

Nuty Drogi Neokatechumenalnej

— Jak skompilować nuty?

Andrzej Stanisław Odyniec

30 października 2022

Dokument ten piszę
głównie z myślą o sobie, bo
choć sam kompilowałem nuty
wiele razy, przedtem sam je wyrzeźbiłem
no i wreszcie sam napisałem cały zestaw
tajemniczych makr do automatyzacji składu,
to pamięć ludzka jest zawodna na tyle, że gdy po
dziesięciu latach kolejny raz brałem się za pracę nad
tym tematem, bardzo wiele rzeczy musiałem sobie
z niemalym wysiłkiem przypominać. Jednak gdyby
ktoś posługując się poniższymi notatkami skompilował
ten dokument albo sam go sobie modyfikował,
ucieszyłoby to mnie niezmiernie. Tutaj należy
wyraźnie powiedzieć, że poza wszelkimi trudnościami
z uruchomieniem środowiska składu \TeX owego
dochodzą jeszcze niemałe kłopoty związane ze
składem nut, który jest zupełnie dodatkową
własnością tego systemu i wymaga
nie lada wspomagania. Za to
jest bardzo stabilny i daje
dość ładne rezultaty.

Spis treści

1	Struktura dokumentu	3
1.1	Obsługa akordów	3
1.1.1	Makrodefinicje <code>upflag</code> oraz <code>upbox</code>	4
1.1.2	Masowe definiowanie akordów <code>chorddef</code> oraz <code>chmajfamdef</code>	5
1.1.3	Definiowanie obrazków <code>chp</code>	6
1.2	Obsługa struktury śpiewnika	7
1.3	Dokument główny	8
2	Preprocessing	9
2.1	Język M-Tx	9
2.2	Poszczególne preprocesory	11
2.2.1	Generowanie plików midi	13
2.3	Makra do składu tekstu pieśni	13
3	Warsztat oraz przebiegi kompilacji	14
3.1	Platforma MiKTeX	15
3.2	Sortowanie alfabetycznego spisu treści	15
3.3	Budowa nutek	15
3.4	Budowa całości	16
3.5	Edycyjne Środowisko zintegrowane	16

1 Struktura dokumentu

Początkiem jest plik `liturg.tex` (dla nut śpiewnikowych `spiewnik.tex`), który ma klasyczną strukturę L^AT_EXową. Należy pamiętać, że dokument jest mocno spasowany i wszelkie zmiany totalne (w prologu dokumentu) mogą mieć tragiczny wpływ na wiele stron. Każda sekcja w prologu powinna być opatrzona krótkim komentarzem, np.:

```
% pakiety do fontów i~matematyki
% (akordy składane są w~trybie matematycznym)
\usepackage{fontspec}
\usepackage{polyglossia}
\setmainlanguage{polish}
\usepackage{unicode-math}
```

Powyższa sekcja jest związana z przejściem na unikody tak na wejściu jak i w produkowanym PDFie, co wymaga odpowiednich mechanizmów tak przy selekcji fontów jak i przy obsłudze matematyki. Dalej znajdziemy obsługę fontów, grafiki, identyfikacji tak wersji jak i użytkownika (w wypadku indywidualnych transpozycji niektórych pieśni), ustawień w produkowanym PDFie (jak np. strona startowa) aż do:

```
% pakiety do obsługi akordów oraz struktury śpiewnika
\usepackage{chords,spiewnik}
```

czyli włączenia obsługi systemu składu akordów (`chords.sty`) oraz struktury śpiewnika (`spiewnik.sty`).

1.1 Obsługa akordów

Obsługa akordów jest zdefiniowana w stylu `chords.sty`, który można wywołać z jednym z następujących parametrów definiujących proporcje długości do szerokości w ewentualnych piktogramach akordów:

- short** piktogramy akordów będą składane w postaci krótkiej, po dwie odległości międzystrunowe pomiędzy progami, czyli dla układów trójpozycyjnych w proporcji 6×5 a dla układów czteropozycyjnych w proporcji 8×5;
- medium (domyślne)** piktogramy akordów będą nieco dłuższe, po trzy odległości międzystrunowe pomiędzy progami, czyli dla układów trójpozycyjnych w proporcji 9×5 a dla układów czteropozycyjnych w proporcji 12×5;
- long** piktogramy akordów będą składane w proporcjach wymiarowych jak na gryfie gitary, po cztery odległości międzystrunowe pomiędzy progami, czyli dla układów trójpozycyjnych w proporcji 12×5 a dla układów czteropozycyjnych w proporcji 16×5.

Składanie symboli akordów nad tekstem opiera się na przywoływaniu akordu komendą zawierającą w nazwie:

1. literę akordu (wielkie litery [FCGDAEH] oznaczają akordy durowe a małe [fcgdaeh] — akordy molowe),
2. podwyższenie [is] albo obniżenie [es, s] (obsługiwane są też wyjątki, jak [es, as oraz b=hes]),
3. rodzaj akordu i jego alteracje z ewentualnymi wariantami (np. [S] oznacza akord dominantowy septymowy).

Literowy opis tonacji z numeru kwinty generuje makro `\cn`. Jedyne parametry musi być liczbą całkowitą w zakresie 1 ... 35 lub -1 ... -35 tak, że tonacja C-dur odpowiada liczbie 16 a tonacja c-moll liczbie -16. Kolejne liczby odpowiadają kolejnym tonacjom w kole kwintowym, przy czym kolejność określa moduł (wartość bezwzględna) argumentu. Tak więc wartości bezwzględne poniżej 16 odpowiadają tonacjom bemolowym a powyżej 16 — krzyżykowym. Na przykład -14 oznacza g-moll a +18 D-dur.

Komendy stawiające akordy przywołują akord wg numeru kwinty na kole kwintowym. Ten numer będzie przed wygenerowaniem literowego opisu tonu ewentualnie zmodyfikowany poprzez dodanie licznika `\transposeQ` tak więc lokalne modyfikacje tego licznika dokonują automatycznej transpozycji tonacji na kole kwintowym, gdyż wszystkie akordy będą tworzone wg przesuniętych numerów kwint na kole.

Tutaj warto wspomnieć, że do stawiania akordów nad pięciolinia służą komendy zaczynające się dodatkowo od prefiksu `\scr` (od ang. *scores* = nuty). Komendy te należy przywoływać w trybie matematyki. Aby skład akordów był wygodniejszy, w prologu dokumentu z nutami zredefiniowana jest komenda `\upflag` (patrz rozdział następny), odpowiedzialna za wysunięcie akordu nad tekst i włączenie matematyki przy notacji typu `\Fis` tak, że w plikach z nutami można używać klasycznej notacji akordowej ale wtedy trzeba pilnować, aby nuty były składane wewnątrz grupy, gdzie `\upflag` była zredefiniowana i aby ta redefinicja podczas składania tekstu po nutach (dotyczy nut liturgicznych) nie była już aktywna. Jeśli z jakichś powodów nie możemy o to zadbać, należy akordy nad nutami zapisywać jako `$_scrFis$`.

1.1.1 Makrodefinicje `upflag` oraz `upbox`

Jedną z metod stawiania akordu jest wysunięcie go nad wiersz z miejsca, w którym użyto komendy, Odpowiedzialne są za to następujące makra:

```
\def\chordskip{1.6ex}
\def\chordup{1.9ex}
\def\upflag#1{\raise\chordskip\hbox{\strut}%
\smash{\rlap{\raise\chordup\hbox{\red$#1$}}}}
\def\upbox#1{\raise\chordskip\hbox{\strut}%
\smash{\raise\chordup\hbox{\red$#1$}}}
```

Makro `\upbox{}` umieści opis akordu w pudełku o zerowych wymiarach pionowych podniesionym ponad linię bazową o `\chordup{}` i poprzedzając ją wcześniej `\strutem` (czyli pustym znakiem o wymiarach pionowych nawiasu ale o szerokości zerowej) jednak podniesionym do góry na `\chordskip{}`. Tak więc o ile `\chordup{}` określa wysokość stawiania akordów nad linię bazową to `\chordskip{}` dba o odepchnięcie od linii powyżej.

Parametrami tymi można sterować położeniem akordów nad wierszem poniżej (`\chordup`) oraz odstępem akordów od wiersza powyżej (`\chordskip`). Jednak zmieniając ich domyślne ustawienia należy całą pieśń, wraz z nagłówkiem i treścią zamknąć w grupie (wziąć w dodatkową parę nawiasów klamrowych) aby zmienione definicje dotyczyły tylko tej pieśni a następnych już nie.

Makro `\upflag{}` także umieszcza akord nad wierszem ale dodatkowo w pudełku zerowej szerokości z materiałem wysuniętym w prawo (czyli stawia taką flagę, chorągiewkę z akordem, skierowaną w prawo). Nie zajmuje to przestrzeni pomiędzy literami poza minimalnym zakłóceniem naturalnych podcięć (*kerningu*), które by między literami wystąpiły, gdyby flagi z akordem nie postawiono. Komenda stawiająca akord używa domyślnie makra `\upflag{}`.

Czasami dla utrzymania rytmu pionowego zachodzi potrzeba *odepchnięcia się* od poprzedniego wiersza tak, jakby w bieżącym wierszu był postawiony akord, chociaż akurat go nie ma. Do tego celu służy *jałowy* akord, który jest pusty ale zachowuje pionowe odstępy: `\I` wykorzystująca `\upflag{}`.

1.1.2 Masowe definiowanie akordów `chorddef` oraz `chmajfamdef`

Komendy stawiające akordy robi (raczej wyłącznie wewnątrz `chords.sty`) makro `\chorddef{ }{ }`, definiujące całą rodzinę makr przywołujących opis akordu tak, że pierwszy argument jest liczbą przekazywaną do makra `\cn`, drugi jest przyrostkiem do generowanej makrodefinicji a trzeci — przyrostkiem do generowanego przez tę rodzinę opisu. Np. dla argumentów `{22}{N}{~9}` wygenerowane zostaną makra dla akordów nonowych: `\FisN` — przywołujące „chorągiewkę” nad tekstem, `\tFisN` — przywołujące tekst do włożenia w `\upflag` lub `\upbox`, `\ddFisN` — przywołujące sam opis akordu bez obrazka i `\lyFisN` — przywołujące obrazek akordu o ile jest zdefiniowana komenda rysująca ten obrazek.

Jest wiele zwyczajów opisu akordu. Definiując akordy, ich grupy i rodziny należy wybrać jakąś jedną metodę, w miarę jednoznaczną, szczególnie, gdy chodzi o akordy nietypowe, alterowane.¹

¹Zwykle przyjmuje się, że jeśli nie postawiono indeksu górnego, to akord jest trójdźwiękiem, czyli jego najwyższym stopniem jest kwinta. Jeśli jednak potrzebujemy dalszych stopni w akordzie, to stawiamy je w indeksie górnym tak, że X^7 oznacza akord-czterodźwięk z dodaną septymą w odległościach pomiędzy stopniami takimi jak w przypadku akordu dominantowego. Jeśli dodamy kolejną tercję do akordu, uzyskamy pięciodźwięk — akord

Ponieważ zwykle przyrostek do opisu będzie stawiać indeksy górne i dolne, należy komendy stawiające akordy wywoływać w trybie matematyki. W trybie matematyki używa się specjalnych fontów a symbole tak litery podstawy akordu jak i stopni mogą wymagać przywołania fontu tekstowego, dlatego oznaczenie podstawy akordu zamykamy w komendach `\chordsmaj` oraz `\chordsstop` dla akordów durowych i `\chordsmin` oraz `\chordsstop` dla akordów molowych. Indeksy górne i dolne zamykamy natomiast w pary komend `\chordssub` oraz `\chordsstop`. Komendy te mogą być zdefiniowane w razie zabawy z fontami.

Aby nie definiować grupy akordów dla każdego dźwięku na kole kwintowym, ułatwieniem są komendy definiujące iteracyjnie rodziny akordów dla całego koła kwintowego tak dla akordów durowych `\chmajfamdef{}{}{}` oraz molowych `\chminfamdef{}{}{}` otrzymujące tylko dwa ostatnie argumenty komendy `\chorddef{}{}{}`.

W tym trybie pracy definicja wszystkich akordów septymowych dominantowych wyraża się komendą `\chmajfamdef{S}{}\chordssub7\chordsstop}`. Warto tutaj zauważyć że przyrostek komendy może być tylko literą. Cyfry w \TeX u kończą komendę i do niej nie należą. Dlatego nie mogliśmy zrobić dla akordów septymowych komend postaci `\X7` ale raczej `\XS`.

1.1.3 Definiowanie obrazków chp

Dość skomplikowane jest makro `\chp{}` generujące obrazek akordu. Wywołanie ma format: `\chp{p1,p2,p3,...,pn}`, gdzie p_i jest ciągiem trzyznakowym postaci: *osp* lub dwuznakowym postaci: *sp*.

Znak *s* oznacza strunę i jest jedną z liter: ‘e’, ‘h’, ‘g’, ‘d’, ‘a’, ‘E’ odpowiadających strunom E1, H2, G3, D4, A5 i E6. Znak *p* jest cyfrą w zakresie 1–9 i oznacza pozycję (pole między progami), na której należy umieścić palec. Pozycja liczona jest względem początku wycinka chwytни (gryfu). Znak *o* oznacza sposób zaznaczenia umieszczenia palca i może być literą: ‘F’ (generuje czarną kropkę) oznaczającą obowiązkowe umieszczenie palca, ‘f’ (generuje okrąg) — oznacza dodatkowe ułożenie palca albo literą ‘u’ (generuje dwie

nonowy X^9 . Można dodać jeszcze jedną tercję do akordu aby otrzymać sześciódźwięk undecymowy X^{11} . Formalnie więc górny indeks ma specyfikować najwyższy stopień akordu z założeniem, że poniżej tego stopnia są obecne wszystkie tercje. Trudno jest zagrać wszystkie sześć dźwięków, toteż niektóre się pomija, co zaznacza się zwykle w indeksie dolnym ze znakiem minus, np. X_{7-}^9 oznacza akord nonowy ale bez septymy co można by także zapisać jako X_{2+} czyli trójdźwięk z dodaną sekundą (nona w przewrocie jest sekundą). Jednak taki zapis nie jest jednoznaczny, gdyż dodanie interwału poniżej najwyższego stopnia akordu może być rozumiane jako zastąpienie innego dźwięku — tutaj mogłoby to być rozumiane jako trójdźwięk z dodaną sekundą zamiast tercji. Jednak zapis formalnie purystyczny często jest tak skomplikowany, że ludzie preferują uproszczenia, np. wolą X^4 zamiast X_{7-}^{11} co miałoby oznaczać trójdźwięk z dodaną kwartą zamiast akordu undecymowego bez septymy i nony. Jeżeli już, to akord ten powinien być oznaczony jako X_{4+} .

pionowe kreski) co oznacza tłumienie nieużywanej struny. Brak znaku *o* jest równoważne literze ‘F’.

Pierwszy parametr może na pozycji *s* mieć literę ‘P’, ‘p’, ‘B’ lub ‘b’, wtedy wybiera on sposób oznakowania wycinka chwytни. I tak wielka litera oznacza wycinek czteropolowy a mała litera — trzypolowy. Litera ‘B’ lub ‘b’ oznacza chwyt barrè — wtedy pierwszy próg zostanie poprzedzony dodatkową kreską. W takim parametrze cyfra na pozycji *p* oznacza numer pierwszej pozycji wycinka chwytни. Brak takiego parametru jest równoważny umieszczeniu go w postaci ‘b1’.

Popularny akord a-moll złożymy wywołaniem: `\chp{h1,g2,d2}`. Akord C-dur złożylibyśmy rozkazem: `\chp{h1,d2,a3,fE3}`, akord d-mol rozkazem: `\chp{e1,g2,h3,uE1}` natomiast akord c-moll: `\chp{b4,h1,g2,d2}`. Nie ukrywam, że zmienna liczba parametrów makra `\chp` skomplikowała nieco jego kod i tak trudny. Dodam więc tylko, że obrazek składany jest przy pomocy klasycznego środowiska semigraficznego L^AT_EXa o nazwie `picture`.

Pozostaje mieć nadzieję, że po powyższych uwagach kod `chords.sty` będzie w miarę czytelny.

1.2 Obsługa struktury śpiewnika

Ponieważ pieśń, będąc niejako rozdziałem publikacji nie jest jednak typowym rozdziałem, nie numeruje się go, w tytule poza nazwą umieszcza się inne informacje a na dodatek wymaga nieco innych metod indeksowania, do pisania pieśni stworzyłem odrębne środowiska L^AT_EXowe. Wszystkie te rzeczy są zdefiniowane w pliku `spiewnik.sty`. Używanie tych makr opiszę dalej. Tutaj tylko wspomnę o zawartości.

W pliku z tym stylem znajdziemy:

- zestaw deklaracji i makr do znakowania kolorem kartek
- oznaczenia wykonawcy fragmentu (K, W, D, P, ...)
- obsługę zawilości przypisów dolnych w pieśniach
- obsługę nagłówka i treści pieśni
- prawy nawias klamrowy do powtórzeń
- mechanizm deklarowania tonacji i budowania oznaczniaka tonacji w nagłówku
- mechanizm deklaracji autora i budowania indeksu źródłowego
- mechanizm deklaracji adresu (biblijnego, tematycznego) i budowania indeksu analitycznego
- mechanizm deklarowania alternatywnych tytułów i budowania indeksu alfabetycznego

Styl `spiewnik.sty` można wywołać z następującymi parametrami:

`nocolorbar` podanie tego parametru wyłączy rysowanie w pozycji dolnej paginy barwnego paska odpowiadającego kolorowi kartek z oryginalnego śpiewnika włosko-polskiego. Zamiast tego kolor kartki będzie wypełniał

symbol roku danej pieśni. Kolor tak pokazany jest wprawdzie słabo widoczny ale nadal dostarcza informacji o kolorze kartki. Takie rozwiązanie może być korzystne w wypadku drukowania na drukarce atramentowej w celu oszczędzenia drogich atramentów barwnych.

`nocolorcode` parametr ten ma znaczenie, o ile podano w wywołaniu pakietu także parametr `nocolorbar`. Wyłącza on całkowicie wypełnianie kolorem kartek nawet wnętrza symbolu roku pieśni. Takie rozwiązanie może być przydatne w wypadku drukowania śpiewnika na kolorowym papierze dobieranym barwą do grup stron.

1.3 Dokument główny

Dokument główny zawiera przywołanie plików z poszczególnymi pieśniami oraz pod nutami teksty pieśni jak w śpiewniku. Inaczej, niż w Śpiewniku, nie potrzeba stawiać nagłówków pieśni, gdyż tutaj nie tworzone są indeksy a tytuły pieśni do Spisu treści brane są z nagłówków nut. Na przykład:

```
\begin{music}
\includescores{Anafora}
\end{music}
\vfil
\begin{songbody}
\P P\G an z w\ a ami.
\W \G I z duchem Tw\ a oim.
\P \F W górę s\E erca.
\W Wzn\F osimy je do P\E ana.
\P Dzi\ES ęki składajmy Panu Bogu nasz\ a emu.
\W Godne t\G o i sprawiedl\ a iwe, Godne t\G o i sprawiedl\ a iwe.
\P Z\ a aprawdę g\ d odne to i sprawiedl\ a iwe, s\ E uszne i zbawi\ a enne
    składać Tobie dzięki z\ d awsze i na każdym mi\ a ejscu,
    wsz\ E echmogący i miłośnierny B\ a oże,
    kt\F óry w sposób cudowny stworzyłeś człowiei\ a eka
    \F i jeszcze cudowniej go odkup\ E iżeś;
    T\ a y nie opuszczasz grzeszn\G ika,
\split song
    ale go szukasz z ojcowską mił\ a ością.
    Ty w męce Twego S\G yna zwyciężyłeś grzech i śmi\ a erć,
    a w Jego zmartwychwst\G aniu przywróciłeś nam ż\ a ycie.
    Ty wlałeś do naszych serc Ducha Święt\G ego,
    by uczynić nas Twoimi synami i dziedzic\ a ami.
    Ty nas zawsze odn\G awiasz przez sakramenty zbawi\ a enia.
\PW \ a Abyśmy, uwolnieni z niewoli grz\G echu,
    zostali przemienieni\F eni na podobieństwo Twojego S\E yna.
\P B\ a ogosławimy, uwielbiamy Cię, P\G anie,
    w jedności z całym Kości\ a ołem,
    za te cuda Twojego miłosi\G erdzia
    śpiew\ a amy, śpiew\F amy, h\G ymn Twojej chw\ a ały.
\end{songbody}
```

Przywoływany w komendzie `\includescores{}` plik `*.ex` jest wygenerowany

automatycznie w procesie przetwarzania wstępnego z odpowiedniego pliku `*.mtx` zawierającego nuty w notacji `abc` z podpisanym tekstem.

Makro `\includescores{}` jest nieco inaczej zdefiniowane w tomie nut liturgicznych a inaczej w tomie nut wspólnotowych. W szczególności w tomie nut wspólnotowych pieczęć daty dla stron, na których składany jest przywoływany kod z nutami, brana jest wprost z daty modyfikacji odpowiedniego pliku `.mtx`. W tym celu pole na pieczęć daty dla tego tomu jest nieco poszerzone. W tomie nut liturgicznych rok pieśni jest ustawiany komendą `\Year{}` podobnie jak w śpiewniku tekstowym.

Ze względu na to, że dotknięcie pliku `.mtx` zmienia datę, po ewentualnych zmianach nieistotnych dla treści tekstu nutowego (poprawki interpunkcyjne czy zabiegi redakcyjne w nagłówkach tytułowych) należy przywrócić plikowi `.mtx` poprzednią datę modyfikacji.

2 Preprocessing

Zapis nutowy przygotowany jest w logice podsystemu M-Tx Dirka Lauriego. Ten podsystem ma bardzo uproszczony zapis nutowy i dodatkowo jest dobrze przystosowany do podpisywania pod nutami tekstu pieśni. System M-Tx generuje kod w nieco bardziej skomplikowanym podsystemie PMX Dona Simonsa. System ten z kolei generuje kod `TEX`owy w logice zestawu makr `MuSiXTEX`a Daniela Taupina. Kod ten jest bezpośrednio przystosowany do składania surowym `TEX`em — bez `LATEX`a. Aby spać wszystkie nuty w jeden dokument, wygodnie jest posłużyć się `LATEX`em, ale w tym celu należy pousuwać z wygenerowanych w powyższym procesie plików `*.tex` przeszkadzające instrukcje (robimy to prostym programem `awk`owym) i uzyskujemy pliki `*.ex`.

2.1 Język M-Tx

Typowy zapis nutowy wyglądałby następująco:

```
Part: Małżeństwo znakiem miłości Bożej
Title: 3. Prefacja na mszę za nowożeńców
%MIDIFILE: Prefacja-na-msze-za-nowozencow-3
Composer: odmawia się w-Mszy za nowożeńców
Poet: Prefacja 81.
Meter: m11/8/0/0
Space: 3 10 20
PMX: w185m
Systems: 15
Size: 20
Style: Singer
%\input mszalmtx \

%%\\stafftopmarg6\Interligne\staffbotmarg4\Interligne\relax\
```

Tak wygląda nagłówek nut. Jest tutaj miejsce na nazwę części większego dzieła (**Part:**) gdzie zapisujemy podtytuł, zazwyczaj jednozdaniowe streszczenie prefacji. Następnie mamy tytuł (**Title:**), gdzie podajemy tytuł, zwykle jest to nazwa prefacji.

Pod klauzulą **Title:** w naszych nutach występuje dodatkowo wiersz komentarza **%MIDIFILE: Prefacja-na-msze-za-nowozencow-3**, gdyż zaczyna się od znaku „%” i tym samym jest ignorowany przez preprocesor języka M-Tx. Jest jednak wyławiany przez skrypty AWK-owe, aby nadać wygenerowanemu plikowi muzycznemu **.mid** bardziej ludzką nazwę, tę tutaj właśnie wyspecyfikowaną. Więcej w rozdziale „Generowanie plików midi”.

Następnie jest linia autorska: autor (**Poet:**) — kompozytor (**Composer:**). W klauzuli **Poet:** podajemy lokalizację pieśni (indeks strony klasycznego śpiewnika albo jakas forma zaadresowania źródła, np. numer prefacji w *Mszale*). W klauzuli **Composer:** podajemy twórcę pieśni albo przeznaczenie śpiewu liturgicznego.

Jeżeli wartością klauzuli **Poet:** jest tekst, zostanie on zamieszczony bez żadnych łączy. Jeżeli jest to sama liczba (być może z kropką w środku), skrypty oraz makra **T_EX**owe spróbują odszukać tradycyjny indeks strony pieśni pochodzący ze śpiewnika tekstowego, z trzeciego parametru komendy **\pageid** i powiązać te nuty z pieśnią tak, aby działały łąca pomiędzy nutami i tekstem pieśni w śpiewniku osadzonym w nutach. Tekst referencyjny będzie zawierał rzeczywisty numer strony w śpiewniku tekstowym (z uwzględnieniem układu śpiewnika osadzonego: klasycznego lub alfabetycznego).

Natomiast łące będzie kierowało do identycznej, jak identyfikator nut (nazwa pliku z nutami pieśni), etykiety pieśni (zadeklarowanej w **\songid** w śpiewniku tekstowym), o ile numer referencyjny był dodatni. Jeżeli na stronie będzie kilka pieśni, łące trafi do tej z właściwym identyfikatorem **songid**. Jeśli liczba jest dodatkowo poprzedzona znakiem „-” (minus), łące do tekstu pieśni będzie kierowało nie poprzez etykietę pieśni ale na rzeczywisty numer strony, na której znajduje się pieśń o podanym numerze tradycyjnym.

Łąca w drugą stronę (ze śpiewnika tekstowego do nut) działają wyłącznie w oparciu o identyfikator pieśni (**\songid**). Znaczy to, że jeśli w nutach mamy kilka wersji zapisu dla tej samej pieśni, i mają one różne identyfikatory pieśni (bo nazwy plików są różne), ta wersja, która ma nazwę pasującą do **songid** w śpiewniku tekstowym powinna mieć raczej w klauzuli **Poet:** dodatni numer tradycyjnej strony z **\pageid** (choć nie jest to konieczne) a pozostałe raczej taki sam, lecz ujemny (i to jest konieczne, gdyż ich nazwy nie zostaną odnalezione wśród etykiet w śpiewniku tekstowym). Podobnie powinny mieć te nuty, które są częścią zapisu pozycji tekstowej i nie są w stanie trafić łącem przez identyfikator pliku → identyfikator pieśni. Muszą trafiać poprzez numer strony (przetłumaczonej z tradycyjnej na rzeczywistą, zależnie od tego, czy trafiają w śpiewnik w układzie klasycznym czy alfabetycznym).

Pozostałe parametry właściwie nie powinny być zmieniane ze względu na potrzebę zgodności wszystkich pieśni ze sobą. Może zajść potrzeba podania metrum (zwykle jest to metrum ukryte pierwszego taktu recytatywu). Może być też potrzebna zmiana liczby pięciolinii (**Systems:**), gdyby potrzebnych było ich więcej lub mniej. Wiersze z podwójnymi znakami %% są przekazywane bezpośrednio do następnych faz i służą ustawieniom wymiarów albo przywołaniu makr dla kompilacji T_EXem (bez L^AT_EXa). Po nagłówku zaczynają się nuty. Każda pięciolinia opisana jest trzema wierszami: górnym zawierającym akordy gitarowe (U:), środkowym zawierającym nuty i symbole muzyczne oraz dolnym (L:) zawierającym tekst.

```
U: ~ \G \a
    m11/8/0/0 \recy{\hskip-.75em}\ r2 g8 a4 a2
L: Pan z_wa-mi.
```

```
U: ~ \G ~ ~ \a
    m13/8/0/0 r2 g8 g g a4 a2
L: I~z-du-chem two-im.
```

```
U: ~ \F ~ \E
    m12/8/0/0 r2 c8 c b4 b2
L: W_gó-rę ser-ca.
```

```
U: ~ \F ~ ~ ~ \E
    m15/8/0/0 r2 c8 c c a~c b4 b2
L: Wzno-si-my je do Pa-na.
```

Każda pozycja w wierszu górnym opiewa na jedną wartość w zapisie (nutę lub pauzę). Znak „~” oznacza brak symbolu akordu nad tą wartością muzyczną. Pozycje w wierszu środkowym to wysokości i wartości poszczególnych nut. Tam, gdzie takt długi zawiera recytatyw, należy dla każdej frazy podać ukryte metrum (zaczyna się od m i zawiera dodatkowo informację o nierytmiczności 0/0). W dolnym wierszu z tekstem sylaby wydzielamy kreskami a słowa łączymy w sylaby podkreśleniem. Sylaby są podpisywane wyłącznie pod poszczególnymi nutami albo ciągami połączonymi łukiem (w zapisie nutowym nawiasy okrągłe) albo połączonymi „płótkami” (w zapisie nutowym nawiasy kwadratowe). Po szczegóły notacji odsyłamy do dokumentacji pakietów.

2.2 Poszczególne preprocesory

Preprocessing wygląda tak:

```
prepmx %1
pmxab %1
awk "$0!~\\/\\eject\\endmuflex\\&&$0!~\\/\\bye\\&&$0!~\\/\\input\\&&
    $0!~\\/\\nopagenumbers/{
    gsub(\"\\\\\\\\\\startmuflex\\\", \"\\");print $0}" %1.tex >%1.ex
```

Program `prepmx` jest preprocesorem M-Tx i tworzy plik `*.pmx`. Następnym krokiem jest wywołanie preprocesora systemu PMX, a mianowicie `pmxab`. Utworzony przezeń plik `*.tex` można skompilować `luaTEX`em ale można też potraktować powyższym skryptem `awk`-owym aby uzyskać plik `*.ex`. Taki plik można już przywołać w `liturg.tex`. Prolog ma zdefiniowane makra, które zadają o spis treści i hiperłącza.

Preprocesory `prepmx` oraz `pmxab` mogą nie być obecne w używanej dystrybucji T_EXa albo ze względu na prowadzone prace mogą zawierać błędy, szczególnie ze względu na zmiany w wersjach stabilnych z 2013 pod kątem procesorów 64-bitowych. W takiej sytuacji należy je ściągnąć z sieci i umieścić wraz z binariami używanej dystrybucji T_EXa aby, będąc na ścieżce (`PATH=`), mogły być odnalezione przez system operacyjny.

Ponieważ, jak wspomniałem, najnowsza wersja nie musi oznaczać najlepszej, może okazać się konieczne poszukanie wersji starszej a działającej. U mnie dobrze działa wersja 0.60d preprocesora `prepmx.exe` natomiast wersja 0.61 z niezrozumiałych powodów niezależnie od ustawień po przetworzeniu nagłówka umieszcza w pliku `pmx` dodatkowo komendę `w190m` wymuszającą szerokość składu nut na 19cm. Należałoby `awkiem` usuwać tę zbędnie dodaną komendę z wygenerowanego pliku `pmx` albo we wszystkich pieśniach pomimo deklaracji szerokości składu w klauzuli nagłówka PMX: `w185m` zapisać ją jeszcze raz przed pierwszą strofą w komentarzu `%` wstrzykiwanym do `pmx'a`. Jeszcze nowsze wersje mają zupełnie przeprogramowaną obsługę pauz wielotaktowych, które w piosenkach występują rzadko (a M-Tx używany jest głównie do zapisu piosenek) natomiast w monodiach Kiko występują dość często. W efekcie nowsze wersje jak na razie nie obsługują w strofach z pauzą wielotaktową tekstu górnego (akord), zmiany metrum przed pauzą oraz kresek taktowych (podwójnych, repetycji itp.). Zanim uda się to poprawić, prościej jest pozostawać przy wersji starszej a działającej.

W najnowszej wersji `pmxab.exe` 2.74 pojawiła się kolejna dolegliwość: przy najdłuższej chyba pieśni *Eli Eli, Eli, lammà sababtani?* wstawia do generowanej komendy `\interstaff{}` wartość `***` co wywraca T_EXa (poprzednio była wyliczana tutaj wartość 0.2 ale przy składzie L^AT_EXem wartość ta jest ustawiona ogólnie w prologu). Ponieważ nuty są ostatecznie składane w L^AT_EXu, wymuszanie podziałów na strony przez preprocesor jest nieporządane. Dlatego deklaruję tylko jedną stronę aby podziały realizowała procedura wyjściowa (ang. *output routine*). Jednak z liczby systemów na stronie PMX jakoś sobie wylicza `interstaff` i w najnowszej wersji mu to nie wychodzi. Wynik mu się nie mieści w szerokości pola i procedury we/wy języka programowania (kiedyś to był Fortran ale obecnie jest to C) wypełnia wyjście gwiazdkami. Dlatego używam i proponuję użycie wersji 2.7 preprocesora `pmxab.exe`.

Wersje wykonywalne preprocesorów można znaleźć najpewniej w archiwum <http://icking-music-archive.org/software>.

2.2.1 Generowanie plików midi

Preprocesor PMX, jeśli został zasilony poleceniem I, potrafi wygenerować plik `midi`. Nie należy się po plikach `midi` spodziewać wiele, gdyż w zapisie nutowym mamy jedynie prymki a na dodatek większość zapisu to recytatyw, w którym wartości nut niekoniecznie odpowiadają tym, które powinien zaśpiewać prezbiter. Przebieg melodii w wygenerowanym pliku `midi` nie uwzględnia także oznaczeń artykulacyjnych (np. `fermat`). Jednak dla słabo czytających nuty przesłuchanie takiego pliku `*.mid` może być pomocne. Pliki `midi` mają nadawane nazwy wpisane w komentarzu nagłówka `*.mtx`, np. w pliku `060bjawi.mtx` jest wpisany komentarz:

```
%MIDIFILE: Prefacja-o-Objawieniu-Panskim
```

który zostanie użyty do nadania wygenerowanemu plikowi `L060bjawi.mid` nazwy `Prefacja-o-Objawieniu-Panskim.mid`.

Pliki `midi` nie tylko są na koniec pakowane do archiwum `zip` ale także są podczas procesu kompilacji całego tomu osadzane w nim jako załączniki. Załączniki można potem wykorzystywać na dwa sposoby:

1. korzystając z hiperłącza po prawej stronie tytułu pieśni (nutka w prostokącie). Hiperłącze to umieszczone jest na warstwie niedrukowalnej, więc wspomniana nutka nie będzie widoczna na wydruku. Hiperłącze to wykonuje w JavaScript akcję `exportDataObject` z uruchomieniem wyeksportowanego załącznika. Jeśli przeglądarka plików `pdf` będzie obsługiwać tak JavaScript jak i wspomnianą akcję, po kliknięciu w nutkę plik `midi` powinien zagrać w domyślnym odtwarzaczu systemowym. Ta funkcjonalność jest dość nowa i obecnie obsługują ją tylko przeglądarki Adobe pod Windows i Linux.
2. korzystając z ręcznego uruchomienia załącznika poprzez otwarcie listy załączników (ikona spinacza biurowego obok zakładek) oraz dwukliknięcia wybranego z listy pliku `midi`. Tę możliwość daje kilka innych przeglądarek, ale nie wszystkie pokazują osadzone załączniki.

W każdej z tych sytuacji dochodzi do wypakowania załącznika do katalogu roboczego i jego uruchomienia a ponieważ czynność jest potencjalnie niebezpieczna (załącznik mógłby być wirusem a system mógłby go uruchomić w inny sposób, niż zakłada rozszerzenie nazwy), trzeba wyrazić zgodę na takie działanie.

2.3 Makra do składu tekstu pieśni

Po przywołanych nutach liturgicznych (nie składamy tekstu w nutach śpiewnikowych) umieszczamy treść pieśni w środowisku `songbody`. Gdyby tekst

nam się nie mieścił, możemy alternatywnie użyć środowiska `songbodye`, które używa nieco mniejszego stopnia pisma (11pt zamiast 12pt) albo `sonbgodyt`, które sięga po pismo w stopniu 10pt.

Wewnątrz tego środowiska obowiązują nieco inne zasady, niż w trybie poziomym składu `TEX`owego, bo jest ono zanurzone w tabeli (dokładniej: środowisko `TEX`owe `\hailgn`), gdzie pierwsza kolumna przeznaczona jest na oznaczenia roli a druga na treść wiersza. Dodatkowo włączony jest mechanizm respektowania znaku końca wiersza, który normalnie jest przez `TEX`a ignorowany.

Wiersz można zacząć jedną z komend oznaczających role (zdefiniowane są one zaraz na początku stylu `spiewnik.sty`). W wierszu można używać komend wstawiających akordy. Należy pamiętać, że każda komenda `TEX`owa „pożera” spacje po niej występujące, chyba że występuje po niej następna komenda albo znak różny od litery (np. pusta grupa `{}`).

Grupę wierszy można wziąć w środowisko `rightbrace` z jednym parametrem. W ten sposób obejmujemy te wiersze klamrą po prawej stronie a pośrodku tej klamry po prawej umiścimy wartość parametru. Środowisko `bis` nie ma parametru ale zdefiniowane jest jak poprzednie lecz z parametrem o wartości $2\times$.

Wszelki materiał pionowy (np. odstępy pionowe, `\medskip`) można wstawiać jako argumenty komendy `\sepf{}`.

Do podziału pieśni na kolumny można użyć komendy `\splitsong`.

3 Warsztat oraz przebiegi kompilacji

Opis będzie dotyczyć systemu operacyjnego Windows, chociaż można wszystko zrobić także w systemie Linux. Szczegółowa instrukcja przygotowania środowiska pod Windows znajduje się w pliku `InstalacjaSrodowiska` (w `TEX`u i PDF-ie), który jest częścią pakietu dystrybucyjnego i odwołuje się do paczki `Environment.zip`, gotowej do pobrania oddzielnie a zawierającej elementy tego środowiska, sprawdzone i działające pod systemami Windows 10 oraz Windows 11.

W niektórych komentarzach do nut podawane są oryginalne, hebrajskie tytuły pieśni. Nazwy te wymagają dostępności fontu z alfabetem hebrajskim. Użyłem tutaj darmowego fontu `Noto Sans Hebrew`. Plik z tym fontem w formacie True Type dołączony jest do źródeł. Należy go zainstalować w systemie czyli plik `NotoSansHebrew-Regular.ttf` wrzucić do katalogu `C:\Windows\Fonts` albo wybrać z menu lokalnego dla tego pliku opcję `Zainstaluj` (prawy przycisk myszy).

3.1 Platforma MiKTeX

Jako platformy \TeX owej używam MiKTeXa <http://miktex.org>.

Przygotowane pliki kompilujemy używając komendy `lualatex liturg.tex` dla nut liturgicznych albo `lualatex spiewnik.tex` dla śpiewnikowych. Jeżeli z jakichś powodów budujemy „od zera”, czyli bez plików ze spisem treści, indeksami itd., to do czasu wypisania plików indeksów, przesortowania ich (patrz rozdział następny) oraz ponownego przekompilowania w celu wypisania poprawnego indeksu liter w indeksie alfabetycznym należy kompilować ignorując błędy. Można to uzyskać, odpowiadając `q` na komunikaty albo pierwsze dwa uruchomienia zrobić komendą `lualatex -interaction=nonstopmode SP.tex`. Dopiero, gdy plik `spiewnik.ias` (patrz następny rozdział) będzie niepusty, można kompilować bez parametru `-interaction=nonstopmode`.

Na potrzeby preprocesingu należy uzyskać pewność, że mamy zainstalowany `awk` np. stąd: <http://gnuwin32.sourceforge.net/packages/gawk.htm>. Niekiedy gotowe skrypty używają `awk`-a do pewnych korekt plików tekstowych. Np. przetworzone nuty do postaci \TeX owej są przygotowane do składania każdej pieśni w postaci odrębnego artykułu \TeX owego. Aby złożyć całość \LaTeX em, należy dokonać podmian pewnych wierszy w postaci \TeX owej (pliki `*.tex`) na postać składającą się w kompletny tom (pliki `*.ex`).

3.2 Sortowanie alfabetycznego spisu treści

Zbiory z nutami nie są wyposażone w zaawansowane indeksy analityczne czy źródłowe, jak to jest ze śpiewnikiem tekstowym ale liczba pieśni w tomie z nutami śpiewnikowymi jest na tyle pokaźna, że odnalezienie konkretnej pieśni może okazać się trudne. Dlatego po spisie treści umieszczony jest ten sam spis ale w porządku alfabetycznym. Pochodzi on ze spisu treści `spiewnik.toc` posortowanego alfabetycznie do pliku `spiewnik.ias`. Pamiętajmy, że sortujemy kodowanie `utf8`. Na potrzeby systemu Windows przygotowany jest w Power-Shellu skrypt `sortix.ps1`, który posługując się skrypcem w AWK-u `sort.awk` wywołanym w środowisku CMD, wykona tę pracę. Tutaj podobnie, jak przy indeksach śpiewników tekstowych, przed sortowaniem przetwarzamy spis treści skrypcem `corr.awk`, ale nie jest on pusty, gdyż ze spisu treści należy usuwać niektóre wiersze, nie mające sensu w alfabetycznym spisie treści:

```
!/vspace/&&!/nopagebreak/&&!/Alfabetyczny spis/&&!/{}chapter}/{
  if($0~/Psalmodia [0-9] /){gsub("Psalmodia ","Psalmodia ");
  print}
```

3.3 Budowa nutek

Musi \TeX jest systemem dwuprzebiegowym i po pierwszym przebiegu składu z nutkami należy wykonać:

```
musixflx.exe liturg.mx1
```

dla nut liturgicznych albo

```
musixflx.exe spiewnik.mx1
```

dla nut śpiewnikowych a następnie powtórzyć skład. Niekiedy może się okazać potrzebnym przed pierwszym przebiegiem usunąć pliki *.mx1 oraz *.mx2.

3.4 Budowa całości

Robi ją automatycznie wsad `compileliturg3pass.bat` dla nut liturgicznych oraz `compilesiewnk3pass.bat` dla nut śpiewnikowych.

Ponieważ podczas budowy nut śpiewnikowych `spiewnik.pdf` nuty liturgiczne `liturg.pdf` jak i śpiewnik kantora `SP.pdf` są osadzane w budowanym pdf-ie, pliki `SP.pdf` w katalogu nadrzędnym oraz `liturg.pdf` w katalogu bieżącym powinny być zbudowane wcześniej. Dzięki temu w tak zbudowanym tomie `spiewnik.pdf` będą możliwe skoki pomiędzy tekstem a nutami.

3.5 Edycyjne Środowisko zintegrowane

Gdyby naszła nas ochota intensywniej popracować na edycją tego, czy innych tekstów w TEX u — $\text{L}\text{A}\text{T}\text{E}\text{X}$ u, można rozbudować warsztat o środowisko edycyjne integrujące się z systemem Windows oraz z $\text{MiK}\text{T}\text{E}\text{X}$ em. Dość dobrze do tych celów nadaje się TEX studio, które można pobrać stąd: <http://texstudio.sourceforge.net>.