TeXConf 2019 一般講演 発表資料

原ノ味フォントと ToUnicode CMap

細田 真道

http://www.trueroad.jp

2019 年 10 月 12 日

おことわり

- 経緯（のようなもの）
  - 本資料は 2019 年 10 月 12 日に開催予定だった TeXConf 2019 一般講演向けに制作していたものです
  - 本資料を鋭意制作中に台風 19 号の影響で TeXConf2019 が中止となりました
  - 中止により締め切りがなくなってしまった？ため制作が停滞しておりました
  - 遅れましたが 11 月中に公開できればと思い制作を進め
    なんとか 11 月 30 日公開にこぎつけました
- 本資料について
  - 元々は講演時間に合わせて後でスライド数を減らすつもりでしたが、
    講演がなくなってしまったため減らしておりません…
  - 一部に 10 月 12 日以降の状況も含まれております
  - 不十分なところや間違いなどあればご連絡ください
    正誤表や修正版などの公開も検討します
- URL 等
  - 本資料やアブストラクトについて、
    ソースファイルや関連資料を以下にて公開しております
  - `https://github.com/trueroad/tr-TeXConf2019`

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
  - XeTeX / LuaTeX、Unicode、日本語対応等
- 第 10 回日本 OSS 奨励賞受賞
  - LilyPond
- FIT2018 FIT 論文賞受賞
  - 「数式組版」のお世話になりました

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

原ノ町駅

ここ数か月よく行っているんですが
原ノ味フォントとは関係ありません。。。

はじめに

はじめに

- 源ノフォント
  - 源ノ明朝・源ノ角ゴシック
  - オープンソースの Pan-CJK フォント
  - 明朝・ゴシックそれぞれ 7 ウェイト
  - CID-keyed OpenType/CFF フォント

はじめに

- 源ノフォント
  - 源ノ明朝・源ノ角ゴシック
  - オープンソースの Pan-CJK フォント
  - 明朝・ゴシックそれぞれ 7 ウェイト
  - CID-keyed OpenType/CFF フォント
- 日本語 OpenType/CFF フォントとの違い
  - 従来は Adobe-Japan1 (AJ1)
  - 源ノフォントは Adobe-Identity-0 (AI0)

TEXで源ノフォント

- 最新のTEX Live 2019では使用可能
  - 関係各位のご尽力で
    AI0対応、源ノフォント対応が進んだ

TEXで源ノフォント

- 最新のTEX Live 2019では使用可能
  - 関係各位のご尽力で
    AI0対応、源ノフォント対応が進んだ
- ただし、一部で使用困難や問題発生
  - AI0なので…AJ1ではないので…

原ノ味フォント

- 源ノフォントを AJ1 へ組み替え
  - AI0 に起因する問題を解決
    - 各種 TEX エンジン
    - DVI ドライバ
    - Ghostscript

原ノ味フォント

- 源ノフォントを AJ1 へ組み替え
  - AI0 に起因する問題を解決
    - 各種 TEX エンジン
    - DVI ドライバ
    - Ghostscript
- AJ1 と AI0 は何が違うのか？
- AI0 だとどんな問題があるのか？

AJ1 と AI0

フォント

- **いくつの文字**について
  統一されたデザインの
  グリフ（字形）
  を収録したもの
- 様々な形式がある
  - OpenType もその一つ

源ノフォント

- いくつかの文字
  - Pan-CJK フォント
    - CJK 各言語の多数の文字を収録
- 統一されたデザイン
  - 同じフォント内の文字は
    デザインが統一されている
    - 明朝体・ゴシック体、ウェイトなど
- グリフ（字形）
  - 収録した文字のアウトラインデータ
    を持っている
- 形式
  - CID-keyed OpenType/CFF

CID (Character ID)

- 様々な種類の文字 (Character)
  ひとつひとつに付けられた
  ユニークな識別子 (ID)
- 0 から始まる番号
  - 0 番は CID+0、1 番は CID+1 のように表記

CID-keyed

- CID キー方式
  - CID をキーとして
    使いたい文字のグリフを指定する方式
- アプリケーションの動作
  - 使いたい文字を CID 番号で指定
  - そのグリフのアウトラインを取得
  - 表示や印字をする

使いたい文字	CID（AJ1 フォントの場合）
あ	CID+843
漢	CID+1533
☃	CID+8218

使いたい文字と CID の例

文字コード

- 普通のアプリはCIDを知らない
  - **文字コード**を**符号化**（エンコーディング）したもので取り扱っている
- 文字コード
  - JIS X 0208、Unicode など

文字コード規格の例

- JIS X 0208
  - 日本語でよく使われる文字を集め、それらを識別する番号を付与したもの
    - 区と点それぞれ 10 進数や、2 桁の 10 進数 2 つをハイフンでつなぐなどで表記する
  - 符号化
    - ISO-2022-JP（JIS コード）・Shift_JIS・EUC-JP など

文字		JIS X 0208		
		ISO-2022-JP	Shift_JIS	EUC-JP
あ	4 区 2 点 / 04-02	24 22	82 A0	A4 A2
漢	20 区 33 点/ 20-33	34 41	8A BF	B4 C1

文字と符号化の例

文字コード規格の例

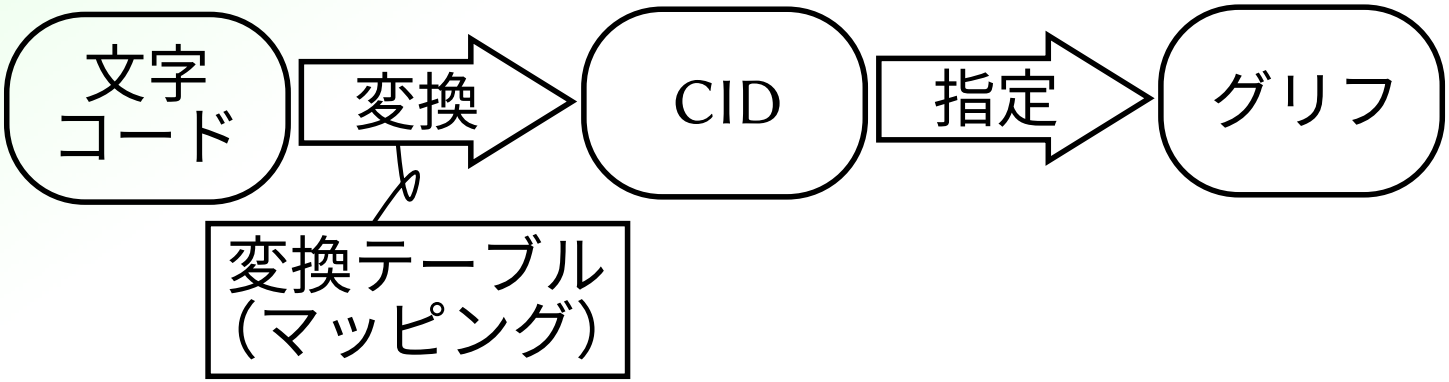
- Unicode
  - JIS X 0208 も含めた世界中の様々な規格にある文字などを集め、それらを識別する番号を付与したもの
    - “U+” の後に 16 進数 4〜6 桁を付け表記
  - 符号化
    - UTF-8・UTF-16・UTF-32 など

文字	Unicode			
		UTF-8	UTF-16	UTF-32
あ	U+3042	E3 81 82	3042	00003042
漢	U+6F22	E6 BC A2	6F22	00006F22
☃	U+2603	E2 98 83	2603	00002603

文字と符号化の例

文字コードとCID

- 全く異なった体系にある
- 変換テーブル（マッピング）を用意して CID に変換する必要がある
- CID を使ってグリフを指定し アウトラインを取得する



文字コレクション

- CID-keyed フォントが持つ情報
  - どのような文字を収録しているかを示す
- ROS とも呼ばれる
  - “Registry”（登録者）
  - “Ordering”（版）
  - “Supplement”（追補）
- これらをハイフンでつないで呼ばれる

	Registry-Ordering-Supplement
AJ1-7	Adobe - Japan1 - 7
AI0	Adobe - Identity - 0

文字コレクションの例

Adobe-Japan1 (AJ1)

- 日本語で使われる様々な文字を集め、識別のためCIDを割り当てたもの
- 最初のAdobe-Japan1-0 (AJ1-0)から最新のAdobe-Japan1-7 (AJ1-7)まである

	制定年	追加されたCID			追加数	合計
AJ1-0	1992	0	〜	8283	8,284	8,284
AJ1-1	1993	8284	〜	8358	75	8,359
AJ1-2	1993	8359	〜	8719	361	8,720
AJ1-3	1998	8720	〜	9353	634	9,354
AJ1-4	2000	9354	〜	15443	6,090	15,444
AJ1-5	2002	15444	〜	20316	4,873	20,317
AJ1-6	2004	20317	〜	23057	2,741	23,058
AJ1-7	2019	23058	〜	23059	2	23,060

AJ1 の例

	Unicode	AJ1
あ	U+3042	CID+843
漢	U+6F22	CID+1533
☃	U+2603	CID+8218
𰻞	U+32FF	CID+23058
𠮟	〃	CID+23059
飴	U+98F4	CID+7634
飴	〃	CID+1151
見	U+898B	CID+1887
見	U+2F92	〃

文字と Unicode、AJ1 CID の例

見た目が異なれば別の CID

- 横書き用と縦書き用
  - Unicode は区別しない

	縦横	Unicode	AJ1
𠮷	横書き	U+32FF	CID+23058
𠮟	縦書き	〃	CID+23059

- JIS2004 字形と JIS90 字形
  - Unicode は区別しない

	字形	Unicode	AJ1
飴	JIS2004	U+98F4	CID+7634
飴	JIS90	〃	CID+1151

見た目が同じなら同じCID

- 普通の漢字と部首
  - Unicode は区別する

	由来	Unicode	AJ1
見	漢字	U+898B	CID+1887
見	部首	U+2F92	〃

変換テーブル（マッピング）

- 文字コードから CID に変換する方法は 2 種類ある
  - CMap リソース
    - pTEX/upTEX + dvipdfmx/dvips などで使用（AJ1 フォントの場合）
    - フォントファイルとは別のファイルを指定
  - cmap テーブル
    - LuaTEX/XETEX などで使用
    - フォントファイルに内蔵されている

CMap リソース

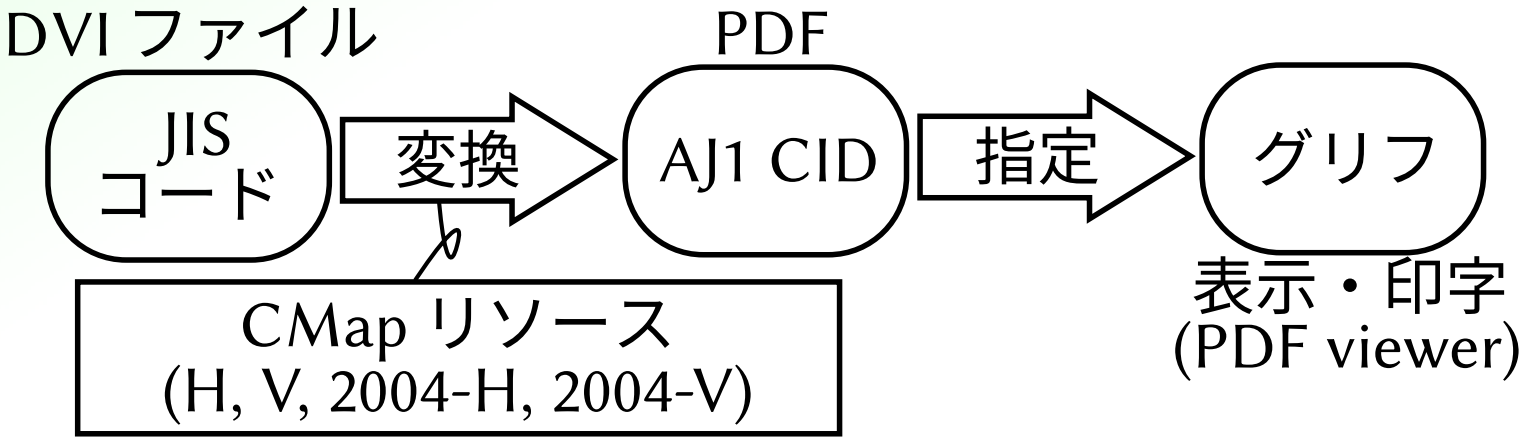
- 入出力の種類別にファイルがある

	入力	出力 (AJ1)	
		字形	縦横
H	ISO-2022-JP (JIS)	JIS90	横書き
V	ISO-2022-JP (JIS)	JIS90	縦書き
2004-H	ISO-2022-JP (JIS)	JIS2004	横書き
2004-V	ISO-2022-JP (JIS)	JIS2004	縦書き
UniJIS-UTF16-H	UTF-16	JIS90	横書き
UniJIS-UTF16-V	UTF-16	JIS90	縦書き
UniJIS2004-UTF16-H	UTF-16	JIS2004	横書き
UniJIS2004-UTF16-V	UTF-16	JIS2004	縦書き

CMap リソースの例

pTEX

- DVI ファイルは JIS コード
- dvipdfmx マップファイルの設定
  - JIS コード用 CMap を
    使用する字形・縦横別に指定する
    - H, V, 2004-H, 2004-V
- dvipdfmx の動作

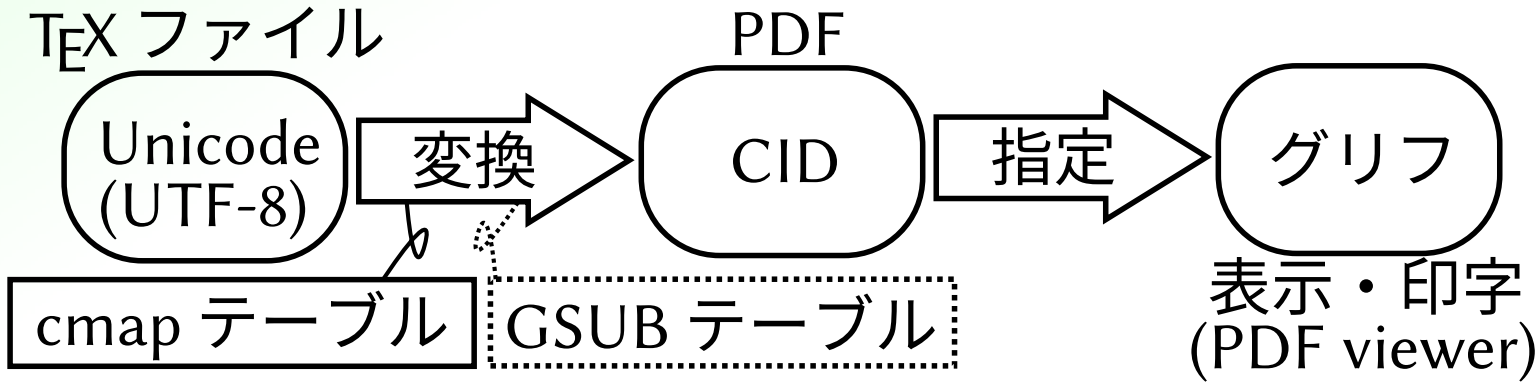


cmapテーブル

- OpenTypeに内蔵されている
- 基本的に1種類しかない
  - 入力：Unicodeのみ
  - 出力：字形固定、横書きのみ
- OpenType featureで
  字形や縦書きを切り替える
  - GSUBテーブル

LuaTEX

- TEX ファイルは Unicode (UTF-8)
- フォントの指定のみ
  - マッピングの指定は無し
  - 必要に応じて OpenType feature の指定
- LuaTEX の動作



Adobe-Identity-0 (AI0)

- AJ1
  - 「日本語のため」の文字コレクション
  - 以下は収録できない
    - AJ1 で定義されていない文字
      （日本語で使われない）
    - 同じ文字の複数バリエーション
  - フォントが違っても同じ CID は同じ文字
- AI0
  - 「特別な目的」の文字コレクション
    - 何も事前に定義されていない
    - フォント制作者が自分で自由に決める
  - フォントが違えば同じ CID でも違う文字

源ノフォント

- 明朝とゴシック間で CID が異なる
- バージョン間で CID が異なる

	Unicode	AJ1	AI0	
			源ノ明朝 1.001	源ノ角ゴシック 2.001
あ	U+3042	CID+843	CID+1461	CID+1461
漢	U+6F22	CID+1533	CID+24743	CID+24227
☃	U+2603	CID+8218	CID+1274	CID+1281
𠮷	U+32FF	CID+23058	—	CID+2184
𠮟	〃	CID+23059	—	CID+65359
飴	U+98F4	CID+7634	CID+45263	CID+44358
飴	〃	CID+1151	CID+61214	CID+62049
見	U+898B	CID+1887	CID+38198	CID+37348
見	U+2F92	〃	〃	〃

CMap リソース

- 源ノフォント用は一応存在する
  - 入力：UTF-16, UTF-32 のみ
    - DVI ファイルが JIS コードの pTEX で使えない
  - 出力：JIS2004 字形、横書きのみ
    - 字形切り替え、縦書きができない
- インストール・設定が煩雑
  - 明朝・ゴシックで別ファイルになっている
  - バージョンアップすると
    CMap リソースも入れ替える必要あり
- 事実上「使えない」

cmapテーブル

- AI0 でも AJ1 と同様に使える
  - LuaTEX/XƎTEX
    - 字形切替や縦書きも AJ1 と同程度に利用可
    - 設定で AJ1/AI0 を意識する必要なし
      （cmap テーブルはフォントファイルに内蔵）
- upTEX + dvipdfmx なら利用可能
  - マップファイルで設定可
  - ただし制約あり
    - OTF パッケージは大部分が利用不可
    - AJ1 と文字幅が異なると問題が発生、など

源ノフォントを使う

それでも源ノフォントを使う

- AI0 の問題を回避して使う方法を考える
- ただし、いずれの方法も
  - 設定が煩雑
  - 制約がある
- 注意が必要

pTEXで使うには

- 使う方法は（一応）ある
  - CMap リソースを作成する
  - VF を使う
  - DVI ファイルを書き換える

upTEXで使うには

- dvipdfmxなら使える
- dvipsでは…
  - 源ノフォントのCMapリソースを使う

TEXエンジンを移行する

- pLATEX専用のクラスファイルを何とかする
  - 新規作成する
  - 改変する

原ノ味フォント

源ノフォントの問題

- 一部の環境
  - pTeX
  - upTeX + dvips
  - OTF パッケージ、など
- 利用困難
  - 設定が煩雑
  - 制約がある、など

原ノ味フォント

- 源ノフォントの問題はAI0が原因
  - AJ1なら発生しない…
- AJ1へ組み替えればいいのでは？
- そこで**原ノ味フォント**を制作

原ノ味フォントとは

- 「原ノ味」の意味
  - 源ノフォントから「氵（さんずい）」を取った
    - グリフやテーブルが抜けているので
  - AJ1 をもじる
    - AJI にして音から「味」をあてた
  - 「ノ」はカタカナ
- オープンソース
  - SIL Open Font License
    - 源ノフォントの派生フォントなので

制作の経緯

- ipsj.clsで源ノフォントを使いたい
  - 情報処理学会 LaTeX スタイルファイル
  - pLaTeX 専用
- しかし、様々な問題が発生、一筋縄ではいかない

制作の経緯

- 源ノフォントはオープンソース
  - 一定の条件下で改変や
    再配布が認められている
- 源ノフォントのグリフを使った
  AJ1 フォントが作れるのでは？
- ということで制作を開始

搭載グリフ

- 漢字
  - AJ1-6 漢字グリフすべて搭載
    - ルビ用の「注」は非搭載
      （AJ1 漢字グリフ範囲外）
  - JIS X 0208 漢字グリフすべて搭載
  - JIS X 0213 漢字グリフすべて搭載

搭載グリフ

- 非漢字
  （ひらがな、カタカナ、英数字、記号類）
  - JIS X 0208 横書きグリフすべて搭載
  - JIS X 0208 縦書きグリフは
    以下 4 グリフを除きすべて搭載
    - 「‖」「° 」「′ 」「″ 」
- JIS X 0213 非漢字グリフには抜けあり
- その他 AJ1-6 非漢字グリフには抜けあり

文字幅

- 非漢字の一部に文字幅を強制的にAJ1へ合わせたものがある
  - ギリシャ文字、キリル文字、一部の記号類
- 不格好な表示になることがある
  - 文字幅は正しいので後続文字の位置がズレるような問題は発生しない

生成プログラム

- オープンソースで公開している
  - 自分で原ノ味フォントを生成できる
  - カスタマイズも可能

動作

- 源ノフォントの OTF ファイルを fonttools の ttx で XML にする
- この XML や CMap リソースなどから CID の対照表を作る
- 対照表に基づいて XML の CID を変換
- 最後に再び ttx で OTF ファイルを生成

CIDの対応

- 源ノフォントはAI0フォント
  - CIDの並び方がAJ1とは異なる
- AJ1化するには

  AI0 CID → AJ1 CID

  対照表が必要
- 自動で対照表を作成する
  - グリフ数が多く人手では困難
    - 日本語用でも2万近い数

pTEX・upTEXで使う

- 源ノフォントと異なり簡単に使える
  - フォントファイルとマップファイルを適切に配置
    - マップファイルも原ノ味フォントのサイトで配布している
  - kanji-config-updmap コマンドで設定

pTEX・upTEXで使う

- 注意点
  - 非搭載グリフは使えない
    - □に×を重ねたようなダミーグリフが出る
  - 文字幅の問題で一部に表示が不格好なグリフがある
    - 源ノフォントのような後続文字の位置がズレる問題は無し

その他の環境で使う

- LuaTEX、XETEX、LilyPond
  - 他のフォントと同様に利用可能
  - IVS や OpenType feature も使える

ToUnicode CMap

ToUnicode CMap

- PDF 中の文字が CID で指定
  - Unicode などは失われている
- テキスト抽出には
  CID → Unicode
  というテーブルが必要
- これが ToUnicode CMap

テキスト抽出

- PDF からテキスト抽出
  - PDF viewr からのコピー＆ペーストなど
- AJ1 フォント
  - PDF に ToUnicode CMap が無くても OK
  - PDF viewer が持っていればよい
- AI0 フォント
  - PDF に ToUnicode CMap が必要

ToUnicode CMapの自動生成

- dvipdfmx (pTEX/upTEX)
  xdvipdfmx (XETEX)
  - AJ1 フォント：自動生成・埋め込みせず
  - AI0 フォント：自動生成・埋め込みする
- LuaTEX
  - 全フォント：自動生成・埋め込みする

逆変換による自動生成

- 生成方法
  - PDF で使用する CID で cmap テーブルを参照
  - この逆変換で ToUnicode CMap を生成

逆変換による自動生成

- 問題1（いわゆる部首問題）
  - CID+1887 は U+898B？ それとも U+2F92？

	由来	Unicode	AJ1
見	漢字	U+898B	CID+1887
見	部首	U+2F92	〃

- どちらがよりふさわしいか
  - 個々の CID によって異なる
  - 現状は「部首」のブロックを決めて選択されないようにして回避している

逆変換による自動生成

- 問題 2（非デフォルト CID が抜ける）
  - CID+23059（縦書き）
  - CID+1151（JIS90 字形）
    - JIS2004 フォントの場合

	種類	Unicode	AJ1	
舘	横書き	U+32FF	CID+23058	←○
舘	縦書き	〃	CID+23059	←×
飴	JIS2004 字形	U+98F4	CID+7634	←○
飴	JIS90 字形	〃	CID+1151	←×

- cmap テーブルに載っていない
  →逆変換不可→テキスト抽出できない

逆変換による自動生成

- 問題 3（非デフォルト CID が化ける）
  - cmap テーブル
    - 開き括弧「(」U+FF08 → CID+674
  - GSUB テーブル vert（縦書き）
    - CID+674 → CID+7899「﹇」
  - 逆変換生成した ToUnicode CMap
    - CID+7899 → U+FE35
- 本来 U+FF08 が抽出されるべきだが U+FE35 になる

調整済の ToUnicode CMap

- 調整済の AJ1 用 ToUnicode CMap がある
  - 問題 1〜3 が起きにくいよう配慮されている
- PDF viewr がこれを持っている
  - AJ1 なら PDF に ToUnicode CMap が埋め込まれてなくてもテキスト抽出できる
  - PDF のファイルサイズが小さくて済む
  - 問題 1〜3 が起きにくい

PDF/A-2u

- PDF/A
  - 遠い将来の PDF viewer でも
    利用できるよう配慮した PDF の規格
  - 通常の PDF 規格をベースに
    守るべき事項が追加されている
- PDF/A-2u
  - PDF/A の一つ
  - テキスト抽出（Unicode 取得）できる
    ことが必須

PDF/A-2u

- PDF/A 作成
  - 本件公開アブストラクトの PDF が PDF/A-2u 準拠
    - ※現在 TeXConf 2019 公式ページにある PDF は非準拠
  - GitHub にて TEX ソース公開中
- バリデーション
  - veraPDF でバリデーションチェック可能
  - 公開アブストラクト PDF（GitHub 版）は veraPDF 1.15.8 で準拠の確認済

テキスト抽出の条件

- veraPDF ドキュメントより
  - 基本的には ToUnicode CMap が必要
  - 例外的に無くてもよい条件がある
  - そのうちの一つが AJ1 の場合
- AJ1 なら PDF に ToUnicode CMap が **無くてもよい**
  - 無くても PDF/A-2u 準拠できる
    →テキスト抽出できる
- AI0 にはないメリット

LuaTeX と ToUnicode CMap

LuaTeXとToUnicode CMap

- dvipdfmx/xdvipdfmx
  - AJ1 フォントなら ToUnicode CMap を生成せず、埋め込まない
- LuaTeX
  - AJ1 フォントでも ToUnicode CMap を生成して、埋め込む
  - ROS を Al0 へ書き換えてしまう
  - つまり、AJ1 フォントでも逆変換問題が発生してしまう

ToUnicode CMap削除

- AI0 に書き換えても CID は変わらない
- PDF の ROS を AJ1 に戻し、
  ToUnicode CMap を削除すると…
- 逆変換の問題を解消することができた
  - PDF のファイルサイズが小さくなる
  - 前節の問題 1〜3 が起きにくくなる
  - AJ1 のメリットが得られる

削除ツール

- 自動的に ToUnicode CMap 削除ができるツール pdf-rm-tuc を作成
  - `https://github.com/trueroad/pdf-rm-tuc`
- 公開アブストラクト PDF（GitHub 版）は本ツールで削除済
  - veraPDF 1.15.8 で PDF/A-2u 準拠の確認済

おわりに

おわりに

- AJ1 と AI0 の違い
- 源ノフォントと原ノ味フォント
- ToUnicode CMap と PDF/A-2u
- LuaTEX 向け ToUnicode CMap 削除ツール

原ノ味フォントの今後

- 抜けているグリフを埋める？
  - 源ノフォントに存在しないグリフ
  - 使用頻度は低そうだが…
    - 和文フォントから使うことは
      あまりないのでは？？？
- 埋めるには？
  - グリフの「変形」ができるか？
    - かなりの実装量が必要
  - ルビ用などはコピーした方がよいか？
    - 全く同じ形状のグリフになるが
      ないよりはマシか？

その他の話題

- PDF/A の解説は少ない？
  - 印刷用の PDF/X はたくさんありそう
  - 本資料は PDF/A-2u 準拠
  - 本資料が助けになれば幸い
- PDF/A のまま電子署名するツール
  - 見つからなかったので実験的に作成

    https://gist.github.com/trueroad/0b0a2127aff508caf583265fbef4b644
  - クライアント証明書をどうするか
    - 有償のものが多い
    - CAcert は無償だが筆者は保証ポイントが無く名前を入れられない

関連資料

- 本資料やアブストラクトについて、ソースファイルや関連資料を公開中
  - https://github.com/trueroad/tr-TeXConf2019
- 不十分なところや間違いなどあればご連絡ください
  正誤表や修正版の公開も検討します