



# Extract PDFmarkによる PDFファイルサイズ削減

細田 真道

<http://www.trueroad.jp>

2017年10月14日

# 自己紹介

- 楽譜作成プログラム LilyPond コミッタ
  - ビルドシステム、フォント、PDF 等
- GNU 公式文書フォーマット Texinfo コミッタ
  - X<sub>Y</sub>TeX / LuaTeX、Unicode、日本語対応等
- 第 10 回日本 OSS 奨励賞受賞
  - LilyPond

**URL** <http://www.trueroad.jp>

**GitHub** trueroad

**Twitter** @trueroad\_jp

**Facebook** trueroad.jp

**GPG Key fingerprint**

49B8 ED79 B6A8 C46E 2F6D ABB3 FCD0 C162 1E80 A02D

## 1. はじめに

## 2. PDF

- しおり
- ページモード
- ハイパーリンク
- フォント

## 3. LilyPond

## 4. Texinfo

## 5. $\text{\TeX}$ で図を貼り込む

- 図のフォント
- フォント重複の解消
  - フルセット（非サブセット）埋め込み

- 非埋め込み

- Ghostscriptで失われるもの

## 6. Extract PDFmark

- インストール
- 仕組み
  - pdfmark
  - 抽出
  - Ghostscriptの入力
- 使い方
- 注意事項

## 7. おわりに

# はじめに

# はじめに

- $\text{T}_{\text{E}}\text{X}$  /  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  で PDF ドキュメントを作成
  - 図としてたくさんの小さな PDF を用意
  - メインの PDF へ貼り付ける
- 図 PDF は同じフォントを使っていることが多い

# LilyPond の場合

- マニュアルは Texinfo 形式
  - $\text{X}\text{\_}\text{Y}\text{\_}\text{T}\text{\_}\text{E}\text{\_}\text{X}$  で処理し、PDF を生成
- 楽譜作成プログラムなので、、、
  - マニュアルには楽譜の断片を多数含む
    - LilyPond で PDF として生成
    - 図として貼り込む
  - もちろん同じフォントが多い

# 図 PDF のフォント

- 図 PDF にフォントが埋め込まれていると、、、
  - そのままメイン PDF に埋め込まれる
- 複数の図に同じフォントが埋め込まれていると
  - メイン PDF にフォントが重複して複数回埋め込まれる
- ファイルサイズの増加につながる

# フォント重複防止

- ファイルサイズを削減するには？
  - 図 PDF 作成時に埋め込み方法を工夫
  - メイン PDF を Ghostscript で処理
- しかし
  - Ghostscript で処理すると失われるものが



# 失われるもの

- PDF ページモード
  - PDFを開いたときにどんな表示にするか
- リンクの宛先名
  - 外部から場所を指定したリンク
- Extract PDFmark を使えば保持できます

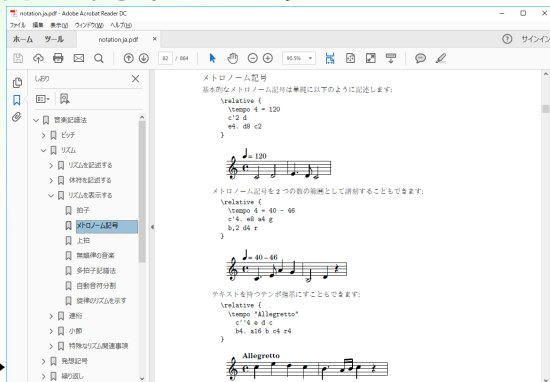
# PDF

# PDF

- PDF の機能
  - しおり
    - ブックマーク、アウトライン、とも
  - ページモード
  - ハイパーリンク
  - フォント

# しおり

- 文書構造をツリー状に表示
  - 章・節など
- 目的の部分へ簡単にジャンプできる



ここ→

# ページモード

- ページモードを指定しておく
  - PDFを開いたとき、最初からしおりが表示される、など

# ハイパーリンク

- リンクする
  - URL
  - 同じ PDF 内のどこか
  - 他の PDF のどこか、など
- リンクされる
  - 宛先名 (named destination) を設定
    - 名前と場所を指定
    - 名前に設定された場所へジャンプできる
  - PDF 外部から名前を指定してリンク
    - PDF 相互間
    - HTML からのリンク、など

# フォント

- フォントが無い環境でも正しく表示
- 埋め込み
  - フルセット（非サブセット）埋め込み
    - フォントを丸々フルセットで埋め込む
    - ファイルサイズが大きくなる
  - サブセット埋め込み
    - 使用しているグリフのみ埋め込む
    - ファイルサイズを抑え正しく表示可
  - 非埋め込み
    - フォントを埋め込まない
    - ファイルサイズは最小
    - 正しく表示できないことがある

# LilyPond



# LilyPond

- ソースファイルをコンパイル

```
\relative  
{  
  \clef treble  
  \key e \major  
  \time 4/4  
  \tempo "Allegro"  
  
  \partial 8 e''8 |  
  gis gis gis fis16 e b'4. b16 a |  
}
```

- 楽譜のPDFなど生成



# Texinfo

# Texinfo

- GNU 公式文書フォーマット
  - LilyPond のマニュアルでも利用
- Texinfo 形式のファイルから各種形式のドキュメントを出力できる
  - HTML など
    - スクリプトで変換
  - PDF
    - plain TeX が使われる
- 図 PDF を貼り付けると
  - ワークフローは  $\text{\LaTeX}$  と同じ
  - 本発表では区別しません

# 例（冒頭部分）

```
\input texinfo-ja.tex

@documentencoding UTF-8
@documentlanguage ja

@settitle 吾輩は猫である
@afourpaper

@titlepage

@title 吾輩は猫である
@author 夏目漱石

これは日本語Texinfoファイルのサンプルとし
...
```

- 1行目：plain T<sub>E</sub>X用マクロを読み込む
- 2行目以降：@コマンドでマークアップ

# T<sub>E</sub>X で図を貼り込む

# 図のフォント

- 普通に PDF を作ると、、、
  - フォントがサブセット埋め込みされる

name	type	encoding	emb	sub	uni	object	ID
AUVQXI+Emmentaler-20	Type 1C	Custom	yes	yes	no	8	0
RDUTAJ+TeXGyreSchola-Bold	Type 1C	WinAnsi	yes	yes	no	10	0

- 図として T<sub>E</sub>X へ取り込むと、、、
  - メイン PDF にすべて埋め込まれる
- 複数の図で同じフォントがあると
  - 同じフォントが複数回埋め込まれる
  - 重複は解消されない

# フォント重複の解消

- 解消方法
  - 図 PDF 作成時の埋め込み方法を工夫
  - メイン PDF を Ghostscript で処理
    - 埋め込みを制御
- 2つのアプローチ
  - フルセット埋め込み
  - 非埋め込み

# フルセット埋め込み

- 図 PDF をフルセット埋め込みにする
  - 図 PDF のサイズが増大
  - メイン PDF のサイズも非常に大きくなる
- 同じフォントはすべて全く同じもの
  - 後から重複しているものを取り除く  
「統合」が可能
- 中間ファイルは大きくなるが、最終  
ファイルのサイズを減らすことが可能



# フルセットの問題

- フルセット埋め込み PDF の作成が困難
  - Ghostscript のフルセット埋め込みに問題
  - 最新 9.22 でも正しいフルセットにならない
    - 表面的にはできたように見える
    - 実際にはすべてのグリフを含んでいない

# フルセットの問題

- フォントの統合が困難
  - Ghostscript バージョン別の動作
    - ～9.16 フォント統合可能
    - 9.17～9.21 統合には要オプション
      - dPDFDontUseFontObjectNum
    - 9.22～ オプション廃止、統合不可能
  - gs-devel メーリングリストでは、、
    - 重複削除は意図したものではない
    - 文字化けする可能性があるので廃止

# 非埋め込み

## ● 図 PDF を非埋め込みにする

name	type	encoding	emb	sub	uni	object	ID
TeXGyreSchola-Bold	Type 1	WinAnsi	no	no	no	12	0
Emmentaler-20	Type 1	Custom	no	no	no	8	0
Emmentaler-20	Type 1	Custom	no	no	no	14	0
Emmentaler-20	Type 1	Custom	no	no	no	10	0

- メイン PDF も非埋め込みになる
  - LilyPond の場合、音符もフォントで表現
  - 通常的环境には音符フォントが無い
  - 音符の表示ができない
- 必要なフォントを埋め込む処理をする
  - Ghostscript が使える
  - 埋め込むフォントを渡す必要がある

# フォントの渡し方（基本）

- 基本的には、、、
  - 特定ディレクトリへフォントを置く
  - Ghostscript の設定ファイルを編集する
- 非常に煩雑
- 自動化が困難
  - 使用しているフォントをいちいち登録していく必要がある
- OTC フォントが使えない

# フォントの渡し方（別解）

- Ghostscript はフォントが埋め込まれたファイルを読み込むことができる
- これを利用する
  - フォントリソースのみ含まれた PostScript ファイルを用意する
    - テンポラリファイルでよい
    - OTC フォントも CFF 抽出すればよい
  - PDF とともに入力ファイルとして渡す

# フォントの渡し方 (LilyPond)

- 使用しているフォントのフォントリソースを書き出すオプション  
`-dfont-export-dir`  
がある
- ここで指定したディレクトリを  
Ghostscript へ渡せばよい

# Ghostscript で失われるもの

- メイン PDF を  
Ghostscript で処理すると、、、
  - ページモードが失われる
    - 「しおり」を開くように設定しても  
→開かれない状態に変わる
  - 宛先名が失われる
    - 外部からの場所を指定したリンクが  
→常にドキュメントの先頭へジャンプ
- Extract PDFmark で保持する

# Extract PDFmark



# インストール

- パッケージ (extractpdfmark) から
  - Debian 9 stretch
  - Ubuntu 17.04 Zesty Zapus
  - Cygwinその他にもパッケージ化されてるものあり
- ソースから <https://github.com/trueroad/extractpdfmark>
  - Autotools なので比較的簡単
    - 依存ライブラリが揃っていれば以下で OK

```
$ ./configure  
$ make  
$ make install
```

# pdfmark

- PDF の機能を PostScript で記述
  - 例：「しおり」を開くページモード指定

```
[ /PageMode /UseOutlines /DOCVIEW pdfmark
```

- これを含んだ PostScript を Ghostscript で処理し PDF を生成すると
- 開いたときに「しおり」が出るようになる

# 抽出

- PDF の仕様は公開されている
  - ページモードや宛先名の読み取り可
  - 各種ライブラリあり
- Extract PDFmark の動作
  - Poppler ライブラリで読み取る
  - pdfmark の形式で出力する

# Ghostscriptの入力

- 特徴
  - 一度に複数の入力を扱える
  - 入力ファイル毎に形式を個別判断
    - PostScript
    - PDF
- 以下の両方を入力に与えると、、、
  - pdfmark が記述された PostScript
  - PDF
- 入力 PDF の機能とは関係なく pdfmark が適用された PDF を出力

# 使い方

- フォント非埋め込みのアプローチ
  - フォントリソースを  
fonts/\*.font.ps  
に置く

```
$ extractpdfmark TeX出力.pdf > 抽出pdfmark.ps
$ gs -q -dBATCH -dNOPAUSE -sDEVICE=pdfwrite \
  -sOutputFile=最終.pdf \
  fonts/*.font.ps \
  TeX出力.pdf 抽出pdfmark.ps
```

# 注意事項

- Ghostscript
  - 宛先名の扱い
    - 9.19 まで英数字以外の名前が扱えない (bug 696974)
    - Texinfo はノード名がそのまま宛先名になるので 9.20 以降が必要
  - しおりの項目名
    - 9.21 まで「しおり」の項目名に「作成」が入っていると化ける (bug 698552)
    - 9.22 で修正済

# 注意事項

- LilyPond
  - 非埋め込みアプローチで  
TrueType フォントを使うと、、、  
**欧文フォント** WinAnsi エンコーディング  
外の文字が化ける  
**和文フォント** 全体が文字化けする
  - そのため LilyPond で非埋め込みを指定するオプション-dgs-neverembed-fonts は、あえて TrueType フォントを埋め込む
- 他の作図ソフトでも、同様のことが発生する可能性あり（未確認）

# おわりに



# LilyPondでの効果

- Extract PDFmark 採用前後

LilyPond バージョン	生成に要する ディスク容量	全 PDF 合計	記譜法マニュアル (英語版)
2.19.51 (2016 年 11 月)	4.6 GB	280 MB	36 MB
2.19.52 (2016 年 12 月)	3.4 GB	104 MB	9.9 MB

- 全 PDF 合計サイズが半分以下に
  - うち、特に楽譜の断片（つまり図）が多い  
記譜法マニュアルは 1/3 以下（英語版）に

# コミュニティの動き

- 先月、フォント重複解消に関する大きな動きがあり
  - Ghostscript 9.22rc1 で  
-dPDFDontUseFontObjectNum が廃止
- これに端を発する議論が  
メーリングリストで活発に
  - gs-devel / lilypond-devel

# おわりに

- この動きにより
  - フルセット埋め込みのアプローチは事実上使用不可に
  - LilyPond の関連オプションは今後変更
- 先月の議論で、まさに「ひっくり返った」感じ
- 議論を通じて様々な知見が得られ勉強になった
  - 本発表にも最新の有用な情報を盛り込めた

# 参考

- 本発表関連ファイルを下記にて公開中
  - <https://github.com/trueroad/tr-TeXConf2017>
- 本発表のアブストラクト
  - TeXConf 2017 のページで PDF 入手可
  - PDF ファイルそのものが  
ファイルサイズ削減のサンプル
    - 複数の図を貼り付けた PDF
    - 非埋め込みアプローチで  
ファイルサイズ削減
  - 上記 github にて
    - T<sub>E</sub>X ソースファイルや生成用 Makefile  
図 PDF などの中間ファイルを公開中

# 楽譜の例



