
.PHONY: score
score: $(piece).pdf

# `make parts' tippen, um alle Stimmen zu erstellen.
# `make foo.pdf' tippen, um die Stimme für das Instrument `foo' zu erstellen.
# Beispiel: `make symphony-cello.pdf'.
.PHONY: parts
parts: $(piece)-cello.pdf \ 
   $(piece)-violinOne.pdf \ 
   $(piece)-violinTwo.pdf \ 
   $(piece)-viola.pdf \ 
   $(piece)-oboes.pdf \ 
   $(piece)-horn.pdf

# `make movements' tippen um Dateien für die vier Sätze einzeln zu erstellen.
.PHONY: movements
movements: $(piece)I.pdf \ 
        $(piece)II.pdf \ 
        $(piece)III.pdf \ 
        $(piece)IV.pdf

all: score parts movements

archive:
    tar -cvvf stamitz.tar \       # this line begins with a tab
       --exclude=*pdf --exclude=*~ \
       --exclude=*midi --exclude=*/.tar \
       ./Stamitz/*

Unter Windows ergeben sich bestimmte Komplikationen. Nachdem man GNU Make für
Windows heruntergeladen und installiert hat, muss man den richtigen Pfad in den Umge-
bungsvariablen des Systems setzen, damit die DOS-Kommandozeile das Make-Programm
finden kann. Um das vorzunehmen, kann man mit der rechten Maustaste auf "Arbeitsplatz" klicken, dann
Eigenschaften und Erweitert geklickt werden. Hier wählt man Umgebungsvariablen. In der Liste Systemvariablen wählt man Path und mit einem Klick auf Bearbeiten kann man den Pfad zu der .exe-Datei von GNU Make hinzufügen, der etwa wie folgt aussieht:

   C:\Program Files\GnuWin32\bin


## WINDOWS VERSION
##

piece = symphony
LILY_CMD = lilypond -ddelete-intermediate-files \ 
   -dno-point-and-click \ 
   -djob-count=$(NUMBER_OF_PROCESSORS)

# 8.3 Bezeichnung für CURDIR erhalten (Workaround wg. Leerzeichen in PATH)
workdir = $(shell for /f "tokens=**" %%b in ("$(CURDIR)\") do @echo %%~sb)

.SUFFIXES: .ly .ily .pdf .mid

VPATH = \ 
   $(workdir)/Scores \ 
   $(workdir)/PDF \ 
   $(workdir)/Parts \ 
   $(workdir)/Notes

%.pdf %.mid: %.ly
   $(LILY_CMD) $<  # diese Zeile beginnt mit Tabulator
if exist "$*.pdf" move /Y "$*.pdf" PDF/  # begin with tab
if exist "$*.mid" move /Y "$*.mid" MIDI/  # begin with tab

notes = \ 
   cello.ily \ 
   figures.ily \ 
   horn.ily \ 
   oboe.ily \ 
   trioString.ily \ 
   viola.ily \ 
   violinOne.ily \ 
   violinTwo.ily

$(piece)I.pdf: $(piece)I.ly $(notes)
$(piece)II.pdf: $(piece)II.ly $(notes)
$(piece)III.pdf: $(piece)III.ly $(notes)
$(piece)IV.pdf: $(piece)IV.ly $(notes)

$(piece).pdf: $(piece).ly $(notes)

$(piece)-cello.pdf: $(piece)-cello.ly cello.ily
$(piece)-horn.pdf: $(piece)-horn.ly horn.ily
Kapitel 5: Vorschläge zum Schreiben von LilyPond-Eingabe-Dateien

$(piece)-oboes.pdf: $(piece)-oboes.ly oboe.ily
$(piece)-viola.pdf: $(piece)-viola.ly viola.ily
$(piece)-violinOne.pdf: $(piece)-violinOne.ly violinOne.ily
$(piece)-violinTwo.pdf: $(piece)-violinTwo.ly violinTwo.ily

.PHONY: score
score: $(piece).pdf

.PHONY: parts
parts: $(piece)-cello.pdf \ 
   $(piece)-violinOne.pdf \ 
   $(piece)-violinTwo.pdf \ 
   $(piece)-viola.pdf \ 
   $(piece)-oboes.pdf \ 
   $(piece)-horn.pdf

.PHONY: movements
movements: $(piece)I.pdf \ 
   $(piece)II.pdf \ 
   $(piece)III.pdf \ 
   $(piece)IV.pdf

all: score parts movements

Die nächste Make-Datei ist für ein lilypond-book-Dokument, das in LaTeX gesetzt wird. Das Projekt hat einen Index, welcher erfordert, dass der Befehl latex zweimal aufgerufen wird, um die Verweise zu aktualisieren. Ausgabedateien werden in einem out-Verzeichnis für die .pdf-Dateien gespeichert und in htmlout für die html-Dateien.

SHELL=/bin/sh
FILE=myproject
OUTDIR=out
WEBDIR=htmlout
VIEWER=acroread
BROWSER=firefox
LILYBOOK_PDF=lilypond-book --output=$(OUTDIR) --pdf $(FILE).lytex
LILYBOOK_HTML=lilypond-book --output=$(WEBDIR) $(FILE).lytex
PDF=cd $(OUTDIR) && pdflatex $(FILE)
HTML=cd $(WEBDIR) && latex2html $(FILE)
INDEX=cd $(OUTDIR) && makeindex $(FILE)
PREVIEW=$(VIEWER) $(OUTDIR)/$(FILE).pdf &

all: pdf web keep

pdf:
   $(LILYBOOK_PDF) # begin with tab
   $(PDF) # begin with tab
   $(INDEX) # begin with tab
   $(PDF) # begin with tab
   $(PREVIEW) # begin with tab

web:
   $(LILYBOOK_HTML) # begin with tab
Kapitel 5: Vorschläge zum Schreiben von LilyPond-Eingabe-Dateien

\begin{verbatim}
$(HTML)  # begin with tab
cp -R $(WEBDIR)/$(FILE)/ ./ # begin with tab
$(BROWSER) $(FILE)/$(FILE).html & # begin with tab

keep: pdf
  cp $(OUTDIR)/$(FILE).pdf $(FILE).pdf # begin with tab

clean:
  rm -rf $(OUTDIR) # begin with tab

web-clean:
  rm -rf $(WEBDIR) # begin with tab

archive:
  tar -cvvf myproject.tar \ # begin this line with tab
    --exclude=out/* \ 
    --exclude=htmlout/* \ 
    --exclude=myproject/* \ 
    --exclude=*midi \ 
    --exclude=*pdf \ 
    --exclude=*~ \ 
    ../MyProject/*

TODO: soll auch unter Windows funktionieren


lilypond-book --output=out --pdf myproject.lytex
  cd out
  pdflatex myproject
  makeindex myproject
  pdflatex myproject
  cd ..
  copy out\myproject.pdf MyProject.pdf

Siehe auch

\end{verbatim}
Anhang A GNU Free Documentation License

Version 1.3, 3 November 2008
http://fsf.org/

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 functional and useful 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, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.
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. (Thus, 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. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.
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 Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.
A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document 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, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. 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, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include 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, PostScript or PDF 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.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”. ) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

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 (or copies in media that commonly have printed covers) 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 computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. 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 fewer than five), unless they release you from this requirement.

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”, Preserve 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. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements 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 to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

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, and that you preserve all their Warranty Disclaimers.

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 “Acknowledgements”, 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, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which 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 half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket 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, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.
Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

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. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.
ADDENDUM: How to use this License for your documents

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 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled ``GNU Free Documentation License''.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being list their titles, with the Front-Cover Texts being list, and with the Back-Cover Texts being list.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

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

## A

ABC ................................................. 45  
Aktualisierung von alten Eingabedateien .......... 16  
Aktualisierung von LilyPond-Datei .......... 16  
Aufruf von dvips .................................. 32  
Ausgabe, Ausführlichkeit .......................... 4  
Ausgabedatei, Dateigröße .......................... 40  
Ausgabedateiname .................................. 4  
Ausgabeformat ........................................ 2  
Auswertung von Ausdrücken, Scheme .............. 2

## B

Benutzung auf der Komandozeile ............... 56

## C

Coda Technology ...................................... 46  
convert-ly ............................................ 16

## D

Dateigröße, PDF ...................................... 40  
Dateiname der Ausgabe bestimmen .................. 4  
Dateisuche ............................................. 2  
docbook ............................................... 20  
DocBook, Noten hinzufügen ............................ 20  
Dokument, Noten einfügen ............................. 20  
dvips .................................................. 32

## E

Editoren ................................................. 41  
Eingabe von Text ....................................... 14  
emacs ............................................... 41  
enigma ............................................... 46  
error messages ....................................... 12  
ETF .................................................. 46  
Externe Programme, LilyPond-Dateien erstellen 46

## F

Fataler Fehler .......................................... 12  
Fehler .................................................. 12  
Fehlermeldung, Format .................................. 12  
Fehlermeldungen ....................................... 12  
Fehlerprotokoll, Scheme .............................. 12  
Finale ............................................... 46  
Flexible vertikale Abstände in Systemgruppen .... 15  
Form der Fehlermeldungen ............................ 12  
Formatvorlagen ....................................... 48

## H

Header in \LaTeX-Dokumenten ......................... 25  
Hervorhebung, Syntax .................................. 41  
HTML ............................................... 20  
HTML, Noten hinzufügen .............................. 20

## J

Jail, Programms ausführen ........................... 2

## K

Kerker, Programms ausführen ....................... 2  
Kommandozeile, lilypond aufrufen ................. 2  
Kommandozeilen-Optionen für lilypond .......... 1  
Konturschriften ...................................... 32  
Korrigierte Musik überspringen .................... 49

## L

LANG .................................................. 10  
LaTeX ............................................... 20  
LaTeX, Noten in .................................... 20  
Leichteres Editieren .................................. 42, 46  
lilypond auf der Komandozeile ..................... 2  
lilypond aufrufen .................................... 1  
lilypond-book ........................................ 56  
LILYPOND_DATA_DIR .................................. 10  
LILYPOND_LOGLEVEL .................................. 10  
Logstufe ............................................ 4

## M

MacOS X ............................................... 1, 20, 42  
make ............................................... 50  
Make-Dateien ........................................ 50  
Makefile ............................................ 50  
Manuals .............................................. 1  
Mehr Information ..................................... 42  
MIDI ................................................. 42  
Minimalbeispiele .................................... 50  
Modi, Editor ........................................ 41  
MusicXML ............................................. 44  
Musikwissenschaft .................................... 20

## N

Notationsfragmente extrahieren ................... 46

## O

Oktavenüberprüfung ................................ 48  
OpenOffice.org ...................................... 46  
Optionen an der Komandozeile ..................... 1  
Optionen, Komandozeile ............................. 2
P
PDF-Ausgabe ........................................... 4
PNG-Ausgabe ........................................... 4
point and click ........................................ 40
Point and Click, Kommandozeile ...................... 5
Portable Document Format (PDF) ....................... 4
Portable Network Graphics (PNG) ...................... 4
PostScript-Ausgabe .................................... 4
Programmierfehler .................................... 12
pspdfopt ................................................ 4

S
Scheme Fehler ......................................... 12
Scheme, Auswertung von Ausdrücken ................ 2
Staff ...................................................... 42
Suchpfad ................................................ 2
Syntax highlight ........................................ 41
Syntaxhervorhebung .................................... 41

T
Takt- und Taktzahlüberprüfung .......................... 13, 48
Taktstriche .............................................. 13
Terminal, lilypond aufrufen ............................ 2
texi ..................................................... 20
texinfo .................................................. 20
Texinfo, Noten hinzufügen .............................. 20
Tipparbeit durch Variablen und Funktionen einsparen ........ 48
Titel und lilypond-book ............................... 25
Tutorium ................................................. 1
Typ1 Schriften .......................................... 32

U
Umgebungsvariablen .................................... 10
Update von alten Eingabedateien .................... 16

V
Variablen, Umgebungs- ................................ 10
Verzeichnis, Ausgabe speichern in .................... 4
vim ...................................................... 41
Voice .................................................... 42

W
Warnung ................................................ 12
Wie eine LilyPond-Eingabe-Datei funktioniert ......... 14
Windows ................................................. 20

Z
Zeichenkodierung ...................................... 15