Politechnika Wrocławska
Wydział Informatyki i Telekomunikacji

Kierunek: **Informatyka Stosowana (IST)**
Specjalność: **Zastosowania Specjalistycznych Technologii Informatycznych**

PRACA DYPLOMOWA MAGISTERSKA

Opracowanie algorytmu generacji grafu DSP do rozwiązania problemu syntezy dźwięku

Automated generation of signal processing graphs for sound synthesis

Mateusz Bączek

Opiekun pracy
dr inż. Maciej Hojda

Słowa kluczowe: synteza, dźwięk, graf, optymalizacja

WROCŁAW, 2022

Streszczenie

Praca prezentuje metodę automatycznej konstrukcji grafu przetwarzania sygnałów dźwiękowych, które wykonują syntezę zadanego przez użytkownika dźwięku. Wytworzony w ramach pracy algorytm może zostać wykorzystany jako narzędzie w pracy inżynierów dźwięku, podczas tworzenia nowych instrumentów elektronicznych lub efektów specjalnych. W przeciwieństwie do technik wykorzystujących sieci neuronowe jako narzędzia syntezy, wynikiem działania algorytmu wytworzonego w ramach pracy jest zrozumiały dla człowieka graf przetwarzania sygnałów, przypominający konwencjonalne konfiguracje syntezatorów dźwięku wykorzystywane w programach do pracy nad dźwiękiem.

Słowa kluczowe: synteza, dźwięk, graf, optymalizacja

Abstract

TODO!

Keywords: synthesis, sound, audio, graph, optimisation

Spis treści

Spis rysunków

Spis tabel

Spis listingów

Skróty

OGC (ang. *Open Geospatial Consortium*)
XML (ang. *eXtensible Markup Language*)
SOAP (ang. *Simple Object Access Protocol*)
WSDL (ang. *Web Services Description Language*)
UDDI (ang. *Universal Description Discovery and Integration*)
GIS (ang. *Geographical Information System*)
SDI (ang. *Spatial Data Infrastructure*)
ISO (ang. *International Standards Organization*)
WMS (ang. *Web Map Service*)
WFS (ang. *Web Feature Service*)
WPS (ang. *Web Processing Service*)
GML (ang. *Geography Markup Language*)
SRG (ang. *Seeded Region Growing*)
SOA (ang. *Service Oriented Architecture*)
IT (ang. *Information Technology*)

Rozdział 1

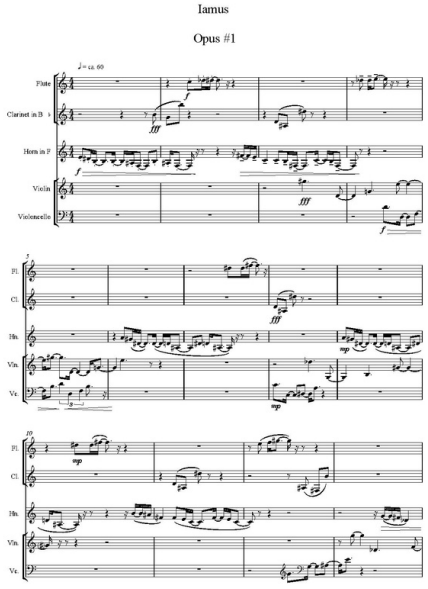
Wstęp

1.1. Wprowadzenie

Rozpowszechnione algorytmy sztucznej inteligencji wspomagające pracę inżynierów dźwięku można podzielić na dwie główne dziedziny [1] :

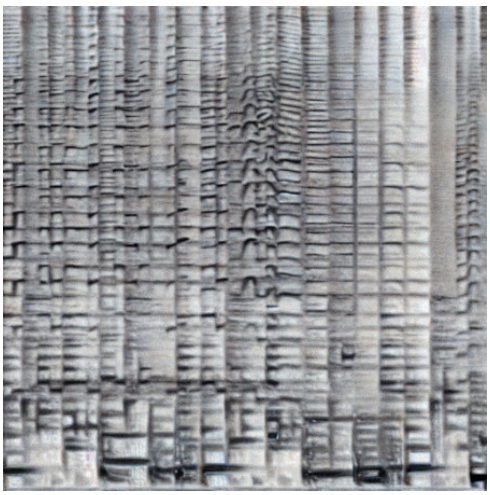
1. algorytmy generujące symboliczny zapis nutowy (lub plik MIDI) (1.1),
2. algorytmy generujące gotowy plik audio (1.2).

Pierwsza grupa algorytmów znana jest już od lat 80, gdyż zagadnienie generowania zapisu symbolicznego jest mniej wymagające niż wytworzenie pełnego pliku audio. Powszechnie wykorzystywana jest w nich teoria muzyki, pozwalająca określić matematyczne relacje w rytmach, melodiach i progresjach akordów. Wiedza dotyczącą teorii muzyki pozwala na wyznaczenie możliwej przestrzeni stanów, w której generowana jest kompozycja, natomiast modele matematyczne takie jak łańcuchy markova służą za mechanizmy decyzyjne, które "nawigują"w przestrzeni stanów.



Rys. 1.1: Zapis nutowy utworu *Opus One*, wygenerowany przez komputer *Lamus*.

Druga grupa algorytmów, generująca gotowe nagrania, rozwija się korzystając z nowych możliwości, które zapewniają algorytmy z grupy stable diffusion [3]. Najnowsze projekty generujące pliki audio zgodne z opisem tekstowym (przykładowo `"smutny jazz"` bądź `"muzyka taneczna w stylu Depeche Mode"`) szkolone są w taki sam sposób jak algorytmy stable diffusion. Zamiast na obrazach artystów, modele takie jak Stable Riffusion [2] szkolone są na spektrogramach, które następnie są w stanie wygenerować. Po wygenerowaniu spektrogramu przez model (1.2), jest on konwertowany do pliku audio za pomocą odwrotnej transformaty Fouriera.



Rys. 1.2: Przykładowy spektrogram wygenerowany przez algorytm *Stable Riffusion* dla danych wejściowych `funk bassline with a jazzy saxophone solo`.

Tematyka realizowanej pracy magisterskiej obraca się wokół zagadnień związanych z komputerową kompozycją i improwizacją, realizuje jednak zadanie, którego nie da się zaklasyfikować do żadnej z dwóch wyżej wymionionych dziedzin. Generowanie grafów przetwarzania sygnałów dźwiękowych może być porównane z procesem projektowania instrumentu muzycznego.

Opisane wyżej (1.1) algorytmy pozwalają muzykowi lub inżynierowi dźwięku na zainspirowanie się melodią lub wygenerowanym plikiem dźwiękowym. Wynik pracy algorytmu implementowanego w ramach pracy magisterskiej jest **gotowym instrumentem elektronicznym**, który może być wykorzystany w programie do komponowania muzyki.

1.2. Cel i zakres pracy

Celem pracy jest zbadanie, czy algorytmy optymalizacyjne są w stanie samodziel- nie wytworzyć graf przetwarzania sygnałów audio, który wykona syntezę próbki dźwięku zadanej przez użytkownika. Problem poruszany w pracy można zakwalifikować do grupy zagadnień związanych z pojęciem *computer-aided design* w dziedzienie inżynierii dźwięku. Docelowo zaimplementowany algorytm będzie automatyzował pracę inżyniera dźwięku, tworząc i konfigurując grafy przetwarzania sygnałów dźwiękowych, tak jak robi się to w programach typu *digital audio workstation* (Ableton, FL Studio). Badania obejmują dwa zagadnienia:

1. metody generowania grafu przetwarzania sygnałów oraz późniejszej modyfikacji grafu,
2. dobór funkcji błędu, na podstawie której algorytm optymalizujący będzie przeszukiwał możliwą przestrzeń grafów przetwarzania sygnałów.

Pierwsze zagadnienie sprowadza się do przetestowania szeregu algorytmów pozwalających na wygenerowanie grafu przetwarzania sygnałów DSP oraz ich modyfikację (przykładem mody-

fikacji grafu może być wprowadzanie w nim losowych zmian lub krzyżowanie dwóch grafów DSP w przypadku wykorzystania algorytmu genetycznego).

Drugie zagadnienie obejmuje przetestowanie szeregu algorytmów, które można wykorzystać jako funkcję celu, która będzie optymalizowana poprzez konfigurowanie grafu przetwarzania sygnałów dźwiękowych.

Rozdział 2

Praca z szablonem

2.1. Środowisko

Szablon powinien dać się skompilować w dowolnym środowisku, w którym zainstalowano system LaTeX. Zalecaną konfiguracją, która działa w systemie Windows, jest: MiKTeX (windowsowa dystrybucja latexa) + TeXnicCenter (środowisko do edycji i kompilacji projektów latexowych) + SumatraPDF (przeglądarka pdfów z nawigacją zwrotną) + JabRef (opcjonalny edytor bazy danych bibliograficznych). Narzędzia te można pobrać ze stron internetowych, których adresy zamieszczono w tabeli 2.1.

Tab. 2.1: Wykaz zalecanych narzędzi do pracy z wykorzystaniem szablonu (na dzień 09.02.2021)

Narzędzie	Wersja	Opis	Adres
MiKTeX	22.12	Zalecana jest instalacja Basic MiKTeX 32 lub 64 bitowa. Brakujące pakiety będą się doinstalowywać podczas kompilacji projektu	http://miktex.org/download
TeXnicCenter	2.02	Można pobrać 32 lub 64 bitową wersję	http://www.texniccenter.org/download/
SumatraPDF	3.4.6	Można pobrać 32 lub 64 bitową wersję	http://www.sumatrapdfreader.org/download-free-pdf-viewer.html
JabRef	5.7	Rozwijane w JDK 15, ma własny instalator i wersję przenośną	http://www.fosshub.com/JabRef.html

Wspomniana nawigacja między TeXnicCenter a SumatraPDF polega na przełączaniu się pomiędzy tymi narzędziami z zachowaniem kontekstu położenia kursora. Czyli edytując tekst w TeXnicCenter po kliku na ikonce podglądu można przeskoczyć do odpowiedniego miejsca w pdfie wyświetlanym przez SumatraPDF. Podwójne kliknięcie zaś w pdfie widocznym w SumatraPDF ustawi kursor we właściwym akapicie w edytorze tekstu TeXnicCenter. O konfiguracji obu narzędzi do takiej współpracy napisano na stronie http://tex.stackexchange.com/questions/116981/how-to-configure-texniccenter-2-0-with-sumatra-2013-2014-2015-version (w sieci można znaleźć również inne materiały na ten temat).

Szablon można również kompilować za pomocą innych narzędzi i środowisk (np. za pomocą multiplatformowego TexStudio oraz różnych dystrybucji systemu LaTeX). Może to jednak czasem wymagać zmiany sposobu kodowania plików i korekty deklaracji kodowania znaków w dokumencie głównym (co opisano dalej). Można go też zaadoptować do wymogów LaTeX-owych

edytorów i kompilatorów działających w trybie on-line, oferowanych na przykład w usłudze Overleaf (https://www.overleaf.com/).

Choć usługa Overleaf ma niewątpliwie wiele zalet (pozwala na edycję kodu przez wielu użytkowników jednocześnie), to podczas pracy nad długimi dokumentami przegrywa ona ze wspomnianymi zintegrowanymi środowiskami TeXnicCenter czy TexStudio. W Overleaf nie można wyświetlić struktury całego projektu (a jedynie listę plików projektowych oraz strukturę bieżąco edytowanego dokumentu), nie można uruchomić wyszukiwania we wszystkich plikach projektowych (nie ma opcji Find in Files ..., która bardzo się przydaje, gdy nie wiadomo, w którym pliku jest wyszukiwany tekst).

Overleaf wymaga użycia kodowania UTF8. Czyli aby dało się wczytać niniejszy szablon pracy dyplomowej do tego środowiska, pliki szablonu muszą być przekodowane, muszą też być zmienione opcje pakietu inputenc oraz odpowiednio zadeklarowane literate (aby dało się używać polskich znaków na listingach). Aby sprawę uprościć przygotowano dwie wersje szablonu: w kodowaniu ANSI oraz UTF8.

Choć w Overleaf istnieje możliwość skorzystania z wersjonowania (system ten integruje się z repozytorium git), to wersjonowanie to nie działa perfekcyjnie. Dlatego **zaleca się korzystanie ze środowisk zintegrowanych zainstalowanych lokalnie, czemu towarzyszyć ma wersjonowanie z pomocą wybranego zdalnego repozytorium (gitlab, github, bitbucket)**.

2.2. Struktura projektu

Pisząc pracę w systemie LaTeXzwykle przyjmuje się jakąś konwencję co do nazewnictwa tworzonych plików, ich położenia oraz powiązań. Przygotowując niniejszy szablon założono, że projekt będzie się składał z pliku głównego, plików z kodem kolejnych rozdziałów i dodatków (włączanych do kompilacji w dokumencie głównym), katalogów z plikami grafik (o nazwach wskazujących na rozdziały, w których grafiki te zostaną wstawione), pliku ze skrótami (opcjonalny), pliku z danymi bibliograficznymi (plik dokumentacja.bib). Taki „układ" zapewnia porządek oraz pozwala na selektywną kompilację rozdziałów.

Przyjętą konwencję da się opisać jak następuje:

- Plikiem głównym jest plik Dyplom.tex. To w nim znajdują się deklaracje wszystkich używanych styli, definicje makr oraz ustawień, jak również polecenie \begin{document}. W nim też należy ustawić metadane, które pojawią się na stronie tytułowej (oraz w stopkach).
- Teksty redagowane są w osobnych plikach. Pliki te zamieszczone są w katalogu głównym (tym samym, co plik Dyplom.tex).
- W pliku streszczenie.tex powinien pojawić się tekst streszczenia ze słowami kluczowymi (tekst ten oraz słowa kluczowe będzie można wykorzystać do wypełnienia formularzy pojawiających się podczas wysyłania pracy do analizy antyplagiatowej w systemie ASAP)
- Plik skroty.tex powinien zawierać wykaz użytych skrótów. Można z tego pliku zrezygnować, jeśli liczba stosowanych skrótów jest nieznaczna.
- Tekst kolejnych rozdziałów powinien pojawić się w plikach o nazwach zawierających numery tych rozdziałów. Według przyjętej konwencji rozdzial01.tex to plik pierwszego rozdziału, rozdzial02.tex to plik z treścią drugiego rozdziału itd.
- Teksty dodatków mają być zapisywane w osobnych plikach o nazwach zawierających literę dodatku. Pliki te, podobnie do plików z tekstem rozdziałów, zamieszczane są w katalogu głównym. I tak dodatekA.tex oraz dodatekB.tex to, odpowiednio, pliki z treścią dodatku A oraz dodatku B.
- Każdemu rozdziałowi i dodatkowi towarzyszy katalog przeznaczony do składowania dołączanych grafik. I tak rys01 to katalog na pliki z grafikami dołączanymi do rozdziału pierwszego,

rys02 to katalog na pliki z grafikami dołączanymi do rozdziału drugiego itd. Podobnie rysA to katalog na pliki z grafikami dołączanymi w dodatku A itd.
- W katalogu głównym zamieszczany jest plik dokumentacja.bib zawierający bazę danych bibliograficznych.
- Jeśli praca nad dokumentem odbywa się w TeXnicCenter, to zgodnie z wymogami tego narzędzia powinien istnieć jeszcze dodatkowy plik projektu. Plik ten pełni podobną rolę jak plik solucji wykorzystywany w zintegrowanych środowiskach programowania. Co prawda plik projektu nie jest wymagany do LaTeX-owej kompilacji (wystarczy kompilować plik główny), niemniej pozwala zapamiętać ustawienia środowiska (w tym ustawienia językowe potrzebne do sprawdzania poprawności wyrazów – patrz następny podrozdział). Plik projektu dostarczony w szablonie ma nazwę Dyplom.tcp.

Tab. 2.2: Pliki źródłowe szablonu oraz wyniki kompilacji

Źródła	Wyniki kompilacji
Dokument.tex - dokument główny Dokument.tcp – plik projektu TeXnicCenter streszczenie.tex – plik streszczenia skroty.tex – plik ze skrótami rozdzial01.tex – plik rozdziału 01 ... dodatekA.tex – plik dodatku A ... rys01 – katalog na rysunki do rozdziału 01 |- fig01.png – plik grafiki |- ... ... rysA – katalog na rysunki do dodatku A |- fig01.png – plik grafiki |- ... ... dokumentacja.bib – plik danych bibliograficznych Dyplom.ist – plik ze stylem indeksu by-nc-sa.png – plik z ikonami CC	Dyplom.bbl Dyplom.blg Dyplom.ind Dyplom.idx Dyplom.lof Dyplom.log Dyplom.lot Dyplom.out Dyplom.pdf – dokument wynikowy Dyplom.syntex Dyplom.toc Dyplom.tps *.aux Dyplom.synctex

2.3. Kodowanie znaków

System LaTeXobsługuje wielojęzyczność. Można tworzyć w nim dokumenty z tekstem zawierającym różne znaki diakrytyczne. Należy jednak zdawać sobie sprawę, w jaki sposób znaki te są obsługiwane. Na kodowanie znaków należy patrzeć z dwóch perspektyw: perspektywy edytowania kodu źródłowego oraz perspektywy kodowania dokumentu wynikowego i użytych czcionek.

Kod latexowy może być edytowany w dowolnym edytorze tekstów. Zastosowane kodowanie znaków w tym edytorze musi być znane LaTeX-owi, inaczej kompilacja tego kodu się nie powiedzie. Informację o tym kodowaniu przekazuje się w opcjach pakietu inputenc. Niniejszy szablon przygotowano w systemie Windows, a LaTeX-owe źródła umieszczono w plikach ANSI z użyciem strony kodowej cp1250. Dlatego w poleceniu \usepackage[cp1250]{inputenc} jako opcję wpisano cp1250.

Szablon można wykorzystać również przy innych kodowaniach i w innych systemach. Jednak wtedy konieczna będzie korekta dokumentu Dyplom.tex odpowiednio do wybranego przy-

padku. Korekta ta polegać ma na zamianie polecenia `\usepackage[cp1250]{inputenc}` na polecenie `\usepackage[utf8]{inputenc}` oraz konwersji znaków i zmiany kodowania istniejących plików ze źródłem latexowego kodu (plików o rozszerzeniu `*.tex` oraz `*.bib`).

Kodowanie znaków jest istotne również przy edytowaniu bazy danych bibliograficznych (pliku `dokumentacja.bib`). Aby `bibtex` poprawnie interpretował polskie znaki plik `dokumentacja.bib` powinien być zakodowany w ANSI, CR+LF (dla ustawień jak w szablonie). W szczególności, jeśli ominąć chce się problem kodowania, polskie znaki w bazie danych bibliograficznych można zastąpić odpowiednią notacją: `\k{a} \'c \k{e} \l{} \'n \'o \'s \'z \.z \k{A} \'C \k{E} \L{} \'N \'O \'S \'Z \.Z`.

Samo kodowanie plików może być źródłem paru problemów. Chodzi o to, że użytkownicy pracujący z edytorami tekstów pod linuxem mogą generować pliki zakodowane w UTF8 bez BOM (lub z BOM – co nie jest zalecane), a pod windowsem – pliki ANSI ze znakami ze strony kodewej `cp1250`. Z takimi plikami różne edytory różnie sobie radzą. W szczególności edytor `TeXnicCenter` podczas otwierania plików może potraktować jego zawartość jako UTF8 lub ANSI – prawdopodobnie interpretuje z jakim kodowaniem ma do czynienia na podstawie obecności w pliku znaków specjalnych. Bywa, że choć wszystko w `TeXnicCenter` wygląda na poprawne, to jednak kompilacja LaTeX-owa „nie idzie". Problemem mogą być właśnie pierwsze bajty, których nie widać w edytorze.

Do konwersji kodowania można użyć edytor `Notepad++` (jest tam opcja „konwertuj" – nie mylić z opcją „koduj", która przekodowuje znaki, jednak nie zmienia sposobu kodowania pliku).

Jeśli chodzi o drugą perspektywę, tj. kodowanie znaków w dokumencie wynikowym, to sprawa jest bardziej skomplikowana. Wiąże się z nią zarządzanie czcionkami, definiowanie mapowania itp. **Szablon przygotowano tak, by wynikowy dokument zawierał polskie znaki diakrytyczne, które nie są zlepkami literki i ogonka.**

2.4. Kompilacja szablonu

Kompilację szablonu można uruchamiać na kilka różnych sposobów. Wszystko zależy od używanego systemu operacyjnego, zainstalowanej na nim dystrybucji LaTeX-a oraz dostępnych narzędzi. Zazwyczaj kompilację rozpoczyna się wydając polecenie z linii komend lub uruchamia się ją za pomocą narzędzi zintegrowanych środowisk.

Kompilacja z linii komend polega na uruchomieniu w katalogu, w którym rozpakowano źródła szablonu, następującego polecenia:

```
> pdflatex Dyplom.tex
```

gdzie `pdflatex` to nazwa kompilatora, zaś `Dyplom.tex` to nazwa głównego pliku redagowanej pracy. W przypadku korzystania ze środowiska `TeXnicCenter` należy otworzyć dostarczony w szablonie plik projektu `Dyplom.tcp`, a następnie uruchomić kompilację narzędziami dostępnymi w pasku narzędziowym.

Aby poprawnie wygenerowały się wszystkie referencje (spis treści, odwołania do tabel, rysunków, pozycji literaturowych, równań itd.) kompilację `pdflatex` należy wykonać dwukrotnie, a czasem nawet trzykrotnie. Wynika to z konieczności zapamiętywania pośrednich wyników kompilacji i ich wykorzystywania w kolejnych przebiegach. Tak dzieje się przy generowaniu odwołań do pozycji literaturowych oraz tworzeniu ich wykazu).

Wygenerowanie danych bibliograficznych zapewnia kompilacja `bibtex` uruchamiana po kompilacji `pdfltex`. Można to zrobić z linii komend:

```
> bibtex Dyplom
```

lub wybierając odpowiednią pozycję z paska narzędziowego wykorzystywanego środowiska. Po kompilacji za pomocą `bibtex` na dysku pojawi się plik `Dyplom.bbl`. Dopiero po kolejnych

dwóch kompilacjach pdflatex dane z tego pliku zostaną odpowiednio przetworzone i zrenderowane w wygenerowanym dokumencie. Tak więc po każdym wstawieniu nowego cytowania w kodzie dokumentu uzyskanie poprawnego formatowania dokumentu wynikowego wymaga powtórzenia następującej sekwencji kroków kompilacji:

```
> pdflatex Document.tex
> bibtex Document
> latex Document.tex
> latex Document.tex
```

Szczegóły dotyczące przygotowania danych bibliograficznych oraz zastosowania cytowań przedstawiono w podrozdziale 5.4.

W głównym pliku zamieszczono polecenia pozwalające sterować procesem kompilacji poprzez włączanie bądź wyłączanie kodu źródłowego poszczególnych rozdziałów. Włączanie kodu do kompilacji zapewniają instrukcje \include oraz \includeonly. Pierwsza z nich pozwala włączyć do kompilacji kod wskazanego pliku (np. kodu źródłowego pierwszego rozdziału \include{rozdzial01.tex}). Druga, jeśli zostanie zastosowana, pozwala określić, które z plików zostaną skompilowane w całości (na przykład kod źródłowy pierwszego i drugiego rozdziału \includeonly{rozdzial01.tex,rozdzial02.tex}). Brak nazwy pliku na liście w poleceniu \includeonly przy jednoczesnym wystąpieniu jego nazwy w poleceniu \include oznacza, że w kompilacji zostaną uwzględnione referencje wygenerowane dla tego pliku wcześniej, sam zaś kod źródłowy pliku nie będzie kompilowany.

W szablonie wykorzystano klasę dokumentu memoir oraz wybrane pakiety. Podczas kompilacji szablonu w MikTeXu wszelkie potrzebne pakiety zostaną zainstalowane automatycznie (jeśli MikTeX zainstalowano z opcją dynamicznej instalacji brakujących pakietów). W przypadku innych dystrybucji LaTeX-a może okazać się, że pakiety te trzeba doinstalować ręcznie (np. pod linuxem z TeXLive trzeba doinstalować dodatkową zbiorczą paczkę, a jeśli ma się menadżera pakietów LaTeX-owych, to pakiety te można instalować indywidualnie).

Jeśli w szablonie będzie wykorzystany indeks rzeczowy, kompilację źródeł trzeba będzie rozszerzyć o kroki potrzebne na wygenerowanie plików pośrednich Dokument.idx oraz Dokument.ind oraz dołączenia ich do finalnego dokumentu (podobnie jak to ma miejsce przy generowaniu wykazu literatury). Szczegóły dotyczące generowania indeksu rzeczowego opisano w podrozdziale 5.5.

2.5. Sprawdzanie poprawności tekstu

Większość środowisk ułatwiających pisanie LaTeX-owych dokumentów wspiera sprawdzenie poprawności tekstu (ang. *spell checking*). Wystarczy odpowiednio je skonfigurować. Niestety, proponowana przez narzędzia korekta nie jest genialna. Bazuje ona na prostym porównywaniu wyrazów (z końcówkami). Nie wbudowano w nią żadnej większej inteligencji. Tak więc proszę nie porównywać jej z korektą oferowaną w narzędziach MS Office (tam jest ona dużo bardziej zaawansowana).

TeXnicCenter korzysta ze słowników do pobrania ze strony openoffice (https://extensions.openoffice.org/). Aby sprawę uprościć słowniki dla języka polskiego (pliki pl_PL.aff oraz pl_PL.dic) dołączono do szablonu (są w katalogu Dictionaries). Pliki te należy umieścić w katalogu C:\Program Files\TeXnicCenter\Dictionaries, a w konfiguracji projektu (Tools/Options/Spelling) należy wybrać Language: pl, Dialect: PL. Jeśli główny tekst pracy pisany jest w innym języku, to trzeba zmienić słownik.

Zaskakujące może jest to, że *spell checker* w TeXnicCenter działa zarówno przy pracy na plikach UTF8, jak i na plikach ANSI. Jeśli byłyby jakieś problemy ze słownikiem wynikające z kodowania znaków, wtedy słownik trzeba przekodować. To powinno pomóc.

2.6. Wersjonowanie

W trakcie edytowania pracy w systemie latex dobrą praktyką jest wersjonowanie tworzonego kodu. Do wersjonowania zaleca się wykorzystać system `git`. Opis sposobu pracy z tym systemem opisano w licznych tutorialach dostępnych w sieci. Szczególnie godnym polecenia zasobem jest strona domowa projektu `https://git-scm.com/`.

Zwykle pracę z systemem wersjonowania rozpoczyna się od utworzenia repozytorium zdalnego i lokalnego. Lokalne służy do bieżącej pracy, zdalne – do współpracy z innymi użytkownikami (z promotorem).

Po utworzeniu repozytorium lokalnego i roboczej gałęzi (czy będzie to `master` czy inna gałąź – ustalają to sami zainteresowani) należy skopiować do niego wszystkie pliki dostarczone w szablonie, a po ich wstępnym przeredagowaniu należy je zaznaczyć do wersjonowania. Potem należy wysłać zmiany na repozytorium zdalne (możliwa jest też ścieżka odwrotna, można zacząć od zmian na repozytorium zdalnym, które pobrane będą do repozytorium lokalnego).

Podczas kompilowania projektu będą powstawały pliki pomocnicze. Plików tych nie należy wersjonować (zabierają niepotrzebnie miejsce, a przecież zawsze można je odtworzyć uruchamiając kompilację na źródłach). `git` posiada mechanizm automatycznego odrzucania plików niepodlegających wersjonowaniu. Mechanizm ten bazuje na wykorzystaniu pliku konfiguracyjnego `.gitignore` zamieszczonego w katalogu głównym repozytorium. O szczegółach `.gitignore` można poczytać na stronie `https://git-scm.com/docs/gitignore`.

W sieci można znaleźć liczne propozycje plików konfiguracyjnych `.gitignore` dopasowanych do potrzeb latexowej kompilacji. Nie trzeba ich jednak szukać. W szablonie zamieszczono przykładowy taki plik. Zawiera on, między innymi, wpis zabraniający wersjonowania pliku wynikowego `Dyplom.pdf`.

Plik `.gitignore` należy umieścić w repozytorium zaraz po jego utworzeniu. Jeśli w repozytorium pojawiły się już jakieś zmiany zanim pojawił się tam `.gitignore`, to wtedy należy wykonać kroki opisane na stronie: `https://stackoverflow.com/questions/38450276/force-git-to-update-gitignore/38451183`

```
> > > > You will have to clear the existing git cache first.
    git rm -r --cached .
> > > > Once you clear the existing cache, adds/stages all of the
   ↪ files in the current directory and commit
    git add .
    git commit -m "Suitable Message"
> > > >
```

Zalecany schemat współpracy dyplomanta z promotorem polega na wykonywaniu w kolejnych iteracjach następujących kroków:

- dyplomant edytuje wybraną część pracy, a po skończeniu edycji wrzuca zmiany do zdalnego repozytorium,
- dyplomant informuje promotora o zakończeniu etapu prac,
- dyplomant może zacząć edycję kolejnego fragmentu pracy, a w tym czasie promotor może dokonać oceny/korekty zmian pobranych ze zdalnego repozytorium,
- promotor wrzuca dokonane przez siebie zmiany do zdalnego repozytorium, informując o tym dyplomanta.

Główna zasada tego schematu polega na niedoprowadzaniu do konfliktów (nadpisywania zmian). Jeśli jednak takie konflikty nastąpią, można je niwelować poprzez odpowiednie scalanie zmian (merdżowanie). Swoją drogą, niezłym narzędziem do porównywania zawartości plików i usuwania niezgodności jest `WinMerge` (`https://winmerge.org/`).

Do wzajemnych powiadomień można wykorzystać pocztę elektroniczną. Można też spróbować wdrożyć mechanizm zatwierdzania zmian (ang. *merge requests*). Można też umówić się na sprawdzanie zawartości repozytorium zgodnie z jakimś przyjętym harmonogramem. Ważne, by wiadomo było obu stronom, na jakim schemacie współpracy mają bazować.

Dobrą praktyką jest też wstawianie w kod komentarzy. Przyjętą powszechnie konwencją jest rozpoczynanie komentarzy od:

- `% TO DO: tekst zalecenia` – jeśli jest to jakieś zalecenie promotora, czy też
- `% DONE: tekst wyjaśnienia` – jeśli jakieś zalecenie zostało wykonane przez dyplomanta.

Jako zdalne repozytorium można wykorzystać: `github`, `bitbucket`, `gitlab` (są to serwisy, które pozwalają zarządzać repozytoriami `git`). Dobrą praktyką jest też uruchomienie klientów git oferujących graficzny interfejs (jak `SourceTree`, `GitCracken` itp.). W narzędziach tych można zobaczyć natychmiast na czym polegały wprowadzone w repozytorium zmiany.

Rozdział 3

Zalecenia dotyczące formatowania

Większa część niniejszego rozdziału ma charakter informacyjny. Treści w nim zawarte mają uzmysłowić czytelnikowi, jak wiele jest parametrów do ustawienia, by zredagowany dokument wyglądał „ładnie”. Parametry te poustawiano w szablonie. Wystarczy z niego skorzystać.

3.1. Rozmiar i układ treści na stronach dokumentu

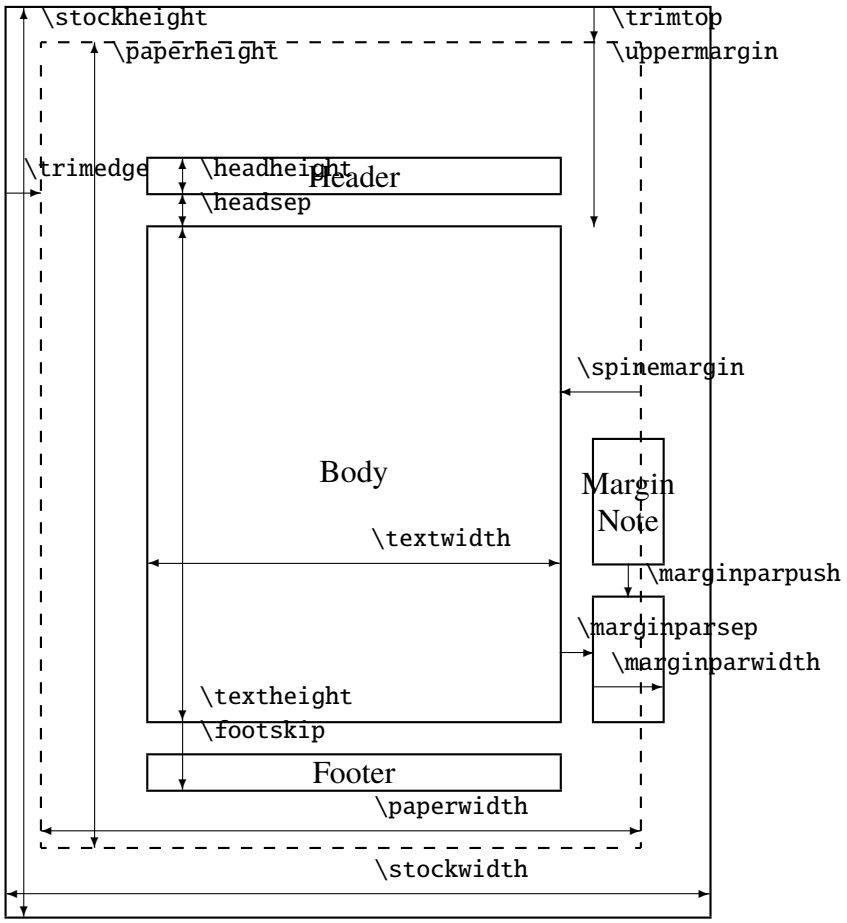
Praca dyplomowa powinna być przygotowana do wydruku na papierze formatu A4 w orientacji pionowej. Marginesy na stronach parzystych i nieparzystych powinny być jednakowe i mieć następujące wartości: lewy = 25mm, prawy = 25mm, górny = 10mm, dolny = 15mm. Wielkość marginesów w szablonie sterowana jest parametrami przedstawionymi na rysunku 3.1. Margines dolny powinien być mierzony do linii bazowej tekstu stopki.

Rzeczywisty układ strony zastosowany w niniejszym dokumencie przedstawiono na rysunku 3.2. Lewy i prawy margines są takie same, więc strony parzyste i nieparzyste wyglądają podobnie, z dokładnością do umiejscowienia notatek marginesowych. Taki rezultat zapewniło zastosowanie poniższych komend.

```
\setlength{\headsep}{10pt}
\setlength{\headheight}{13.6pt}
\setlength{\footskip}{\headsep+\headheight}
\setlength{\uppermargin}{\headheight+\headsep+1cm}
\setlength{\textheight}{\paperheight-\uppermargin-\footskip-1.5cm}
\setlength{\textwidth}{\paperwidth-5cm}
\setlength{\spinemargin}{2.5cm}
\setlength{\foremargin}{2.5cm}
\setlength{\marginparsep}{2mm}
\setlength{\marginparwidth}{2.3mm}
\checkandfixthelayout[fixed]
\linespread{1}
\setlength{\parindent}{14.5pt}
```

3.2. Strona tytułowa

Stronę tytułową przygotowano starając się wypełnić zalecenia zamieszczone na stronie Wydziału Informatyki i Teleinformatyki (`https://wit.pwr.edu.pl/studenci/dyplomanci/praca-dyplomowa`, [dostęp dnia 16.12.2022]). Zastosowano więc pakiet `ebgaramond`. Dostarcza on klonu wymaganej czcionki `garamond`, jednak bez kształtu `slanted` i z pewnymi brakami. Na przykład zamiast literki „ł” w zbiorze `EBGaramond08 Italic` renderuje się samo „l” (braku



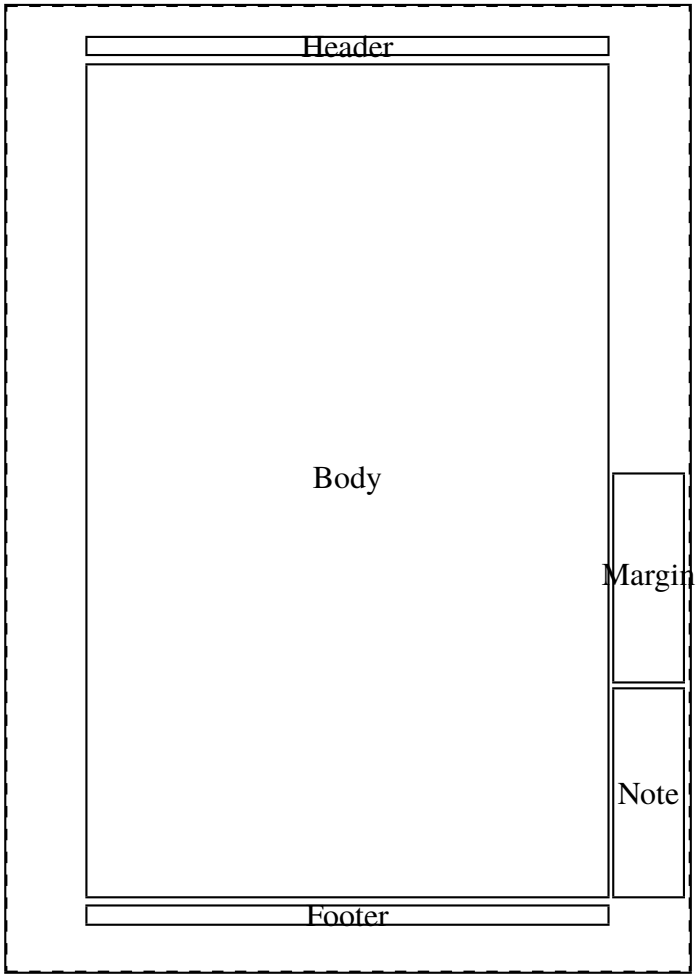
Rys. 3.1: Układ strony nieparzystej dla dokumentu klasy `memoir`

tego nie ma zbiór `EBGaramond12`). Zaletą pakietu w porównaniu do innych jest to, że generalnie dobrze obsługiwane są w nim polskie znaki oraz że pakiet ten można znaleźć w różnych dystrybucjach latexa (`MikTeX` instaluje go automatycznie).

Wielkości znaków użytych do wypełnienia strony tytułowej treścią są następujące:

```
Politechnika Wrocławska (Garamond Bold 20pt 22pt)
Wydział Informatyki i Teleinformatyki (Garamond Bold 16pt 18pt)
Kierunek: Jakiś kierunek (Garamond 14pt 16pt, Garamond Bold 14pt 16pt)
Specjalność: Jakaś specjalność (Garamond 14pt 16pt, Garamond Bold 14pt 16pt
    ↪ )
{P}RACA {D}YPLOMOWA (Garamond 26pt 28pt, Garamond 24pt 26pt)
{I}NŻYNIERSKA (Garamond 26pt 28pt, Garamond 24pt 26pt)
Tytuł pracy w języku polskim (Garamond Bold 18pt 20pt)
Title in English (Garamond Bold 18pt 20pt)
% AUTOR: (Garamond 16pt 18pt) - zamarkowane
Imię Nazwisko (Garamond Bold 16pt 18pt)
Opiekun pracy (Garamond 14pt 16pt)
tytuł/stopień naukowy, Imię Nazwisko (Garamond 14pt 16pt)
%OCENA PRACY: (Garamond 16pt 18pt) - zamarkowane
WROCŁAW, 2021 (Garamond 14pt 16pt)
```

Dashed lines represent the actual page size after trimming the stock.



Lengths are to the nearest pt.

\stockheight = 845pt	\stockwidth = 598pt
\pageheight = 845pt	\pagewidth = 598pt
\textheight = 727pt	\textwidth = 455pt
\trimtop = 0pt	\trimedge = 0pt
\uppermargin = 52pt	\spinemargin = 71pt
\headheight = 14pt	\headsep = 10pt
\footskip = 24pt	\marginparsep = 6pt
\marginparpush = 7pt	\columnsep = 10pt
\columnseprule = 0.0pt	

Rys. 3.2: Rzeczywisty układ strony nieparzystej w tym dokumencie

3.3. Główny tekst

Główny tekst pracy powinien być zredagowany z wykorzystaniem czcionki `Times`, typ normalny, o wysokości 12pt, z odstępem między liniami równym 14.5pt. Istnieje możliwość zmiany odstępu między liniami za pomocą komendy `\linespread`, jednak zaleca się pozostawienie tego odstępu jak w niniejszym dokumencie (`\linespread{1}`). Wymagania odnośnie kroju pisma pozostałych elementów (nagłówków, stopek itp.) zamieszczono w tabeli 3.1.

W szablonie zastosowano czcionkę `texgyre-termes` (dostarcza ją pakiet `tgtermes`). Czcionka ta jest klonem czcionki `Times`, w którym obsługiwane jest środkowoeuropejskie ko-

dowanie znaków (podobnie jak w przypadku czcionki ebgaramond, dzięki czemu polskie literki nie są zlepkami dwóch znaków lecz pojedynczymi znakami).

Wszelkie przykłady źródeł kodu (fragmenty programów, komendy linii poleceń), nazwy plików i uruchamianych programów powinny być pisane czcionką maszynową. W szablonie czcionką maszynową jest t1xtt. Czcionka ta obsługuje polskie znaki. Dostarcza ją pakiet txfonts, który należy wcześniej zainstalować (MiKTeX zainstaluje go automatycznie podczas pierwszej kompilacji szablonu).

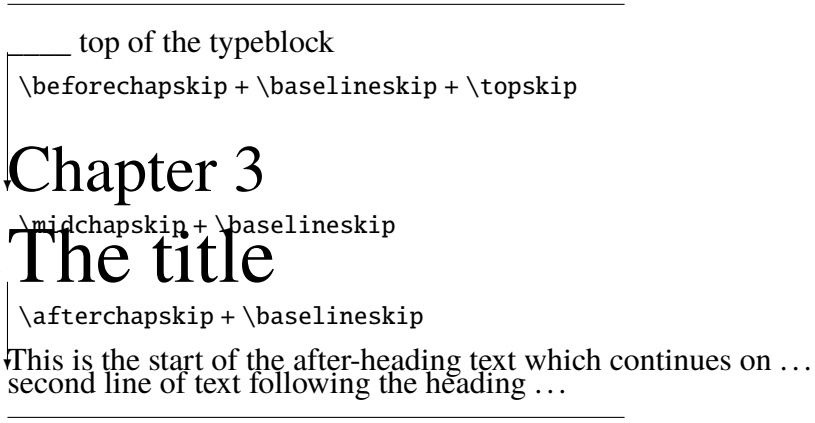
Tab. 3.1: Zestawienie czcionek elementów podziału dokumentu, tekstu wiodącego, nagłówka i stopki oraz podpisów (Rozm. – rozmiar czcionki, Odst. – baselineskip)

Element	Przykład	Czcionka	Rozm.	Odst.
Nr rozdziału	**Rozdział 1**	\huge \bfseries	25pt	30pt
Tytuł rozdziału	**Wstęp**	\Huge \bfseries	30pt	37pt
Nr i tytuł sekcji	**1.1. Wprowadzenie**	\Large \bfseries	17pt	22pt
Nr i tytuł podsekcji	**1.1.1. Cel szczegółowy**	\large \bfseries	14.5pt	18pt
Tytuł podpodsekcji	**Założenia**	\normalsize \bfseries	12pt	14.5pt
Tytuł paragrafu	**Podstawy** Opis ...	\normalsize \bfseries	12pt	14.5pt
Tekst wiodący	Niniejszy dokument ...	\normalsize	12pt	14.5pt
Nagłówek strony	*3.2. Czcionka wiodąca ...*	\small \itshape	11pt	13.6pt
Stopka strony	Imię Nazwisko: ...	\small	11pt	13.6pt
Podpisy tabel	Tab. 3.1: Zestawienie ...	\small	11pt	13.6pt
Podpisy rysunków	Rys. 3.1: Oficjalny ...	\small	11pt	13.6pt

Jeśli w pracy zostaną użyte otoczenia matematyczne, to w dokumencie wynikowym pojawią się dodatkowe czcionki (domyślne latexowe czcionki do wyrażeń matematycznych). Dzięki zastosowaniu opcji extrafontsizes w klasie memoir nie dość, że otrzymuje się większe czcionki (30pt), to jeszcze zamiast Computer Modern do wzorów matematycznych jest stosowana czcionka Latin Modern (wywodząca się z Computer Modern). Stąd lista wszystkich użytych czcionek może być następująca:

```
EBGaramond12-Regular
GaramondNo8-Reg-Norml
TeXGyreTermes-Regular-Normalna
TeXGyreTermes-Bold-Pogrubiona
TeXGyreTermes-Italic-Normalna
t1xtt-Nomal
LMMathItalic12-Regular
LMMathSymbols10-Regular
LMMathExtension10-Regular
LMRoman8-Regular
```

Aby wykorzystać te czcionki poza systemem LaTeX, wystarczy pobrać je spod adresów (ważnych na dzień 1.04.2016): https://www.ctan.org/tex-archive/fonts/cm/ps-type1/bakoma/ttf/?lang=en, http://www.gust.org.pl/projects/e-foundry/latin-modern, http://www.gust.org.pl/projects/e-foundry/tex-gyre, https://bitbucket.org/georgd/eb-garamond/downloads, a następnie zainstalować w systemie. Dzięki temu można będzie np. edytować rysunki używając dokładnie tej samej czcionki, co czcionka użyta w dokumencie.



Rys. 3.3: Parametry sterujące wielkościami odstępów na stronie z tytułem rozdziału

3.4. Formatowanie bloków tekstu

Każdy rozdział pracy powinien rozpoczynać się od nowej strony. Jej wygląd powinien być kontrolowany parametrami pokazanymi na rysunku 3.3. W niniejszym szablonie (dokument klasy `memoir` z opcją `[12pt]`) przyjęto następujące wartości tych parametrów:

- `\beforechapskip` (50.0pt) + `\baselineskip` of `\huge` (30pt) + `\topskip` (12.0pt) = 92pt (3.246cm)
- `\midchapskip` (20.0pt) + `\baselineskip` of `\Huge` (37pt) = 57 pt (2.011cm)
- `\afterchapskip` (40.0pt) + `\baselineskip` of `\normalsize` (14.5pt) = 54.5pt (1.923cm)

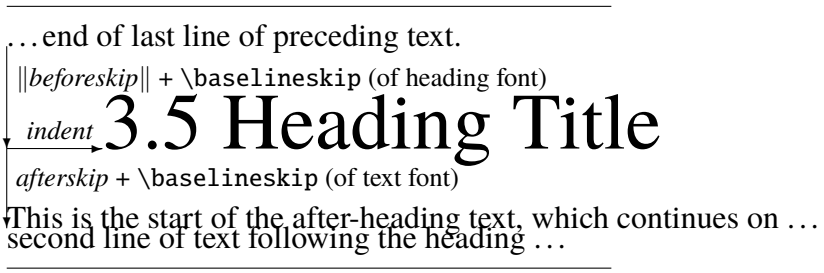
Nieco kłopotów może sprawić dobre ustawienie na stronie tytułów nienumerowanych rozdziałów oraz list generowanych automatycznie (Skróty, Spis treści, Spis rysunków, Spis tabel, Indeks rzeczowy). W szablonie w tym celu zdefiniowano nowy styl rozdziału komendami jak niżej (w szablonie są to komendy zamarkowane)

```
\newlength{\linespace}
\setlength{\linespace}{-\beforechapskip-\topskip+\headheight+\topsep}
\makechapterstyle{noNumbered}{%
\renewcommand\chapterheadstart{\vspace*{\linespace}}
}
```

oraz dokonano przełączenia stylów rozdziałów komendami `\chapterstyle{nonumbered}` oraz `\chapterstyle{default}` podczas dołączania do dokumentu wymienionych nienumerowanych rozdziałów i list. Aby „podnieść do góry” tytuły nienumerowanych rozdziałów (gdy jest to rzeczywiście konieczne) wystarczy odmarkować wspomniane komendy.

Tytuły rozdziałów, sekcji, podsekcji itd. nie powinny kończyć się kropką. Odległości pomiędzy tekstem wiodącym a tytułem sekcji powinien być regulowany parametrami pokazanymi na rysunku 3.4. Rozmiar `\baselineskip` zależy od rozmiaru czcionki (zobacz tabela 3.1), zaś `beforeskip` i `secskip` od poziomu sekcji. W niniejszym szablonie przyjęto następujące wartości tych parametrów (są to wartości dobierane elastycznie podczas kompilacji):

- `indent = 14.5pt`
- `parskip = 0.0pt`
- `beforesecskip = -18.08334pt plus -5.16667pt minus -1.03331pt`
- `aftersecskip = 11.88335pt plus 1.03331pt`
- `beforesubsecskip = -16.79167pt plus -5.16667pt minus -1.03331pt`
- `aftersubsecskip = 7.75pt plus 1.03331pt`
- `beforesubsubsecskip = -16.79167pt plus -5.16667pt minus -1.03331pt`



Rys. 3.4: Kontrola ustawień odległości w tytułach kolejnych sekcji

- `aftersubsubsecskip = 7.75pt plus 1.03331pt`

W szablonie obowiązują również następujące wartości parametrów odpowiedzialnych za odstępy pomiędzy pływającymi figurami, tekstami oraz tekstem i figurą:

- `floatsep = 12.0pt plus 2.0pt minus 2.0pt`
- `intextsep = 14.0pt plus 4.0pt minus 4.0pt`
- `textfloatsep = 20.0pt plus 2.0pt minus 4.0pt`

Pierwsza linia pierwszego akapitu w bloku (po tytule rozdziału, sekcji, podsekcji, podpodsekcji) nie może mieć wcięcia. Pierwsze linie w kolejnych akapitach już powinny mieć wcięcie równe `14.5pt`. Tekst w akapitach powinien być wyrównany z obu stron.

Strony powinny być numerowane numeracją ciągłą (sekwencja arabskich cyfr). Numery stron powinny być umieszczone w ich stopkach (tj. tak jak w niniejszym dokumencie). Wyjątkiem są tutaj pierwsze strony rozdziałów oraz strona tytułowa – na nich numery nie powinny się pojawić.

3.5. Opisy tabel i rysunków

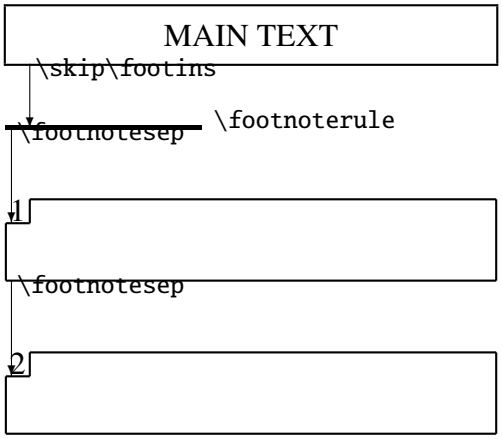
Podpisy powinny być umieszczane pod rysunkami lub nad tabelami wraz z etykietą składającą się ze skrótu Rys. lub Tab. oraz numeru. Podpisy te nie powinny mieć końcowej kropki. Numery występujący w podpisach powinny zaczynać się numerem rozdziału, po którym następuje kolejny numer rysunku lub tabeli w obrębie rozdziału. Etykieta powinna kończyć się dwukropkiem, po którym następuje tekst podpisu. Numer rozdziału powinien być rozdzielony kropką od kolejnego numeru w rysunku bądź tabeli w rozdziale (liczniki tabel i rysunków są rozłączne). Należy pamiętać o tym, żeby w całej pracy tabele miały podobny wygląd (rodzaj czcionki, ewentualne pogrubienia w nagłówku itp.).

3.6. Przypisy dolne

Istnieje możliwość zamieszczania przypisów na dole strony, choć nie jest to zalecane (przykładowo [1]). Sposób parametryzowania ich wyglądu pokazano na rysunku 3.5. W szablonie wykorzystano następujące, domyślne wartości tych parametrów:

```
\footins = 12pt \footnotesep = 8pt
\baselineskip = 10pt note separation = 40pt
rule thickness = 0.4pt
rule length = 0.25 times the \textwidth
```

[1] Tekst przypisu

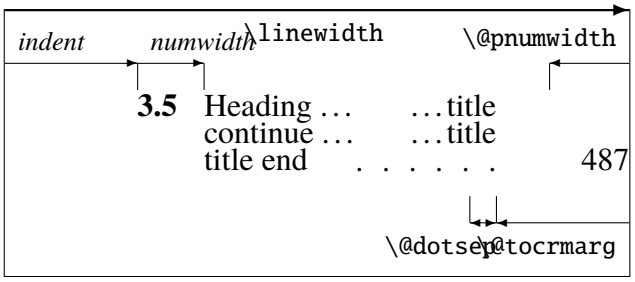


Rys. 3.5: Parametry sterujące przypisami dolnymi

3.7. Formatowanie spisu treści

W klasie `memoir` istnieją komendy pozwalające dość dobrze zarządzać wyglądem spisu treści. Na rysunku 3.6 pokazano, za pomocą jakich parametrów można wpływać na finalną jego postać. W szablonie wykorzystano następujące, domyślne ich wartości:

```
indent = 18pt
numwidth = 28pt
\@tocrmarg = 31pt
\@pnumwidth = 19pt
\@dotsep = 4.5
```



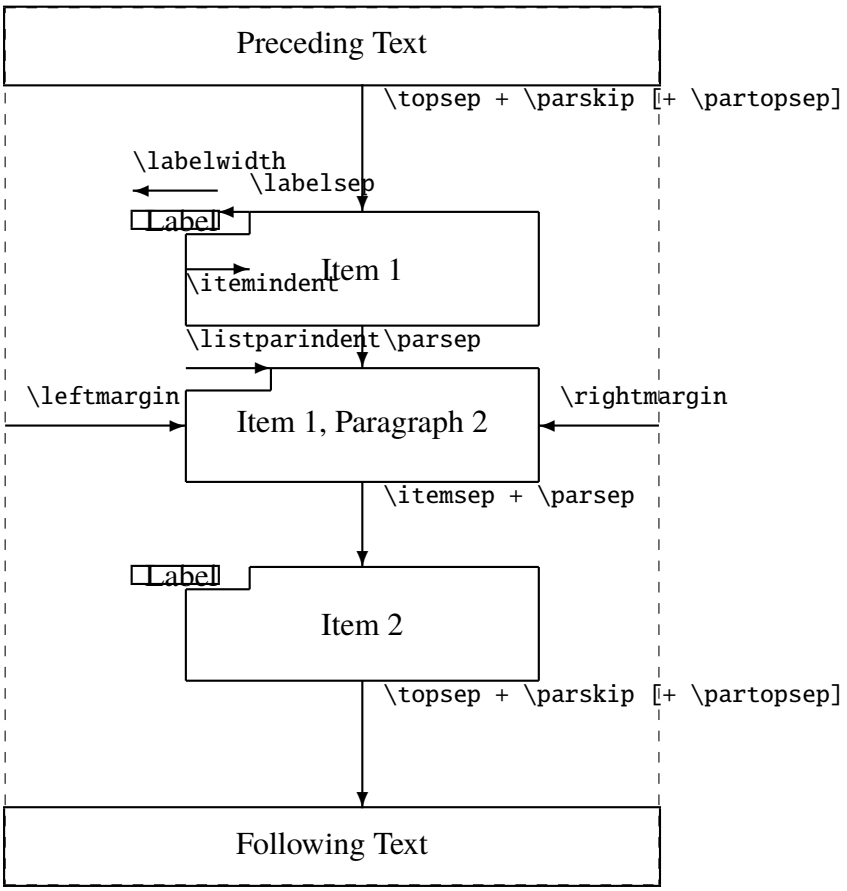
Rys. 3.6: Parametryzacja wyglądu spisu treści

3.8. Formatowanie list wyliczeniowych i wypunktowań

Standardowo sposób formatowania list można parametryzować jak pokazano na rysunku 3.7. Jednak czasem trudno poradzić sobie z niektórymi rzeczami, jak np. znakami wypunktowania. Dlatego w szablonie wykorzystano pakiet `enumi`. Pozwala on na łatwe zarządzanie wyglądem list. W szablonie zastosowano następujące globalne ustawienia dla tego pakietu:

```
\usepackage{enumitem}
\setlist{noitemsep,topsep=4pt,parsep=0pt,partopsep=4pt,leftmargin=*}
\setenumerate{labelindent=0pt,itemindent=0pt,leftmargin=!,label=\arabic*.}
\setlistdepth{4}
\setlist[itemize,1]{label=$\bullet$}
\setlist[itemize,2]{label=\normalfont\bfseries\textendash}
```

```
\setlist[itemize,3]{label=$\ast$}
\setlist[itemize,4]{label=$\cdot$}
\renewlist{itemize}{itemize}{4}
```



Rys. 3.7: Parametryzacja list wyliczeniowych i wypunktowań

W podrozdziale 4.2 pokazano przykład wykorzystania możliwości komend oferowanych w pakiecie `enumi`.

3.9. Wzory matematyczne

Wzory matematyczne, jeśli mają być osobnymi formułami, powinny być wycentrowane, z numeracją umieszczoną na końcu linii i ujętą w okrągłe nawiasy (zobacz równanie (3.1)). Numery równań powinny zawierać numer rozdziału oraz kolejny numer równania w obrębie rozdziału (podobnie jak przy numerowaniu rysunków i tabel). Spełnienie tych warunków zapewnia otoczenie `equation`. Nie wszystkie formuły trzeba numerować (nienumerowane wzory można osiągnąć stosując otoczenie `\equation*`). Właściwie należy numerować tylko te, do których tworzy się jakieś odniesienia w tekście. Jeśli wzory umieszczane są w linijce tekstu, to można zastosować otoczenie matematyczne inline, jak w przykładzie $\int_0^{10\nu \sum i} xdx$ (wyprodukowanym komendą `$\int_{0}^{10\nu\sum i}{x dx}$`). Tylko że wtedy może dojść do rozszerzenia odstępów pomiędzy liniami tekstu (aby zmieścił się wzór).

$$\int_0^{10\nu \sum i} xdx \tag{3.1}$$

Rozdział 4

Redakcja pracy dyplomowej

4.1. Układ pracy dyplomowej

Istnieje wiele sposobów organizacji prac dyplomowych. W dokumentacji klasy `memoir` (`http://tug.ctan.org/tex-archive/macros/latex/contrib/memoir/memman.pdf`) zamieszczono zalecenia odnoszące się do edytorskiej strony amerykańskich prac dyplomowych. Na stronie 363 można przeczytać, że część początkowa pracy może zawierać:

1. *Title page*
2. *Approval page*
3. *Abstract*
4. *Dedication (optional)*
5. *Acknowledgements (optional)*
6. *Table of contents*
7. *List of tables (if there are any tables)*
8. *List of figures (if there are any figures)*
9. *Other lists (e.g., nomenclature, definitions, glossary of terms, etc.)*
10. *Preface (optional but must be less than ten pages)*
11. *Under special circumstances further sections may be allowed*

Po części początkowej powinna pojawić się główna część pracy, złożona z kolejnych rozdziałów. Po niej powinna pojawić się część końcowa, na którą zwykle składają się:

1. *Notes (if you are using endnotes and grouping them at the end)*
2. *References (AKA 'Bibliography' or 'Works Cited')*
3. *Appendices*
4. *Biographical sketch (optional)*

Prace dyplomowe powstające na Wydziale Informatyki i Teleinformatyki Politechniki Wrocławskiej standardowo redaguje się w następującym układzie:

Strona tytułowa
Streszczenie
Strona z dedykacją (opcjonalna)
Spis treści
Spis rysunków (opcjonalny)
Spis tabel (opcjonalny)
Spis listingów (opcjonalny)
Skróty (wykaz opcjonalny)
Abstrakt
1. Wstęp
  1.1 Wprowadzenie
  1.2 Cel i zakres pracy
  1.3 Układ pracy
2. Kolejny rozdział
  2.1 Sekcja
    2.1.1 Podsekcja
      Nienumerowana podpodsekcja
        Paragraf
...
#. Podsumownie i wnioski
Literatura
A. Dodatek
  A.1 Sekcja w dodatku
...
$. Instrukcja wdrożeniowa
$. Zawartość płyty CD/DVD
Indeks rzeczowy (opcjonalny)

Poniżej zamieszczono opis poszczególnych elementów tego układu.

Streszczenie – syntetyczny opis zawartości pracy, zwykle po pół strony w języku polskim i języku angielskim (podczas wysyłania pracy do analizy antyplagiatowej należy skopiować tekst z tej sekcji do odpowiedniego pola formularza).

Spis treści – powinien być generowany automatycznie, z podaniem tytułów rozdziałów i podrozdziałów wraz numerami stron. Typ czcionki oraz wielkość liter spisu treści powinny być takie same jak w niniejszym wzorcu. Powinna też być zachowana zadeklarowana głębokość numeracji (patrz ustawienia w kodzie źródłowym szablonu poprzedzone komentarzem `% INFO: Deklaracja głębokości numeracji`).

Spis rysunków, Spis tabel, Spis listingów – powinny być generowane automatycznie (podobnie jak *Spis treści*). Elementy te są opcjonalne, nie zawsze trzeba je deklarować. Na przykład robienie osobnego spisu tabel zawierającego jedną lub dwie pozycje specjalnie nie ma sensu (taki spis raziłby pustą przestrzenią na stronie).

Skróty – jeśli w pracy występują liczne skróty, to w tej części należy podać ich rozwinięcia.

Wstęp – pierwszy rozdział, w którego pierwszej sekcji *Wprowadzenie* powinien znaleźć się opis dziedziny, w jakiej osadzona jest praca, oraz wyjaśnienie motywacji do podjęcia tematu. W drugiej sekcji *Cel i zakres* powinien znaleźć się opis celu oraz zadań do wykonania, zaś w trzeciej sekcji *Układ pracy* – opis zawartości kolejnych rozdziałów.

Rozdział – logicznie wyróżniona część pracy. Tytuły kolejnych rozdziałów mogą być różne w zależności od charakteru pracy. Inne będą dla pracy o charakterze analityczno-przeglądowym, inne dla pracy o charakterze projektowym. Niemniej zawartość i objętość rozdziałów powinny być dobrze wyważone. W pracy dyplomowej inżynierskiej kolejnymi rozdziałami mogą być: *Założenia projektowe*, *Analiza wymagań*, *Szczegóły implementacji*, *Testy*. W pracy dyplomowej magisterskiej mniej więcej jej połowę powinno poświęcić się na część badawczą (analityczną). Zwykle w tego typu pracy zaczyna się od przeglądu literatury (w wyniku czego powstać ma opis bieżących osiągnięć w danej dziedzinie, ang. *state of the art*). Potem sprecyzować można problemy badawcze, rozwiązywaniem których zajęto się w pracy. Kolejne rozdziały mogą dotyczyć opracowanego rozwiązania, przeprowadzonych eksperymentów i analiz ich wyników itd. Zwykle aspekt praktyczny (inżynierski) w pracy magisterskiej nie jest głównym celem, a raczej środkiem, z pomocą którego dąży się do zrealizowania celu badań (czyli podpiera się nim aspekt badawczy).

Podsumowanie – w rozdziale tym powinny być zamieszczone: podsumowanie uzyskanych efektów oraz wnioski końcowe wynikające z realizacji celu pracy dyplomowej.

Literatura – wykaz źródeł wykorzystanych w pracy (do każdego źródła musi istnieć odpowiednie cytowanie w tekście). Wykaz ten powinien być generowany automatycznie.

Dodatek – miejsce na informacje dodatkowe, jak: *Instrukcja wdrożeniowa*, *Instrukcja uruchomieniowa*, *Podręcznik użytkownika* itp. Osobny dodatek powinien być przeznaczony na opis zawartości dołączonej płyty CD/DVD. Założono, że będzie to zawsze ostatni dodatek.

Indeks rzeczowy – lista referencji do istotnych fraz, do których czytelnik będzie chciał sięgnąć. Indeks powinien być generowany automatycznie. Jego załączanie jest opcjonalne.

4.2. Styl

Pisząc tekst pracy dyplomowej należy zadbać o poprawność językową. Praca powinna być napisana językiem formalnym, w stylu akademickim. W liście wyliczeniowej zamieszczonej poniżej zebrano wybrane reguły odnoszące się do zalecanego stylu wypowiedzi. Lista ta nie jest zamknięta. Może ona być rozszerzona o kolejne zalecenia. (przy okazji proszę zauważyć, że tworząc tę listę wykorzystano mechanizm wyrównania wpisów zależnie od długości najdłuższej etykiety, szczegóły widać w kodzie źródłowym szablonu):

1. Praca dyplomowa powinna być napisana w formie bezosobowej. Co więcej, używane zwroty dobrze jest pisać w formie zwartej. Czyli zamiast „w pracy zostało pokazane ...” lepiej napisać „w pracy pokazano ...”. Wynikowy tekst będzie wtedy krótszy oraz czytelniejszy. Taki styl wypowiedzi przyjęto na uczelniach w naszym kraju, choć w krajach anglosaskich preferuje się redagowanie treści w pierwszej osobie.
2. W tekście pracy można odwołać się do myśli autora, ale nie w pierwszej osobie, tylko poprzez wyrażenia typu: „autor wykazał, że ...”.
3. Odwołując się do rysunków i tabel należy używać zwrotów typu: „na rysunku ... pokazano ...”, „w tabeli ... zamieszczono ...”. Tabela i rysunek to twory nieżywotne, więc nie mogą „pokazywać”. Dlatego też zwroty typu „rysunek pokazuje” uznawane są za niepoprawne.
4. Praca powinna być napisana językiem formalnym, bez wyrażeń żargonowych („sejwowanie” i „downloadowanie”), nieformalnych czy zbyt ozdobnych („najznamienitszym przykładem tego niebywałego postępu ...”)
5. Pisząc pracę należy dbać o poprawność stylistyczną wypowiedzi, a więc: trzeba pamiętać, do czego stosuje się „liczba”, a do czego „ilość”, zamiast „szereg elementów” powinno się pisać „wiele elementów”.
6. W tekstach dotyczących technologii informacyjnych często dochodzi do zapożyczeń z języka angielskiego. Zapożyczenia te mocno zakotwiczają się w języku, stając się częścią tzw. języka branżowego. I trudno zatrzymać ten proces, szczególnie gdy w języku polskim brakuje odpowiedników. Jednak gdy takie odpowiedniki istnieją, stosowanie zapożyczeń uważane jest za błąd. Podczas redakcji pracy w języku polskim należy unikać kalek językowych. Przykładem niepoprawnie stosowanej kalki językowej jest termin „funkcjonalność”. Zaczęto go używać w odniesieniu do rzeczy policzalnych (mówiąc o „wielu funkcjonalnościach” czy też o „wybranej funkcjonalności”) podczas gdy według oficjalnych norm językowych „funkcjonalność” odnosi się do rzeczy, które „dobrze spełniający swoją funkcję” (coś jest „funkcjonalne” jeśli dobrze działa, w przeciwieństwie do czegoś „niefunkcjonalnego”, czego nie da się używać). Mówiąc więc o cechach systemów informatycznych powinno się robić odniesienia do ich „funkcji”, a nie „funkcjonalności”.
7. Redagowane zdania nie powinny być zbyt długie, by czytelnik był w stanie je zapamiętać i zrozumieć. Dlatego lepiej podzielić zdanie wielokrotnie złożone na pojedyncze zdania.
8. Zawartość rozdziałów powinna być dobrze wyważona. Nie wolno więc generować sekcji i podsekcji, które mają zbyt mało tekstu lub znacząco różnią się objętością. Zbyt krótkie podrozdziały można zaobserwować w przykładowym rozdziale 6.
9. Niedopuszczalne jest pozostawienie w pracy błędów ortograficznych czy tzw. literówek – można je przecież znaleźć i skorygować automatycznie.
10. Tutaj pokazano, jak można zarządzać marginesem w otoczeniu `enumerate` (proszę zajrzeć do źródeł latexowego kodu tego rozdziału).

4.3. Prowadzenie badań

Od prac magisterskich oczekuje się wyraźnie wyróżnionego aspektu badawczego. Dlatego podczas redakcji pracy należy zadbać o wykorzystanie odpowiedniego warsztatu naukowego. Poniżej zamieszczono referencje do wybranych materiałów pomocniczych odnoszących się do tej kwestii.

Stawianie hipotez badawczych – krótkie wprowadzenia do pojęcia hipotezy badawczej i problemów badawczych można znaleźć pod adresami: `https://panstatystyk.pl/2022/10/11/ogarniamy_hipotezy_i_pytania_badawcze_w_pracy_magisterskiej/`, `https://obliczeniastatystyczne.pl/hipoteza-badawcza/`. Godny szczególnego polecenia, gdyż

podejmuje się temat hipotez badawczych dla badań eksperymentalnych oraz teoretycznych z punktu widzenia informatyki, jest materiał: https://troja.uksw.edu.pl/zasoby/praca-dyplomowa.pdf

Opracowywanie uzyskanych wyników – podczas prowadzenia badań ważne jest zapewnienie ich obiektywności. Na przykład podczas porównywania działania różnych algorytmów (badań eksperymentalnych) dobrze jest stosować uznane metody statystyczne i metryki. Przykładowe metryki zbierane podczas porównywania algorytmów ewolucyjnych oraz sposoby statystycznego opracowywania uzyskanych wyników znaleźć można w prezentacji Eksperymenty obliczeniowe z algorytmami ewolucyjnymi i porównania algorytmów (http://aragorn.pb.bialystok.pl/~wkwedlo/EA9.pdf).

Ciekawym materiałem zawierającym ogólne wprowadzenie do testów statystycznych jest: https://panstatystyk.pl/2022/10/27/jaki-test-statystyczny-wybrac-cz-i-czyli-ogarniamy-algorytm-wyboru-testu-statystycznego/. Warto też spojrzeć na darmowe narzędzie do obliczania parametrów wybranych rozkładów: EasyFit (https://easyfit.soft32.com/).

W celu wielokryterialnego porównywania rozwiązań systemów informatycznych można zastosować: *Analityczny Proces Hierarchiczny*, *Wnioskowanie rozmyte*, inne metody *Wielokryterialnego podejmowania decyzji*. Więcej materiałów o statystyczny opracowaniu wyników można znaleźć w literaturze (lub w Internecie) pod hasłem testowanie hipotez statystycznych.

Rozdział 5

Uwagi techniczne

5.1. Zasady redakcji

W pracy należy dbać o poprawność redakcyjną zgodnie z zaleceniami:

- nie zostawiać znaku spacji przed znakami interpunkcji (zamiast „powiedziano , że ...” powinno być „powiedziano, że ...”),
- kropki po skrótach, które nie są jednocześnie kropkami kończącymi zdanie sklejać z kolejnym wyrazem znakiem tyldy, np. jak tutaj (`np.~jak tutaj`) lub wstawiać za nimi ukośnik, np. jak tutaj (`np.\ jak tutaj`),
- nie zapominać o formatowaniu wyliczenia (należy zaczynać małymi literami lub dużymi oraz kończyć przecinkami, średnikami i kropkami – w zależności od kontekstu danego wyliczenia),
- nie zostawiać samotnych literek na końcach linii (można je „skleić” z wyrazem następnym stosując znaczek tyldy, jak `w~przykładzie`),
- nie zostawiać pojedynczych wierszy na końcu lub początku strony (należy kontrolować „sieroty” i „wdowy”),
- nie zostawiać odstępu pomiędzy tekstem a nawiasami czy znakami cudzysłowów (znaki te powinny przylegać do tekstu, który obejmują „jak w tym przykładzie”),
- wyrazy obcojęzyczne powinny być pisane czcionką pochyłą (preferowane `\emph{}`) wraz ze skrótem oznaczającym język, w szczególności ma to zastosowanie przy rozwijaniu skrótów, np. OGC (ang. *Open Geospatial Consortium*) (w kodzie latexowym wygląda to tak: `np.~OGC (ang.~\emph{Open Geospatial Consortium})`),
- każdy zastosowany skrót powinien zostać rozwinięty podczas pierwszego użycia, później może już występować bez rozwinięcia (skrót i jego rozwinięcie powinny trafić również do wykazu *Skróty*, jeśli taki wykaz jest dołączany do dokumentu),
- nie wolno zostawiać zbyt dużo białej przestrzeni bez żadnego uzasadnienia.

Odnosząc się do ostatniej zasady, to przekłada się ona na gospodarowanie białą przestrzenią. Poniżej przedstawiono przykład złego użycia listy wyliczeniowej. Jeśli bowiem elementy na liście są reprezentowane przez krótki tekst, to wtedy powstaje biała plama. Widać to szczególnie przy długich listach.

Moje ulubione kolory to:

- biały,
- niebieski,
- czerwony.

Dużo lepiej w takim przypadku albo po prostu wyliczyć wartości po przecinkach:

Moje ulubione kolory to: biały, niebieski, czerwony.

albo wypisać listę w kilku kolumnach:

Moje ulubione kolory to:

- biały,
- niebieski,
- czerwony.

5.2. Rysunki

W niniejszym szablonie numeracja rysunków odbywa się automatycznie według następujących reguł: rysunki powinny mieć numerację ciągłą w obrębie danego rozdziału, sam zaś numer powinien składać się z dwóch liczb rozdzielonych kropką. Pierwsza liczbą ma być numer rozdziału, drugą – kolejny numer rysunku w rozdziale. Przykładowo: pierwszy rysunek w rozdziale 1 powinien mieć numer 1.1, drugi – numer 1.2 itd., pierwszy rysunek w rozdziale 2 powinien mieć numer 2.1, drugi – numer 1.2 itd. Wszystko to załatwia szablon.

Rysunki powinny być wyśrodkowane na stronie wraz z podpisem umieszczonym na dole. Podpisy nie powinny kończyć się kropką. Czcionka podpisu powinna być mniejsza od czcionki tekstu wiodącego o 1 lub 2 pkt (w szablonie jest to czcionka rozmiaru `small`). Ponadto należy zachowywać odpowiedni odstęp między rysunkiem, podpisem rysunku a tekstem rozdziału. Przykłady, jak to zrobić, zamieszczono w szablonie.

W przypadku korzystania z szablon odstępy te regulowane są automatycznie. Podpis i grafika muszą stanowić jeden obiekt. Chodzi o to, że w edytorach tekstu typu Office podpis nie scala się z grafiką i czasem trafia na następną stronę, osieracając grafikę. Korzystającym z niniejszego szablonu i otoczenia `\figure` takie osierocenie nigdy się nie zdarzy.

Do każdego rysunku musi istnieć odwołanie w tekście (inaczej mówiąc: niedopuszczalne jest wstawienie do pracy rysunku bez opisu). Odwołania do rysunków powinny mieć postać: „Na rysunku 3.3 przedstawiono...” lub „... co ujęto na odpowiednim schemacie (rys. 1.7)”.

Jeśli odwołanie stanowi część zdania, to wtedy wyraz „rysunek” powinien pojawić się w całości. Jeśli zaś odwołanie jest ujęte w nawias (jak w przykładzie), wtedy należy zastosować skrót „rys.”. Jeśli do stworzenia obrazka wykorzystano jakieś źródła, to powinny one być zacytowane w podpisie tegoż rysunku.

Należy pamiętać o tym, że „rysunki” to twory nieżywotne. W związku z tym nie mogą ”pokazywać”. Dlatego „rysunek 1.1 pokazuje ...” jest stylistycznie niepoprawne. Zamiast tego zwrotu trzeba użyć „ na rysunku 1.1 pokazano ...”.

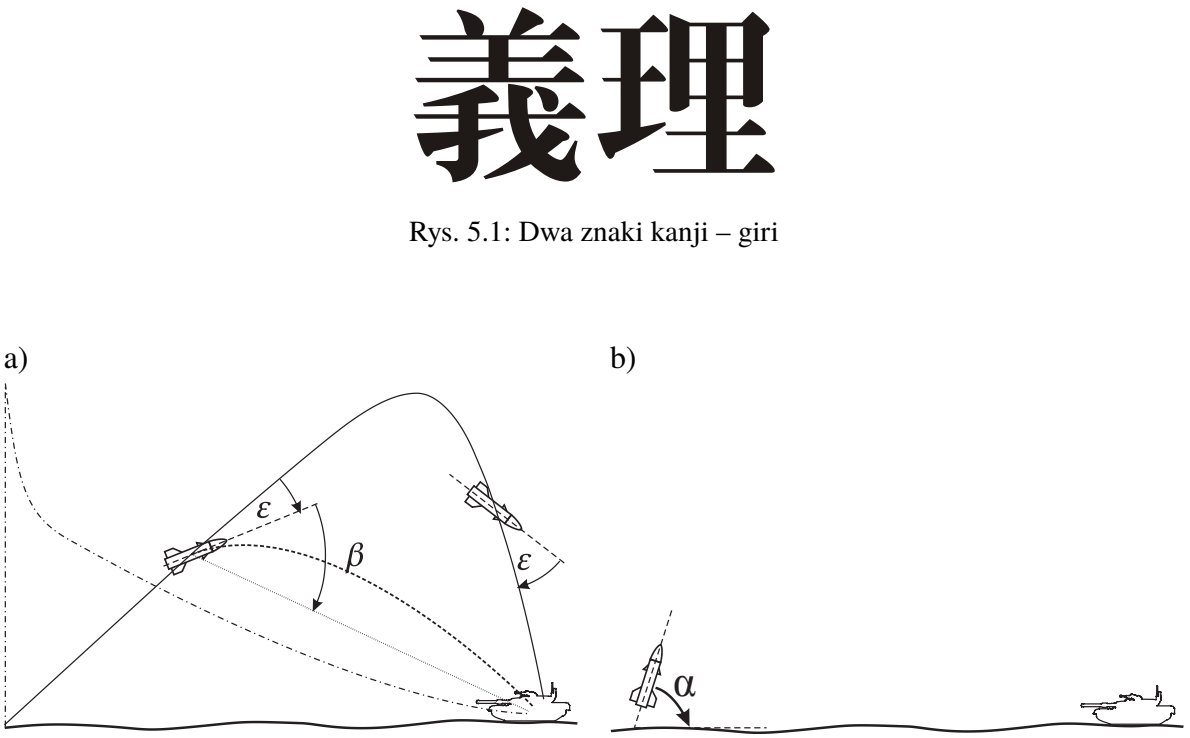
Rysunki można wstawiać do pracy używając polecenia `\includegraphics`. Zalecane jest, aby pliki z grafikami były umieszczane w katalogach odpowiadających numerom rozdziałów czy literom dodatków: `rys01`, `rysA` itd. Sposób wstawiania rysunków do pracy zademonstrowano na przykładze rysunków 5.1 i 5.2.

Listing 5.1: Kod źródłowy przykładów wstawiania rysunków do pracy

```
\begin{figure}[ht]
 \centering
  \includegraphics[width=0.3\linewidth]{rys05/kanji-giri}
 \caption{Dwa znaki kanji - giri}
 \label{fig:kanji-giri}
\end{figure}

\begin{figure}[htb]
 \centering
  \begin{tabular}{@{}ll@{}}
  a) & b) \\
  \includegraphics[width=0.475\textwidth]{rys05/alfa1} &
  \includegraphics[width=0.475\textwidth]{rys05/beta1}
```

```
        % jeśli obraki są różnej wysokości, można je wyrównać do góry
            ↪ stosując vtop jak niżej
        % \vtop{\vskip-2ex\hbox{{\includegraphics[width=0.475\textwidth]{
            ↪ rys05/beta1}}}} &
        % \vtop{\vskip-2ex\hbox{{\includegraphics[width=0.475\textwidth]{
            ↪ rys05/alfa1}}}}
  \end{tabular}
 a) trzy podejścia, b) podejście praktyczne}
 \label{fig:alfabeta}
\end{figure}
```



Rys. 5.1: Dwa znaki kanji – giri

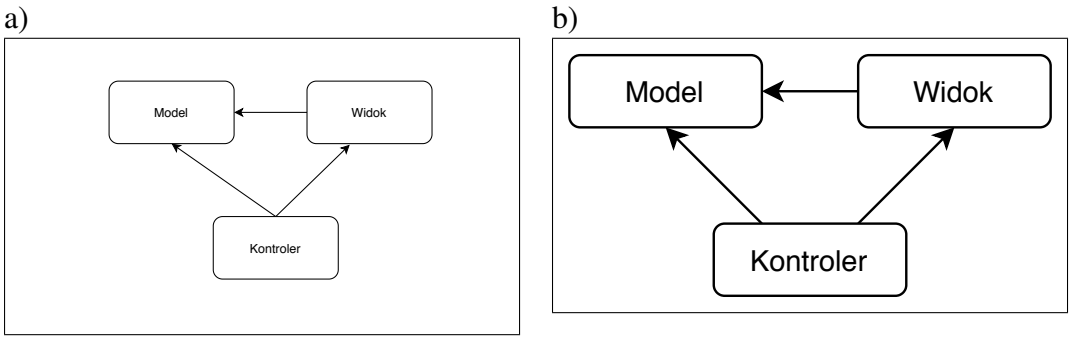
Rys. 5.2: Wyznaczanie trajektorii lotu rakiety: a) trzy podejścia, b) podejście praktyczne

Grafiki wektorowe powinny być dostarczone w plikach pdf. Rozmiar strony w pliku pdf powinien być równy lub minimalnie większy od rozmiaru znajdującej się na nim grafiki (proszę spojrzeć na przykłady grafik wykorzystanych w niniejszym szablonie). Chodzi o to, aby na rysunku nie pojawiała się niepotrzebna biała przestrzeń (rozmiar płótna ma odpowiadać rozmiarowi grafiki bez żadnych marginesów, elementy grafiki powinny być ciasno ułożone). Grafiki rastrowe (głównie zrzuty z ekranu bądź zdjęcia) powinny być dostarczane w plikach o formacie `png` z kompresją bezstratną. Zastosowanie kompresji stratnej, jak `jpg`, wprowadza niepotrzebne artefakty. Podobnie jak w przypadku grafik wektorowych, grafiki rastrowe nie powinny mieć białych marginesów.

Niezłym sposobem generowanie ładnej grafiki wektorowej, w szczególności diagramów, jest posłużenie się, kolejno, następującymi darmowymi narzędziami:

- w `diagrams.net` (dawniej `draw.io`): narysowanie diagramu i wyeksportowanie do pdf (bezpośrednio lub pośrednio, poprzez zapisanie pliku w formacie `svg`, a potem jego wyświetlenie w przeglądarce internetowej i wydrukowanie do `pdf`),
- w `inkscape`: zaimportowanie `pdf`, rozdzielenie grupy, wykasowanie niepotrzebnych elementów (tła), zaznaczenie wszystkiego, przycięcie strony do zaznaczonych (Ctr-Shift-R), zapisanie jako `pdf`.

Na rysunku 5.3 pokazano przykład dobrze i źle (od strony technicznej) narysowanego diagramu. Celowo pokazano ramki, by było widać marginesy. Normalnie ramek tych nie należy stosować.



Rys. 5.3: Przykład diagramu: a) złego, b) w miarę dobrego

Elementy na rysunkach nie powinny być wypełnione 100% czernią ponieważ na wydrukach tworzą się plamy przebijające się przez kartkę. Zamiast tego wypełnienie elementów powinno być ustawione na ok. 90% czerni.

Czcionka na rysunkach nie może być większa od czcionki wiodącej tekstu (jedyny wyjątek to np. jakieś nagłówki). Należy stosować czcionkę kroju Arial, Helvetica bądź tego samego kroju co czcionka dokumentu (`texgyre-termes`).

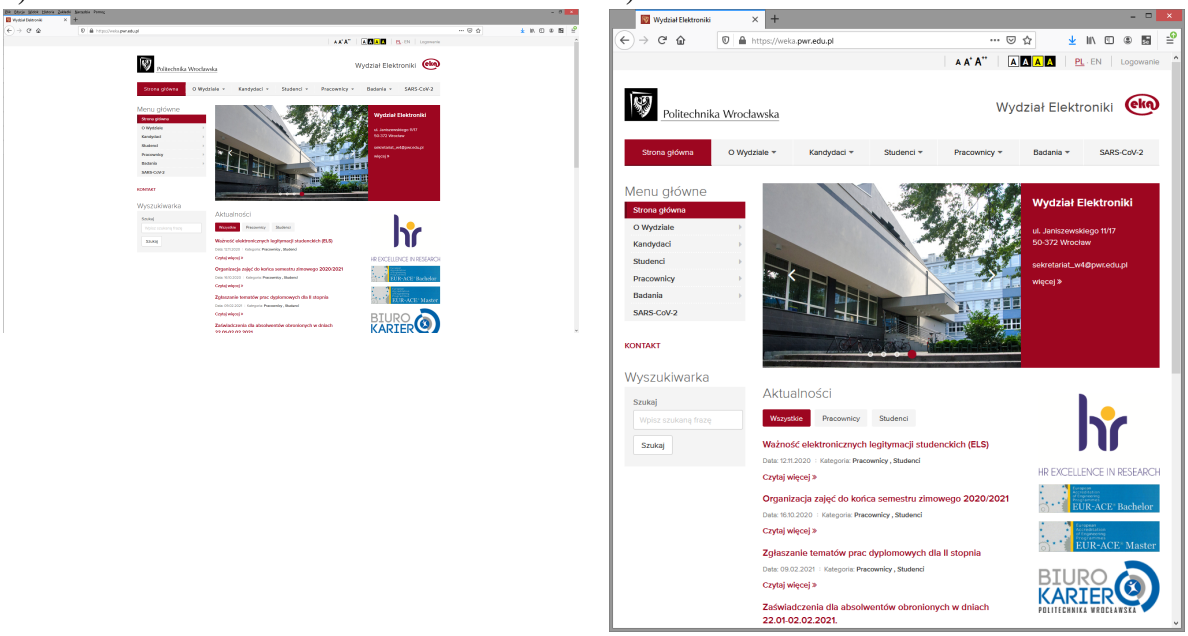
Jeśli na jednym rysunku pojawić się ma kilka grafik, to zamiast stosować `subfigure` lub inne otoczenia należy: dostarczyć tabelę z wstawionymi do niej rysunkami, opcjonalnie adnotować jej części (np. a) i b)), odnieść się do tych części w podpisie (posługując się adnotacjami, jak to zrobiono na rysunkach 5.2 i 5.3, lub opisem słownym, np. „z lewej strony pokazano ...", „po prawej zamieszczono ...").

Jeśli na rysunku zamieszczono w tabeli kilka grafik, to ich pozycjonowaniem (wyjustowaniem od góry) można manipulować za pomocą komendy: `\vtop{\vskip-2ex\hbox{\includegraphics[width=0.475\textwidth]{nazwa}}}`

Na rysunkach nie wolno nadużywać kolorów oraz ozdobników (wiele narzędzi do tworzenia diagramów dostarcza grafikę z cieniowaniem, gradacją kolorów itp. co niekoniecznie przekłada się na czytelność rysunku). Jeśli rysunki są kolorowe, to kolory te powinny być rozróżnialne po konwersji do poziomów szarości (chodzi o to, aby na wydrukach wykonanych na drukarkach monochromatycznych można było dostrzec różnice).

Podczas robienia zrzutów z ekranu należy zadbać o to, by taki zrzut był czytelny po wydrukowaniu. Czyli aby pojawiające się literki były wystarczająco duże, a przestrzenie bez treści – relatywnie małe. Przystępując do robienia zrzutu trzeba odpowiednio wyskalować elementy na ekranie. Na przykład robiąc zrzut z przeglądarki FF najpierw należy wcisnąć CTR–0 (domyślne skalowanie), potem CTR–- (zmniejszenie skali o stopień). Potem dobrze jest zawęzić okno przeglądarki tak, by interesująca treść wypełniła je w całości. Jeśli na obserwowanej stronie jest zbyt dużo pustych obszarów, to należy je jakoś zawęzić (sterując wielkością okna przeglądarki lub aktywnymi elementami interfejsu użytkownika). Zrzut bowiem wcale nie musi być odzwierciedleniem 1:1 domyślnego układu obserwowanych elementów. Ważne jest, by na zrzucie pokazać interesujący, opisywany fragment i żeby ten fragment był czytelny. Nie trzeba też zawsze robić zrzutów w układzie 16:9 (lub innym panoramicznym). Czasem lepiej jest zrobić zrzuty okien niemal kwadratowych, bo lepiej się układają w wynikowym dokumencie. Poza tym można je przeskalować (powiększyć) by zajmowały całą szerokość strony, a wtedy czcionka na wydrukach będzie większa. Takie skalowanie dla zrzutów panoramicznych zwykle się nie udaje. Na rysunku 5.4 pokazano przykłady dobrze i źle zrobionych zrzutów (w celu oszczędzenia miejsca zrzuty umieszczono obok siebie).

a) b)



Rys. 5.4: Przykłady zrzutów z ekranu: a) zły (nieczytelny, zrobiony przy zbyt szerokim oknie, z niepotrzebnymi marginesami, niepotrzebnym paskiem menu), b) w miarę dobry (w miarę czytelny, zrobiony przy zawężonym oknie, byłoby wskazane jeszcze usunięcie z niego beleczki z faviconem (jeśli nic nie wnosi) oraz przycięcie od dołu (jeśli treści tam pokazywane nie są istotne))

Czasem problemem jest tworzenie zrzutów z ekranu, gdy występują na nim dane wrażliwe. Istnieją dwa sposoby na radzenie sobie z tym problemem. Pierwszy polega na zastąpieniu w systemie danych danych rzeczywistych danymi testowymi – wygenerowanymi tylko do celów prezentacji. Zrzut robi się wtedy na bazie danych testowych. Drugi polega na wykonaniu zrzutu z ekranu, na którym pokazano dane rzeczywiste, i następnie zamianie tych danych już w pliku graficznym za pomocą odpowiedniego edytora (np. `gimp`). Czyli oryginalny zrzut z ekranu należy otworzyć w edytorze, a potem nadpisać oryginalny tekst własnym tekstem. Konieczne jest wtedy dobranie odpowiednich czcionek aby nie było widać wprowadzonych zmian.

> Uwaga: takie manipulowanie zrzutami jest usprawiedliwione jedynie w przypadku konieczności ochrony danych wrażliwych czy też lepszego pokazania wybranych elementów. Nie może to prowadzić generowania fałszywych rezultatów!!!

5.3. Wstawianie kodu źródłowego

Kod źródłowy można wstawiać jako blok tekstu pisany czcionką maszynową. Używa się do tego otoczenie `\lstlisting`. W atrybutach otoczenia można zdefiniować tekst podpisu wstawianego wraz z numerem nad blokiem, etykietę do tworzenia odwołań, sposób formatowania i inne ustawienia. Zaleca się stosowanie w tym otoczeniu następujących parametrów:

```
\begin{lstlisting}[label=list:req1,caption=Initial HTTP Request,
                   basicstyle=\footnotesize\ttfamily]
```

Szczególnie przydatne podczas wstawiania większej ilości kodu źródłowego jest zastosowanie parametru `basicstyle=\footnotesize\ttfamily`. Dzięki niemu zmniejsza się czcionka, a przez to na stronie można zmieścić dłuższe linijki kodu. Użycie tak zdefiniowanego parametru nie jest jednak sztywnym zaleceniem. Wielkość czcionki można dobierać do potrzeb.

Listing 5.2: Initial HTTP Request

```
GET /script/Articles/Latest.aspx HTTP/1.1
Host: www.codeproject.com
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml
User-Agent: Mozilla/5.0 ...
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US...
Accept-Charset: windows-1251,utf-8...
```

Można też sformatować kod bez stosowania numerowanego podpisu (wtedy nie zamieszcza się `caption` na liście atrybutów).

```
GET /script/Articles/Latest.aspx HTTP/1.1
Host: www.codeproject.com
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml
User-Agent: Mozilla/5.0 ...
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US...
Accept-Charset: windows-1251,utf-8...
```

Ponadto istnieje kilka sposobów wstawiania kodu źródłowego w bieżącej linijce tekstu:

- korzystając z polecenia `\texttt` ustawiającego czcionkę maszynową, jak w przykładzie `tutaj` (efekt zastosowania komendy `\texttt{tutaj}`). Problemem jednak mogą okazać się znaki podkreślenia i inne znaki kontrolne.
- korzystają z otoczenia `\verb` zapewniającego wypisanie kodu czcionką maszynową jak w przykładzie `tutaj` (efekt zastosowania komendy `\verb|tutaj|`). Problemem jest to, że polecenie `\verb` nie potrafi łamać dłuższego tekstu.
- korzystając z polecenia `\lstin` umożliwiającego wypisanie kodu czcionką ustawianą w opcjach jak w przykładzie `tutaj` (efekt komendy `\lstset{basicstyle=\ttfamily}\lstinline{tutaj}`) lub `tutaj` (efekt komendy `\lstinline[basicstyle=\ttfamily]=tutaj=`).

Poniżej zamieszczono przykłady kodów źródłowych z podświetleniem składni.

Listing 5.3: Opis 1

```
package pl.mrbarozoit.backend;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class BackendApplication {

    public static void main(String[] args) {
        SpringApplication.run(BackendApplication.class, args);
    }

}
```

Jeśli w kodzie źródłowym jest jakiś nieistotny fragment względem omawianego problemu, to można go wykropkować (patrz listing 5.4).

Listing 5.4: Opis 2

```
// Karma configuration file, see link for more information
// https://karma-runner.github.io/1.0/config/configuration-file.html
```

```
module.exports = function (config) {
  config.set({
    basePath: '',
    frameworks: ['jasmine', '@angular-devkit/build-angular'],
    plugins: [
      require('karma-jasmine'),
      require('karma-chrome-launcher'),
      require('karma-jasmine-html-reporter'),
      require('karma-coverage-istanbul-reporter'),
      require('@angular-devkit/build-angular/plugins/karma')
    ],
    ... // opuszczony kod
    autoWatch: true,
    browsers: ['Chrome'],
    singleRun: false,
    restartOnFileChange: true
  });
};
```

Jeśli kod jest wąski, to można sformatować go w dwóch kolumnach jak na listingu 5.5.

Listing 5.5: Kontrakt na model wejściowy endpointu `/api/v1/chess/board/move`

```
{
   "lastPosition": {
      "fenDescription": "string"
   },
   "image": {
      ...
   },
   "positions": {
      "chessboardCorners": [
         ...
      ],
      "tilesCornerPoints": [
         ...
      ]
   },
   "referenceColors": {
      ...
   },
}
```

5.4. Wykaz literatury oraz cytowania

Cytowania powinny być zamieszczane w tekście z użyciem komendy `\cite{}`. Jej argumentem powinien być klucz cytowanej pozycji (lub lista kluczy rozdzielonych przecinkiem bez spacji, jeśli takich pozycji w danym miejscu cytuje się więcej) jaki jest używany w bazie danych bibliograficznych (plik `dokumentacja.bib`). Po kompilacji `bibtex` i `pdflatex` w tekście pojawia się właściwy odsyłacz do pozycji w wykazie literatury (ujęty w kwadratowe nawiasy – zgodnie z tym, co definiuje styl `plabbrv.bst`), zaś w samym wykazie (rozdział Literatura) – zacytowana pozycja. Przykładem cytowania jest: „dobrze to opisano w pracach [**?**, **?**]” (gdzie zastosowano komendę `\cite{JS07,SQL2}`).

Co do zawartości rekordów bibliograficznych - style bibtexowe potrafią „skracać” imiona (czyli wstawiać, jeśli taka wola, inicjały zamiast pełnych imion). Niemniej dobrze jest od razu przyjąć jakąś konwencję. Proponuje się, aby w rekordach od razu wstawiane były inicjały zamiast pełnych imion.

Niekiedy tytuły prac zawierają wyrazy z dużymi i małymi literami. Takie tytuły należy brać w podwójne nawiasy klamrowe, aby `bibtex` nie zamienił ich na postać, w której poza pierwszą literą pozostałe są małe.

Jeśli jakiś cytowany zasób pochodzi z Internetu, to jego rekord w pliku `bib` powinien wyglądać jak niżej.

```
@INPROCEEDINGS{SQL2,
  title={{A MySQL-based data archiver: preliminary results}},
```

```
  author={Bickley, M. and Slominski, Ch.},
  booktitle = {{Proceedings of ICALEPCS07}},
        month = oct,
        day = {15--19},
        year={2007},
  note={\url{http://www.osti.gov/scitech/servlets/purl/922267}
        [dostęp dnia 20 czerwca 2015]}
}
```

A to inny przykład rekordu danych bibliograficznych:

```
@TechReport{JS07,
        author = {Jędrzejczyk, J. and Śródka, B.},
        title  ={Segmentacja obrazów metodą drzew decyzyjnych},
        year = {2007},
        institution = {Politechnika Wrocławska, Wydział Elektroniki}
}
```

5.5. Indeks rzeczowy

Generowanie indeksu po trosze wygląda jak generowanie wykazu literatury – wymaga kilku kroków. Podczas pierwszej kompilacji `pdflatex` generowany jest plik z rozszerzeniem `*.idx` (zawierający „surowy indeks"). Następnie, bazując na tym pliku, generowany jest plik z rozszerzeniem `*.ind` zawierający sformatowane dane. Ten krok wymaga uruchomienia odpowiedniego narzędzia oraz zastosowania plik z definicją stylu `Dyplom.ist`. W kroku ostatnim dokonuje się kolejnej kompilacji `pdflatex` (dzięki niej w wynikowym dokumencie pojawi się Indeks rzeczowy). Domyślnie Indeks rzeczowy zostanie sformatowany w układzie dwukolumnowym.

Oczywiście aby to wszystko zadziałało w kodzie szablonu należy umieścić odpowiednie komendy definiujące elementy indeksu rzeczowego (`\index`) oraz wstawiające sformatowany Indeks rzeczowy do dokumentu wynikowego (`\printindex`). Więcej informacji o tworzeniu indeksu rzeczowego można znaleźć na stronie `https://en.wikibooks.org/wiki/LaTeX/Indexing`. Poniżej przedstawiono przykłady komend użytych w szablonie do zdefiniowania elementów indeksu rzeczowego:

- `\index{linia komend}` – pozycji główna.
- `\index{generowanie!-- indeksu}` – podpozycja.

Generowanie pliku `*.ind` można inicjować na kilka sposobów:

- poprzez wydanie odpowiedniego polecenia bezpośrednio w linii komend

  ```
  makeindex Dyplom.idx -t Dyplom.ilg -o Dyplom.ind -s Dyplom.ist
  ```

- poprzez odpalenie odpowiedniego narzędzia środowiska. Na przykład w `TeXnicCenter` definiuje się tzw. `output profiles`:

  ```
  makeindex "%tm.idx" -t "%tm.ilg" -o "%tm.ind" -s "%tm.ist"
  ```

  a samo generowanie pliku `*.ind` zapewni wybranie pozycji menu `Build/Makeindex`.
- korzystając z odpowiednio sparametryzowanych pakietów i komend wewnątrz kompilowanego dokumentu (czyli od razu przy okazji jego kompilacji).

  ```
  \DisemulatePackage{imakeidx}
  \usepackage[noautomatic]{imakeidx}
  % jeśli chcemy, by indeks by generowany automatycznie programem makeindex
      ↪ :
  %\usepackage[makeindex]{imakeidx}
  % a tak ponoć można przekazać opcje do programu generującego indeks:
  %\makeindex[options=-s podrecznik -L polish -M lang/polish/utf8]
  ```

```
%\makeindex[options=-s podrecznik]
\makeindex
```

Niestety, `makeindex` jest narzędziem, które umieszcza część pozycji w grupie `Symbols`, a nie w grupach związanych z literkami alfabetu. W związku z czym indeksowany element zaczynający się od polskiej literki trafia do grupy `Symbols`, jak np. `\index{Światło}`. Jeśli chce się zamieszczać w indeksie symbole matematyczne, to dobrze jest to robić jak w następującym przykładzie: `\index{$asterisk@$\ast$}` czy też `\index{c@$\mathcal{C}$}`, tj. dostarczając przy okazji klucz do sortowania. Lepiej w tym względzie radzą sobie inne narzędzia, jak `texindy` lub `xindy` dostępne pod linuxem. Korzystając z nich uzyskuje się grupy polskich literek w indeksie rzeczowym (hasła zaczynające się od polskich literek już nie trafiają do grupy Symbols). Przykład polecenia wydanego z linii komend, w którym wykorzystano `texindy` zamieszczono poniżej (zakładamy kodowanie plików w UTF8, można dla niniejszego szablonu zmienić na cp1250):

```
texindy -L polish -M lang/polish/utf8 Dyplom.idx
```

To polecenie wygeneruje `Dyplom.ind` o zawartości:

```
\begin{theindex}
  \providecommand*\lettergroupDefault[1]{}
  \providecommand*\lettergroup[1]{%
      \par\textbf{#1}\par
      \nopagebreak
  }

  \lettergroup{G}
  \item generowanie
    \subitem -- indeksu, 27
    \subitem -- wykazu literatury, 27

  \indexspace

  \lettergroup{L}
  \item linia komend, 27

  \indexspace

  \lettergroup{Ś}
  \item \'Swiat\IeC {\l }o, 28

\end{theindex}
```

Aby mieć większą kontrolę automatyczne generowanie indeksu zostało w niniejszym szablonie wyłączone (indeks trzeba wygenerować samemu, wydając polecenie `makeindex` lub zalecane `texindy`).

5.6. Inne uwagi

Dobrym sposobem na kontrolę błędów występujących podczas kompilacji jest wstawianie linijki `\end{document}` w wybranym miejscu dokumentu. Jest to szczególnie przydatne w przypadkach, gdy błędy te są trudne do zidentyfikowania (gdy wygenerowane przez kompilator numery linii z błędami nie są tymi, w których błędy występują). Wystarczy wtedy przestawić wspomnianą linijkę do kolejnych miejsc, aż znajduję to miejsce, gdzie występuje problem.

Aby osiągnąć apostrofy maszynowe (złożone z samych kresek) należy użyć polecenia `"{}jak tutaj{}"` (podwójny apostrof stojący bezpośrednio przed niektórymi literkami zamienia je na literki z akcentami, aby temu zapobiec dostawiono nawiasy klamrowe). W efekcie

otrzymamy "jak tutaj". Jeśli natomiast apostrofy mają być drukarskie (złożone z kropek i kresek), to należy użyć polecenia `,,jak tutaj''` (dwa pojedyncze przecinki i dwa pojedyncze apostrofy). W efekcie otrzymamy „jak tutaj”. Można też użyć znaków apostrofów odpowiednio zakodowanych „jak tutaj”, tylko że czasem trudno pisze się takie apostrofy w środowiskach kompilacji projektów latexowych.

Oto sposoby ustawienia odstępów między liniami:

- używając komendy `\linespread{...}` (akceptowalne), przy czym atrybutem tej metody jest współczynnik zależny od wielkości czcionki. Dla czcionki wiodącej 12pt odstęp półtora linii osiągnie się komendą `\linespread{1.241}`. Dla innych czcionek wiodących wartości tego parametru są jak w poniższym zestawieniu.

  ```
  10pt 1.25 dla \onehalfspacing
       1.667 for \doublespacing,
                   ponieważ ,,basic ratio'' = 1.2
                  (\normalfont posiada \baselineskip rozmiaru 12pt)
  11pt 1.213 dla \onehalfspacing oraz 1.618 dla \doublespacing,
       ponieważ ,,basic ratio'' = 1.236
                  (\normalfont posiada \baselineskip rozmiaru 13.6pt)
  12pt 1.241 dla \onehalfspacing oraz 1.655 dla \doublespacing,
       ponieważsince ''basic ratio'' is 1.208
                  (\normalfont has a \baselineskip of 14.5pt)
  ```

  Kłopot w tym, że raz ustawiony odstęp będzie obowiązywał do wszystkich czcionek (brak jest mechanizmu zmiany współczynnika w zależności od wielkości czcionki akapitu).
- używając pakietu `setspace` (niezalecane). Ponieważ klasa `memoir` emuluje pakiet `setspace`, w preambule dokumentu należałoby umieścić:

  ```
  \DisemulatePackage{setspace}
  \usepackage{setspace}
  ```

  a potem można już sterować odstęp komendami:

  ```
  \singlespacing
  \onehalfspacing
  \doubelspacing
  ```

  Ten sposób pozwala na korzystanie z mechanizmu automatycznej zmiany odległości linii w zależności od wielkości czcionki danego akapitu.
- korzystając bezpośrednio z komend dostarczonych w klasie `memoir` (zalecane):

  ```
  \SingleSpacing
  \OnehalfSpacing
  \DoubleSpacing
  ```

  Ten sposób również pozwala na korzystanie z mechanizmu automatycznej zmiany odległości linii w zależności od wielkości czcionki danego akapitu.

Na koniec jeszcze uwaga o rozmiarze pliku wynikowego. Otóż `pdflatex` generuje pliki `pdf`, które zazwyczaj mogłyby być nieco lepiej skompresowane. Do lepszego skompresowania tych plików można użyć programu `ghostscript`. Wystarczy w tym celu wydać komendę (pod windowsami):

```
gswin64 -sDEVICE=pdfwrite -dCompatibilityLevel=1.4 -dNOPAUSE -dQUIET \
-dSAFER -dBATCH -sOutputFile=Dyplom-compressed.pdf Dyplom.pdf
```

W poleceniu tym można również wstawić opcję `-dPDFSETTINGS=/prepress` (zapewniającą uzyskanie wysokiej jakości, zachowanie kolorów, uzyskanie obrazków w rozdzielczości 300 dpi). Ze względów licencyjnych ghostscript używa domyślnie algorytmów z kompresją stratną. Przy kompresji może więc dojść do utraty jakości bitmap.

Rozdział 6

Podsumowanie

Podsumowanie jest miejscem, w którym należy zamieścić syntetyczny opis tego, o czym jest dokument. W szczególności w pracach dyplomowych w podsumowaniu powinno znaleźć się jawnie podane stwierdzenie dotyczące stopnia realizacji celu. Czyli powinny pojawić się w niej akapity ze zdaniami typu: „Podczas realizacji pracy udało się zrealizować wszystkie postawione cele". Ponadto powinna pojawić się dyskusja na temat napotkanych przeszkód i sposobów ich pokonania, perspektyw dalszego rozwoju, możliwych zastosowań wyników pracy itp.

6.1. Sekcja poziomu 1

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Nam id nulla a adipiscing tortor, dictum ut, lobortis urna. Donec non dui. Cras tempus orci ipsum, molestie quis, lacinia varius nunc, rhoncus purus, consectetuer congue risus.

6.1.1. Sekcja poziomu 2

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Sekcja poziomu 3

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Paragraf 4 Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

6.2. Sekcja poziomu 1

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Literatura

[1] N. Collins. The analysis of generative music programs. *Organised Sound*, 13(3):237–248, 2008.

[2] S. Forsgren, H. Martiros. Riffusion - Stable diffusion for real-time music generation. 2022.

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer. High-resolution image synthesis with latent diffusion models, 2021.

Dodatek A

Instrukcja wdrożeniowa

Jeśli praca skończyła się wykonaniem jakiegoś oprogramowania, to w dodatku powinna pojawić się instrukcja wdrożeniowa (o tym jak skompilować/zainstalować to oprogramowanie). Przydałoby się również krótkie „*how to*" (jak uruchomić system i coś w nim zrobić – zademonstrowane na jakimś najprostszym przypadku użycia). Można z tego zrobić osobny dodatek.

Dodatek B

Opis załączonej płyty CD/DVD

Tutaj jest miejsce na zamieszczenie opisu zawartości załączonej płyty. Opis ten jest redagowany przed załadowaniem pracy do systemy APD USOS, a więc w chwili, gdy nieznana jest jeszcze nazwa, jaką system ten wygeneruje dla załadowanego pliku. Dlatego też redagując treść tego dodatku dobrze jest stosować ogólniki typu: „Na płycie zamieszczono dokument `pdf` z niniejszej tekstem pracy" – bez wskazywania nazwy tego pliku.

Dawniej obowiązywała reguła, by nazywać dokumenty według wzorca `W04_[nr albumu]_[rok kalendarzowy]_[rodzaj pracy]`, gdzie `rok kalendarzowy` odnosił się do roku realizacji kursu „Praca dyplomowa", a nie roku obrony. Przykładowo wzorzec nazwy dla pracy dyplomowej inżynierskiej w konkretnym przypadku wyglądał tak: `W04_123456_2015_praca inżynierska.pdf`, Takie nazwy utrwalane były w systemie składania prac dyplomowych. Obecnie działa to już inaczej.