

UNIwersytet w Białymstoku
Wydział Matematyki i Informatyki
Instytut Informatyki

John OBMAR

Jak pisać pracę dyplomową? (v6)

Promotor:
dr inż. Mariusz RYBNIK

BIAŁYSTOK 2019

Spis treści

Wstęp	5
1 Dlaczego L^AT_EX?	7
1.1 Dlaczego L ^A T _E X, a nie <i>MS Word</i> ?	7
1.2 Czego używać do edycji kodu L ^A T _E X-a?	8
1.2.1 Polecany edytor online: www.overleaf.com	8
1.2.2 Edytor hardcore pod Windows: WinEdt	9
1.3 Jak pisać w L ^A T _E X-u?	10
2 Zalecenia edycyjne	11
2.1 Format fizyczny pracy oraz załączniki	11
2.2 Struktura pracy	11
2.2.1 Przykładowa subsection	12
2.3 Ogólne zalecenia pisarskie	12
2.4 Częste błędy gramatyczne	14
2.5 Korekta	15
3 Przykłady typowych czynności w L^AT_EX-u	17
3.1 Wypunktowania	17
3.2 Dołączanie rysunków	18
3.3 Przykład tabeli	19
3.4 Przykład formuły matematycznej	20
3.5 Przykład z kodem programu	20
3.6 Referowanie do literatury	20
3.7 Linki wewnętrzne i zewnętrzne	21
Podsumowanie	23
Bibliografia	25

Wstęp

Jest to dokument opisujący podstawowe zasady pisania pracy dyplomowej, a jednocześnie jest przykładem takowej pracy pod względem struktury i formatowania. Źródło \LaTeX -a to plik `main.tex`, plik ten używa klasy `iiuwb.cls` (autorstwa dr hab. Artura Korniłowicza, zmodyfikowany przeze mnie) z formatowaniem dokumentu zgodnie z wymaganiami naszego Wydziału; oraz pliku `biblio.bib`, będącego przykładem wpisów bibliograficznych \LaTeX -a, które używane są do formatowania bibliografii dokumentu. Dokument zawiera dwa przykładowe rysunki: wektorowy z pliku `chords3.eps` oraz rastrowy z pliku `Show02b.jpg`.

Wstęp powinien przedstawić:

1. motywację (dlaczego praca/tematyka ma sens i do czego może posłużyć);
2. cel pracy (co autor ma zamiar osiągnąć, jakie programy, aplikacje, algorytmy stworzyć/zmodyfikować/przetestować/zinterpretować/połączyć);
3. zakres pracy, przeprowadzone czynności (jasno wyszczególniony wkład własny);
4. kontekst pracy (co osiągnięto w danej dziedzinie, jaka jest konkurencja, co istnieje, a czego brakuje);
5. mini-streszczenie pracy (co znajduje się w którym rozdziale – jak poniżej).

W mini-streszczeniu i ogólnie w tekście odnosząc się do (pod)rozdziałów używamy (jeżeli możliwe) bezwzględnej automatycznej rozdziałów/sekcji („rozdział 2”, „sekcja 4.2.1”, etc.) zamiast relatywnych określeń „następny”, „kolejny” czy „ostatni”. Jeżeli mamy zdefiniowane etykiety `\label` to możemy dzięki nim niezawodnie odnosić się do takiej numeracji. Do nienumerowanych rozdziałów (Wstęp, Podsumowanie, Streszczenie) odnosimy się bezpośrednio, np. „We wstępie pracy...”, „W podsumowaniu...”.

Rozdział 1 motywuje obowiązek i korzyści z pisania pracy dyplomowej w \LaTeX -u, rozdział 2 przedstawia podstawowe i szczegółowe zalecenia edycyjne, rozdział 3 prezentuje kilka przykładów popularnych formatowań w \LaTeX -u, wreszcie podsumowanie tego dokumentu mówi o treściach, jakie powinny być w podsumowaniu pracy zawarte.

Rozdział 1

Dlaczego L^AT_EX?

Rozdział ten motywuje imperatyw pisania w świetlistym L^AT_EX-u i porzucenie *MS Word*, jako narowistego edytora typu czarnej skrzynki.

1.1 Dlaczego L^AT_EX, a nie *MS Word*?

Bo jest wygodniejszy i produkuje ładniejsze dokumenty. Konkretniej:

1. Łatwo jest panować nad formatem dokumentu (wszystko jest widoczne w pliku tex), w *Wordzie* mamy mnóstwo słabo widocznych lub niewidocznych ustawień, jak puste linie z różną wielkością czcionki, odstępy po/przed akapitem, odstępy między liniami, możliwe jest pozycjonowanie w poziomie spacjami, tabulacjami, tabulatorami itp. Każdy element tekstu może mieć własne formatowanie/paragrafowanie, słabo widoczne i niezgodne z obowiązującym schematem ogólnym. Generalnie w *Wordzie* mamy mniejszy lub większy bałagan, o ile purytańsko nie przestrzega się porządku.
2. Pozycjonowanie rysunków i tabel w L^AT_EX-u jest **(1)** automatyczne i **(2)** niekiedy trudne do kontroli. Stąd sugeruję świadomy wybór pierwszej opcji. Małe elementy zwykle da się umieścić w wybranym miejscu, z automatycznym ustawianiem większych elementów w wybranych przez kompilator miejscach (np. góra kolejnej strony) wypada się pogodzić (celem jest żeby zawartość dokumentu była jak najsensowniej rozmieszczona). W porównaniu z mało kontrolowanym pozycjonowaniem rysunków/grafiki w *MS Word* i tak jest to bardzo wygodne.
3. Automatyczne i zawsze aktualne spisy treści, rysunków i tabel.
4. Wygodne w edycji i automatycznie numerowane formuły matematyczne.
5. Automatyczne formatowanie bibliografii, „bazodanowe” wpisy bibliografii.
6. Szybka zmiana formatu dokumentu/bibliografii w razie potrzeby (Państwo raczej tego nie będą potrzebować).
7. Dokumenty w L^AT_EX-u można łatwo przetwarzać do PDF, który jest formatem bardziej uniwersalnym od .doc czy .docx.

8. *Last but not least* – mam dosyć pisania o stylach, podwójnych spacjach, ustawieniach akapitu i pustych liniach w dokumentach *Worda*.

1.2 Czego używać do edycji kodu L^AT_EX-a?

Możliwości jest bardzo wiele, zależnie od platformy sprzętowej, „purystycznie” wystarczy notatnik *Windowsa* czy też vim, ale oczywiście dużo wygodniejsze są dedykowane pakiety jak Texworks czy WinEdt. Ostatnio coraz większą popularność zdobywają systemy SaaS, czyli serwisy internetowe w tym przypadku udostępniające edytor, kompilację oraz składające pliki L^AT_EX-a w sieci. Oczywiście korzyści w tym przypadku to dostęp z poziomu przeglądarki www, niezależny od lokalnej maszyny, brak konieczności instalacji oprogramowania czy pakietów, prosta konfiguracja, możliwość udostępnienia projektu promotorowi do pełnego wglądu, komentowania czy nawet edycji. Dość oczywistym kłopotem może być z kolei dostępność Internetu.

Najbardziej polecam darmowy SaaS www.overleaf.com (opis w sekcji 1.2.1. Lokalnie pod *Windows* polecam dość skomplikowany edytor L^AT_EX-a WinEdt (opis w sekcji 1.2.2).

1.2.1 Polecany edytor online: www.overleaf.com

1. Do używania Overleaf.com niezbędna jest rejestracja, możemy swobodnie działać na darmowej licencji.
2. Tworzymy pusty projekt (*Blank Project*).
3. Z menu w górnym lewym rogu wybieramy z comboboxa Spell Check wartość "Polish".
4. Podmieniamy lub kopiujemy zawartość pliku tex.
5. Ładujemy pliki iuuwb.cls i biblio.bib (należy oczywiście uzupełnić przykładowy biblio.bib o swoje własne wpisy bibliograficzne), dodatkowo potrzebne obrazy np. w formatach png, jpg, eps czy pdf. Najbardziej polecane oczywiście wektorowe formaty obrazów (ps, eps, pdf), do których można zapisywać np. schematy rysowane w edytorze Worda, czy sporządzane w wektorowych programach graficznych.
6. Klikamy *Recompile* i voila! Powinien być gotowy pdf, kompilowany z

Mamy możliwość częściowego pokazania źródła tex-a w postaci bogato formatowanej (*Rich Text*) oraz intuicyjnej edycji, nie tracąc pełnej kontroli nad dokumentem z poziomu źródła! W celu udostępnienia projektu klikamy *Share* na górze. Ten projekt jest dostępny do oglądania pod adresem <https://www.overleaf.com/read/sxzpkyfjkpxz>. Więcej w pomocy online serwisu.

1.2.2 Edytor hardcore pod Windows: WinEdt

WinEdt jest dostępny do darmowego użytku pod adresem <http://www.winedt.com/download.html>, (darmowy do ewaluacji przez miesiąc, po miesiącu wyświetla przypomnienia o rejestracji). Jest oprogramowaniem o dużych możliwościach, stąd i nieco skomplikowanym. Podświetla on składnię, zarządza strukturą, automatyzuje czynności i sprawdza pisownię. Jako środowisko L^AT_EX-a pod *Windows* polecam *MiKTeX*: <http://miktex.org/download>. Dodatkowo polecam doinstalować sobie polskie słowniki, zależnie od wersji *WinEdt* jest to bardziej lub mniej skomplikowane, czytanie pomocy *WinEdt* niezbędne.

Możliwe w WinEdt jest też m.in. zdefiniowanie skrótów klawiaturowych, nie wnikając zbytnio w składnię i ich "wyklikiwanie", poniżej przykład definicji skrótów do *italicsa*, **bolda** oraz 4 poziomów nagłówków, zdefiniowane w sekcji *Shortcuts*, pliku konfiguracyjnego *MainMenu.ini* wersji 7.1 WinEdt. Można zaryzykować bezpośrednie przeklejenie kodu w podobne miejsce.

```
MENU="Shortcuts"
CAPTION="Shortcuts"
INVISIBLE=1
ITEM="$Format_Selection_Italics"
    CAPTION="Format Selection Italics"
    MACRO="Exe('%b\Macros\Fonts\Italic.edt');"
    SHORTCUT="24649::Shift+Ctrl+I"
    REQ_DOCUMENT=1
ITEM="$Format_Selection_Bold"
    CAPTION="Format Selection Bold"
    MACRO="Exe('%b\Macros\Fonts\Bold.edt');"
    SHORTCUT="24642::Shift+Ctrl+B"
    REQ_DOCUMENT=1

ITEM="$Format_Selection_Chapter"
CAPTION="Format Selection Chapter"
MACRO="Exe('%b\Macros\Sections\Chapter.edt');"
SHORTCUT="24625::Shift+Ctrl+1"
REQ_DOCUMENT=1
ITEM="$Format_Selection_Section"
CAPTION="Format Selection Section"
MACRO="Exe('%b\Macros\Sections\Section.edt');"
SHORTCUT="24626::Shift+Ctrl+2"
REQ_DOCUMENT=1
ITEM="$Format_Selection_SubSection"
CAPTION="Format Selection Subsection"
MACRO="Exe('%b\Macros\Sections\Subsection.edt');"
```

```
SHORTCUT="24627::Shift+Ctrl+3"
REQ_DOCUMENT=1
ITEM="$Format_Selection_SubSubSection"
CAPTION="Format Selection Subsubsection"
MACRO="Exe('"%b\Macros\Sections\Subsubsection.edt');"
SHORTCUT="24628::Shift+Ctrl+4"
REQ_DOCUMENT=1"
```

1.3 Jak pisać w L^AT_EX-u?

Google zwraca na takie zapytanie około 382 000 wyników, nie ma sensu tego powtarzać. Do dyspozycji jest wspomniany projekt początkowy (z klasą dokumentu), wystarczy modyfikować i uzupełniać jego zawartość, aż do otrzymania pożądanej treści.

Rozdział 2

Zalecenia edycyjne

Rozdział ten przedstawia różne aspekty edycji pracy dyplomowej: format, załączniki, strukturę, ogólne zalecenia pisarskie, częste błędy oraz jak korygować pracę.

2.1 Format fizyczny pracy oraz załączniki

Pracę dyplomową należy **drukować dwustronnie i oprawiać w miękkie okładki** (klasa dokumentu ma fabrycznie naprzemiennie 1cm przeznaczony na oprawę, stąd zawartości stron są naprzemiennie przesunięte do lewej i prawej, to jest zamierzone!). Kolejne rozdziały należy rozpoczynać na stronie „prawej” (na nieparzystych numerach stron), czyli poprzedzać je komendą `\cleardoublepage`. Orientacyjnie: praca licencjacka powinna mieć minimalnie 40 stron (preferowane co najmniej 50). Orientacyjnie: praca magisterska powinna mieć minimalnie 50 stron (preferowane co najmniej 60).

Do pracy dołączamy płytę CD/DVD/BlueRay z następującą zawartością (5 folderów):

1. dokumentacja: praca dyplomowa w wersji elektronicznej;
2. postać wykonywalna aplikacji;
3. kody źródłowe aplikacji;
4. przykłady (m.in. pliki z danymi do testów).
5. zawartość stron www, do których odnosimy się w referencjach, w formie "archiwum strony www" (jeden plik z rozszerzeniem .mht, zapisywany z poziomu przeglądarki www).

2.2 Struktura pracy

W pracy należy wyróżnić (nieformalnie, bo formalnie to będą po prostu rozdziały) część teoretyczną i praktyczną. W części teoretycznej omawiamy zagadnienia związane

z tematem pracy (ujęte w ramce pracy). W części praktycznej prezentujemy koncepcje i rozważania autora oraz jego prace (projektowe, teoretyczne, implementacyjne). Zwykle ostatnim rozdziałem części praktycznej jest „Podręcznik użytkownika” z wymaganiami sprzętowymi i programistycznymi (sprzęt, SO, środowiska uruchomieniowe, serwery aplikacji i baz danych), opisem instalacji systemu/aplikacji, wyliczeniem i opisem zaimplementowanych funkcji oraz ze zrzutami ekranu ilustrującymi poszczególne funkcjonalności.

1. Rozdziały nazywamy, co ukierunkowuje od razu treść rozdziału. Nazwa podrozdziału w ideale powinna być samowystarczalna i nie zależeć od kontekstu (np. nie nazywamy podrozdziału „Podział” czy „Historia”, ale np. „Podział klasyfikatorów”, „Historia Internetu”).
2. W spisie treści nie pojawiają się (zakładając brak modyfikacji proponowanej klasy dokumentu `iiuwb.cls`) więcej niż trzy poziomy struktury (licząc `\chapter` jako pierwszy z nich, czyli jeszcze `\section` oraz `\subsection`); w samej treści pracy możemy używać też `\subsubsection`, a nawet sporadycznie (`\paragraph`).
3. Tytułów (pod)rozdziałów nie kończymy kropką.
4. Tytuły (pod)rozdziałów nie powinny być zbyt długie.
5. Nie należy tworzyć zbyt długich ani zbyt krótkich akapitów (minimum trzy zdania). Tekst akapitu powinien być zwarty tematycznie, najlepiej by kolejny akapit nawiązywał do poprzedniego i stanowiły w ten sposób logiczną całość wywodu.

2.2.1 Przykładowa subsection

Subsection wygląda jak wyżej. Jak widać: jest numerowane trzema liczbami.

Przykładowa subsubsection

Subsubsection wygląda jak wyżej. Można dedefiniować jego numerowanie, ale nie będzie to czytelne (cztery liczby).

Przykładowy paragraph Z kolei *paragraph* wygląda jak obok i nie ma po nim domyślnego przeniesienia do kolejnej linii (*line feed*).

2.3 Ogólne zalecenia pisarskie

1. pojęcia i słowa z języka obcego wyróżniamy poprzez *kursywę*;
2. słowa kluczowe z języka programowania, nazwy klas, metod, funkcji i zmiennych, czy też znaczniki HTML wyróżniamy przez **wytluszczenie**, np. `MyClass`, `<h2>`;

3. nazwy funkcji i metod powinny mieć przy sobie nawias, aby łatwo dało się je zidentyfikować, np. **RetrieveData()**, **toString()**;
4. nie używamy do wyróżniania KAPITALIKÓW (czyli tzw. CAPSÓW);
5. nazwy kontrolki GUI, plików, tabel BD itp. programistycznych elementów (za wyjątkiem słów kluczowych, nazw klas, funkcji, metod i zmiennych) wyróżniamy *kursywą*;
6. nazwy własne lub skróty tych nazw proponuję pisać kursywą, jest czytelniejsze i lepiej widoczne, np. *SSN*, *ERP*, *CSS*, *AJAX*, etc.
7. pierwsze wystąpienie ważnego terminu można zaakcentować wytłuszczając go;
8. ewentualne cytaty (np. przepisy prawne) proponuję w „cudzysłowie” i *kursywą*;
9. ważne klasyfikacje podkreślamy **pogrubieniem**, ułatwia to czytelnikowi zrozumienie istoty podziału; podobnie postępujemy z wyliczanymi etapami procesu;
10. każdy rysunek powinien być czytelny (m.in. wielkość tekstu legendy, opisu osi, jednostka, itp.), zapowiedziany i skomentowany w tekście; rysunek powinien być kontrastowy, na jasnym tle;
11. wykresy wklejane z MS Excela powinny mieć sensowne zakresy i osie (np. procenty w zakresie [0;100], a nie [-10;120]);
12. każda tabela powinna być omówiona w tekście;
13. w tabelach stosujemy sensowne formatowanie liczb, zależnie od kontekstu, np. dwie, trzy cyfry po przecinku;
14. **Twarda spacja** W składaniu tekstu za błąd (tzw. sierotka) uważa się pozostawienie na końcu lub na początku wiersza osamotnionego krótkiego słowa lub symbolu, zwłaszcza jednoliterowego. Zapobiega temu wstawianie tzw. twardych spacji w miejsca, w których nie powinna być *łamana* linia. Twardą spację wstawia się **zamiast** zwykłej spacji, aby podczas kompilacji L^AT_EX pozostawił połączone twardą spacją wyrazy w tej samej linii. Aby wstawić twardą spację w L^AT_EX-u należy zamiast zwykłej spacji użyć symbolu tyldy '~' (tyldę w Windows uzyskujemy naciskając równocześnie *Shift* oraz klawisz tuż pod *Escape*, a następnie spację). Używamy twardej spacji z pewnością po jednoliterowych spójnikach i przyimkach, symboli waluty, np. *w domu*, *i dalej*, *a także*, *o wszystkim*, *u Stefana*, *z zamkiem*, *50 \$*, *Władysław IV*, *50 mAh*. Zaleca się również po wyrazach dwu- a nawet i trzyliterowych (po, **np.**, od, do, za, nie, się). Twarda spacja ma stałą szerokość, równą nominalnej wielkości zwykłej spacji, stąd generowany odstęp może niekiedy wydawać się niestety nieestetyczny.

15. najczęściej stawiamy przecinek przed „jako”, „z czym”, „które”, „jakie”, „od których”, „od jakich”, „tak więc”, „na co”, „ale”;
16. nie stawia się spacji przed znakiem przestankowym: znaki interpunkcyjne (kropkę, przecinek, średnik, wykrzyknik, znak zapytania itp.) stawia się zawsze bezpośrednio po wyrazie;
17. przed „(” stawiamy spację, po - nie; dla „)” odwrotnie;
18. rozróżniamy łączniki „-” oraz półpauzy „ – ” i pauzy „ — ”:

Myślniki, to według powszechnej opinii poziome kreski w tekście. W rzeczywistości myślnik to pauza lub półpauza. Krótka pozioma kreska, to dywiz. W zwykłych programach edycyjnych dywiz traktowany jest równorzędnie ze znakiem minus. Dywiz nie jest oddzielany żadnymi odstępami od tekstu, stanowiąc z nim jedną całość. Zapisujemy zatem: hokus-pokus, XV-lecie, Konstancin-Jeziorna albo ani mru-mru lub bara-bara. Pauza i częściej stosowana półpauza, to kreska oddzielona z obu stron odstępami i stosowana np. między liczbami podającymi wartość przybliżoną np. „... co stanowi 56 – 60% populacji”. Pauzy (półpauzy) używamy również w dialogach literackich na początku wiersza, we wtrąceniach, w wyliczeniach wersowych, między wyrazami przeciwstawnymi np. w wyrażeniu „góra – dół to przeciwne krańce”.¹

Zwykle w L^AT_EX-u używamy półpauzy (dwa minusy otoczone spacjami) lub łącznika (jeden minus nieotoczony spacjami);

19. wypunktowania rozpoczynane wielką literą („wielozdaniowe”) kończymy zawsze kropką;
20. wypunktowania rozpoczynane małą literą (jednozdanowe lub równoważniki zdań) kończymy średnikiem lub przecinkiem, dopiero ostatnie (jak te) kończymy kropką.

2.4 Częste błędy gramatyczne

przy pomocy kogoś – „mając kogoś za pomocnika, korzystając z czyichś usług, poparcia, pomocy”

za pomocą czegoś – „posługując się czymś, używając czegoś”

własność – „to, co ktoś posiada; rzecz własna, mienie, majątek”

właściwość – częściej w liczbie mnogiej „to, co jest charakterystyczne dla danej osoby lub rzeczy; cecha”

porównujemy coś z czymś, a nie coś do czegoś

¹ Za: <http://yestok.pl/gen/y07.php>

2.5 Korekta

W końcowej fazie edycji konieczna jest dokładna korekta całości pracy pod kątem „literówek” (szczególnie wyrazów poprawnych językowo, których nie wykrywa sprawdzanie słownikowe), zapomnianych ogonków, „zjedzonych” lub zdublowanych wyrazów, oraz gramatyki i interpunkcji (zasady: <http://so.pwn.pl/zasady.php?id=629734>).

Rozdział 3

Przykłady typowych czynności w L^AT_EX-u

Niniejszy rozdział zawiera przykłady najczęstszych zastosowań L^AT_EX-a.

3.1 Wypunktowania

Przykład wypunktowania:

- Przed ukośnikiem umieszczamy te ilości komórek w sąsiedztwie, które są wymagane do przeżycia (dla reguły Conwaya będzie to 23)
- Następnie umieszczamy ukośnik: „/” (ASCII 47)
- Po ukośniku umieszczamy te ilości komórek w sąsiedztwie, dla których powstaną żywe komórki na martwych polach (dla reguły Conwaya będzie to 3)

Przykład wypunktowania numerowanego:

1. Przed ukośnikiem umieszczamy te ilości komórek w sąsiedztwie, które są wymagane do przeżycia (dla reguły Conwaya będzie to 23)
2. Następnie umieszczamy ukośnik: „/” (ASCII 47)
3. Po ukośniku umieszczamy te ilości komórek w sąsiedztwie, dla których powstaną żywe komórki na martwych polach (dla reguły Conwaya będzie to 3)

Przykład wypunktowania deskryptywnego (z opisem w nowej linii):

Klasa 1

Prawie wszystkie początkowe wzorce, szybko ewoluują do stałych i niezmiennych modeli, w których wszystkie losowości zanikają.

Klasa 2

Prawie wszystkie początkowe wzorce ewoluują do stałych lub oscylujących struktur. Część losowości pozostaje, a lokalne zmiany w początkowym wzorcu pozostają lokalne.

Klasa 3

Prawie wszystkie początkowe wzorce ewoluują w pseudo-losowy, chaotyczny sposób. Większość struktur zostaje szybko niszczona przez wszechobecny chaos, a lokalne zmiany w początkowym wzorcu mają tendencję do rozprzestrzeniania się w nieskończoność.

Klasa 4

Prawie wszystkie początkowe wzorce ewoluują w kompleksowe struktury, mogące ze sobą oddziaływać w skomplikowany i interesujący sposób. Niekiedy po przejściu dużej ilości pokoleń wynikiem może być klasa 2, z strukturami oscylacyjnymi. Wolfram¹ przypuszcza, że wiele – o ile nie wszystkie – automaty komórkowe klasy czwartej są zdolne do obliczeń. Zostało to udowodnione dla reguły 110 i reguły Conweya.

3.2 Dołączanie rysunków

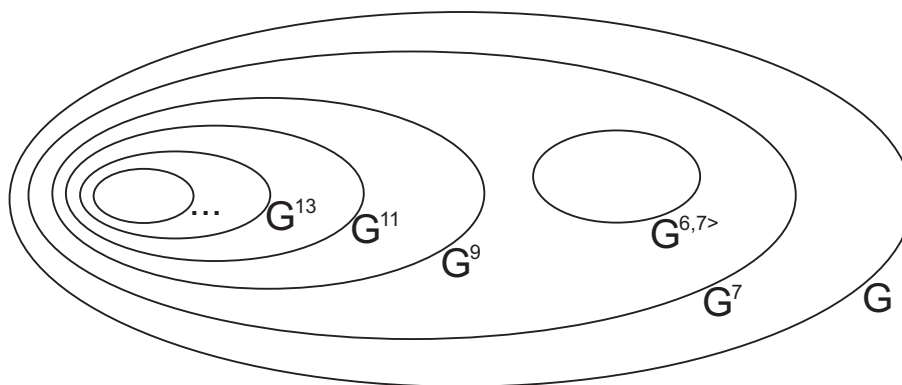
Rysunki centrujemy na stronie.

Poniżej przykład rysunku wektorowego (*eps*) - w *WinEdt* możliwe jest i *TeXify* (do *DVI*, później dopiero do *PDF*) oraz *PDFTeXify* (bezpośrednio do *PDF*), preferowane *TeXify* ponieważ *DVI* jest szybsze, w razie pozostawienia otwartego pliku *DVI* „uaktualnia” plik pokazując te same miejsce dokumentu, co jest bardzo wygodne. Obrazy rastrowe można konwertować do wektorowych np. drukując wirtualną drukarką (*BullZip PDF Printer* -> *Save as EPS*) lub za pomocą programów graficznych, niestety zwykle ze zniekształceniami. Idealem jest utworzenie obrazu wektorowego i zapisanie go jako *eps* lub *ps*.

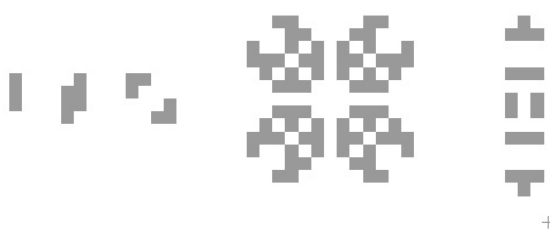
Na rysunku 3.1 pokazano przykładowy rysunek wektorowy (utworzony w *Corel Draw* i zapisany jako *.eps*).

Rysunek 3.2 prezentuje dołączony plik rastrowy (rastrowe są m.in. *jpg*, *png*, *bmp*). Niekiedy mogą powstać zniekształcenia wynikające ze skalowania, zalecany format to *.png*, bo jest kompresowany bezstratnie. Kiedy dołączamy rastrowe obrazy, w *WinEdt* niemożliwa jest opcja *TeXify* (do *PS*), trzeba używać *PDFTeXify* (bezpośrednio do *PDF*). Jeżeli mamy problem ze sporządzeniem rysunków wektorowych, to można używać rastrowych (w dobrej rozdzielczości) i ewentualnie je chwilowo wykomentować, aby korzystać z korzystniejszej kompilacji do *DVI*.

¹Cellular Automata (1983) Stephen Wolfram



Rysunek 3.1: Podklasy funkcji harmoniczych



Rysunek 3.2: Przykładowe oscylatory o okresie 2, 3 oraz 15

3.3 Przykład tabeli

Tabela 3.1 prezentuje ustalanie *Note Importance*.

Tablica 3.1: Determining Note Importance

$\log_2 \frac{NoteLength}{HarmonicFragmentLength}$	Initial value
2	1.6
1	1.4
0	1.2
-1	1.0
-2	0.8
-3	0.7
-4	0.5
-5	0.3

Proste tabele tworzymy ręcznie, bardziej skomplikowanie (łączenie kolumn, wierszy itp.) można wygenerować używając pluginu do Excela, który generuje z selekcji kod LaTeXa tablicy: <http://www.ctan.org/tex-archive/support/excel2latex/>, opisany w <http://blog.modelworks.ch/?p=153>.

3.4 Przykład formuły matematycznej

Analogicznie możemy użyć **t-norm** i **t-conorm** zamiast operacji **max** i **min**, jak w równaniu 3.1.

$$[h]\mu(d) = \sum_{x \in X} t(\mu_R(x), \mu_P(x)) \quad (3.1)$$

3.5 Przykład z kodem programu

Poniżej kod programu z zachowanymi liniami oraz wcięciami (czcionka pomniejszona):

```
class MAP_PUZZLE {
public:
    int state;
    bool alive;
    int n,v[10];
    void inline clear();
    void inline import(int a);
    void inline importnear(int a);
    void inline die();
    void inline dienear(int a);
    void set(volatile int & n,volatile int & s,
            volatile int *& v,volatile bool & alive);
    MAP_PUZZLE();
};
```

3.6 Referowanie do literatury

Zalecenia:

1. literatura powinna zawierać minimalnie 8 poważnych pozycji, preferowane co najmniej 15;
2. tekst pracy powinien co najmniej raz odwoływać się do każdej pozycji bibliografii, a jeżeli jest to sensowne to wielokrotnie;
3. referencje do stron www powinny być wysoce wiarygodne;
4. Wikipedii (i podobnych źródeł o niepewnej reputacji i zmiennej treści) raczej nie podajemy jako bibliografii, proszę szukać poważniejszych źródeł (dla definicji co najmniej Słownik PWN).

Przykład odwołań do literatury: „Zagadnienie te zostało opisane w [1], szerzej w [2], bardzo szczegółowo w następujących trzech pracach [3, 4, 5], wreszcie podsumowane w [6].”

3.7 Linki wewnętrzne i zewnętrzne

Pakiet *hyperref* automatycznie generuje linki w obrębie dokumentu do wszelkiej zawartości "referowalnej", typu sekcje, rysunki, tabele, referencje. Można definiować linki zewnętrzne za pomocą komendy `\url{...}`. Dokumentacja pakietu: <http://en.wikibooks.org/wiki/LaTeX/Hyperlinks>.

Podsumowanie

Podsumowanie ma treści zbliżone do **Wstępu**, dodatkowo powinno oceniać pracę wykonaną przez autora (spełnienie założeń i osiągnięcie celu pracy). W miarę możliwości wypada też przedstawić możliwe zastosowania pracy oraz perspektywy rozwoju (aplikacji/algorytmu, ale też i tematyki z tym związanej).

Roboczo można sporządzić 5 oddzielnych akapitów z tytułami (do późniejszego usunięcia) jak następuje:

1. Motywacja i kontekst
2. Zakres pracy, przeprowadzone czynności (jasno wyszczególniony wkład własny)
3. Osiągnięte wyniki, interpretacja wyników, wnioski, etc.
4. Ocena spełnienia założeń i osiągnięcia celu pracy
5. Perspektywy rozwoju

Docelowo akapitów może być więcej (jeżeli byłyby zbyt długie).

W podsumowaniu pracy proszę używać numerów rozdziałów („rozdział 2”, „we wstępie pracy”, etc.) zamiast relatywnych określeń „następny”, „kolejny” czy „ostatni”.

Bibliografia

- [1] H. R. Arabnia, editor. *Proceedings of the 2006 International Conference on Image Processing, Computer Vision, & Pattern Recognition, Las Vegas, Nevada, USA, June 26-29, 2006, Volume 1*. CSREA Press, 2006.
- [2] IMDB: Of Mice and Men. <http://www.imdb.com/title/tt0105046/>. [dostęp: 2013-03-08].
- [3] K. Delac and M. Grgic. *Face Recognition*. I-Tech Education and Publishing, Vienna, Austria, 2007.
- [4] V. Gomathi, Dr. K. Ramar, and A. Santhiyaku Jeevakumar. Human facial expression recognition using manfis model. *International Journal of Electrical and Electronics Engineering* 3:6, 2009.
- [5] M. Hess and M. Martinez. Facial feature extraction based on the smallest univalue segment assimilating nucleus (susan) algorithm, picture coding symposium. In *Proceedings of Picture Coding Symposium, San Franscisco, California*, pages 152–159, 2004.
- [6] Aliaa A. A. Youssif and Wesam A. A. Asker. *Automatic Facial Expression Recognition System Based on Geometric and Appearance Features*, volume 4:2, chapter Computer and Information Science. Published by Canadian Center of Science and Education, March 2011.

Spis rysunków

3.1	Podklasy funkcji harmoniczych	19
3.2	Przykładowe oscylatory o okresie 2, 3 oraz 15	19

Spis tablic

3.1	Determining Note Importance	19
-----	---------------------------------------	----