UNIWERSYTET W BIAŁYMSTOKU
WYDZIAŁ INFORMATYKI

John James RAMBO

Jak (na)pisać pracę dyplomową? (v9)

PRACA LICENCJACKA

Promotor:
dr inż. Mariusz RYBNIK

BIAŁYSTOK 2025

Spis treści

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, zmodyfikowanej nieznacznie przeze mnie) z formatowaniem dokumentu zgodnym z wymaganiami naszego Wydziału; oraz pliku `biblio.bib`, będącego przykładem wpisów bibliograficznych LaTeX-a, które używane są do formatowania bibliografii dokumentu.

Informacji i porad zawartych w tym dokumencie nie należy przedkładać ponad zalecenia promotora czy dziekana!

Do rzeczy więc: wstęp powinien przedstawiać:

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) dokładnej numeracji rozdziałów/sekcji („rozdział 2”, „sekcja 4.2.1”, etc.) zamiast relatywnych określeń typu: „następny”, „kolejny” czy „ostatni”. Jeżeli mamy zdefiniowane etykiety **\label** to możemy dzięki nim niezawodnie odnosić się do takiej numeracji. Do nienumerowanych rozdziałów (np. Wstęp, Podsumowanie, Bibliografii) odnosimy się bezpośrednio, np. „We wstępie pracy...”, „W podsumowaniu...”. Unikamy podawania numerów stron jako lokacji elementów treści.

Rozdział 1 motywuje 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 LaTeX?

Rozdział ten motywuje imperatyw pisania w świetlistym LaTeX-u i porzucenie *MS Worda*, jako narowistego edytora typu czarnej skrzynki.

1.1 Dlaczego LaTeX, a nie *MS Word*?

Bo jest wygodniejszy i produkuje ładniejsze dokumenty. Konkretniej:

1. Łatwo jest panować nad formatem dokumentu (wszystko jest jawnie 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 i zdefiniowanych stylów. Nawet pomimo tego niektóre automatyczne ustawienia jak np. wcięcia wypunktowań bywają przez program ustalane w sposób niezrozumiały dla użytkownika i wymagają poprawy oraz ciągłej uwagi.

2. Automatyczne i zawsze aktualne spisy treści, rysunków, listingów i tabel.

3. Wygodne w edycji i automatycznie numerowane formuły matematyczne.

4. Automatyczne formatowanie bibliografii, „bazodanowe" wpisy bibliografii.

5. Szybka zmiana formatu dokumentu czy formatu bibliografii w razie potrzeby.

6. Pozycjonowanie rysunków i tabel w LaTeX-u jest w pełni **(1)** automatyczne lecz niestety też zwykle **(2)** bez pełnej kontroli użytkownika. 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 (ostatecznym celem LaTeX-u jest, żeby zawartość dokumentu była jak najsensowniej rozmieszczona, co niekiedy kłóci się z poczuciem estetyki autora dokumentu, przyzwyczajonym też często do ręcznego składania dokumentu). W porównaniu ze średnio kontrolowanym pozycjonowaniem rysunków/grafiki w *MS Word* i tak jest to wygodne. Nawet stosowanie podziałów strony, strony parzystej itp. znaczników może prowadzić do dość nieestetycznych rezultatów.

7. *Last but not least* – w LaTeX-u niemożliwa jest przypadkowa modyfikacja stylów, wielokrotne spacje czy tabulacje, niespójne ustawiania akapitów czy pozycjonowanie pustymi liniami, co nagminnie napotkać można w dokumentach *Worda*.

1.2 Czego używać do edycji kodu LaTeX-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 składujące pliki LaTeX-a w chmurze. Oczywiste 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 konieczność dostępu do Internetu.

Najbardziej polecam darmowy SaaS `www.overleaf.com` (opis w sekcji 1.2.1). Lokalnie pod *Windows* polecam dość skomplikowany edytor LaTeX-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. Ściągamy ze strony `http://iist.uwb.edu.pl/~m.rybnik/index.php/skarb-dyplomanta-uwb/` paczkę zip z projektem przykładowej pracy dyplomowej.

5. Przeklejamy lub kopiujemy zawartość pliku tex w projekcie online zawartością ściągniętego pliku *main.tex*.

6. *Upload*-ujemy do projektu pliki *iiuwb.cls* i *biblio.bib* oraz przykładowe grafiki *chords3.eps* oraz *Show02b.jpg*.

7. Klikamy *Recompile* i voila! Powinien być utworzony pdf, kompilowany z naszego projektu LaTeX-a.

8. Piszemy pracę: edytujemy plik tex, dokładamy własne wpisy bibliograficzne do *biblio.bib*, *upload*-ujemy rysunki, itd.

9. opcjonalnie udostępniamy pracę promotorowi.

Mamy też możliwość pokazania części źródła LaTeX-a w postaci bogato formatowanej (przycisk *Rich Text* nad kodem dokumentu) oraz intuicyjnej edycji, nie tracąc równocześnie pełnej kontroli nad dokumentem z poziomu źródła! W celu udostępnienia projektu klikamy *Share* po prawej na górze i kopiujemy link do odczytu lub edycji. Na przykład projekt overleaf użyty do tego dokumentu jest dostępny do odczytu 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 LaTeX-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 LaTeX-u?

Google zwraca na takie zapytanie około $5 \cdot 10^5$ wyników (przykład wyrażenia matematycznego *inline*), nie ma więc sensu powtarzać podstaw składni. Dalsze rozdziały skupiają się głównie na ogólnych zasadach pisania pracy dyplomowej oraz typowych strukturach LaTeX-a z tym związanych. Do dyspozycji jest wspomniany projekt początkowy (z klasą dokumentu) oraz przykładami popularnych zastosowań, zasadniczo wystarczy więc przez analogię kopiować, modyfikować i uzupełniać jego zawartość, aż do otrzymania ostatecznej 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 Złożenie pracy w Archiwum Prac Dyplomowych

Pracę dyplomową należy złożyć w formie elektronicznej w Archiwum Prac Dyplomowych (`pd.uwb.edu.pl`). Możliwych jest pięć etapów:

Etap 1. Wpisywanie danych pracy – **Autor pracy** Etap startowy. Autor wprowadza streszczenie w języku polskim oraz angielskim. Autor wprowadza słowa kluczowe w języku polskim oraz angielskim (od 3 do około 8).

Etap 2. Przesyłanie plików z pracą – **Autor pracy** Do APD za wyraźną zgodą promotora Autor przesyła:

1. skompilowany dokument pracy w formacie pdf,
2. wypełnione, podpisane i zeskanowane oświadczenie o samodzielności napisania pracy dyplomowej,
3. aplikacja, serwis itp. praca praktyczna; w formie kodu źródłowego oraz skompilowanej, zależnie oczywiście od użytego środowiska.

Etap 3. Akceptacja danych – **Promotor** Promotor zleca sprawdzenie tekstu pracy systemem antyplagiatowym, który porównuje tekst z innymi pracami z baz oraz z dokumentami ze stron internetowych oraz wykrywa ewentualne maskowanie kopiowania treści typu białe znaki czy litery z innych alfabetów. Po uzyskaniu pozytywnego wyniku z tego systemu promotor akceptuje pliki i dane wprowadzone przez Autora pracy.

Etap 4. Wystawianie recenzji **Pracę na etapie 4 uznaje się za złożoną - terminy!**. Obecnie obowiązujące termin podstawowy to koniec sesji zasadniczej, ewentualne jednokrotne przedłużenie można uzyskać do końca sesji poprawkowej.

Etap 5. Praca gotowa do obrony Oprócz pracy w etapie 5, obecnie do obrony należy też zaliczyć wszystkie przedmioty objęte programem studiów (uzyskanie tzw. absolutorium).

2.2 Format pliku pracy oraz płyta CD/DVD

Pracy dyplomowej nie ma obowiązku drukować, dostępna jest w formie elektronicznej w APD. Orientacyjny rozmiar pracy licencjackiej to minimalnie 40 stron (preferowane co najmniej 50). Orientacyjnie: praca magisterska powinna mieć minimalnie 50 stron (preferowane co najmniej 60). Do pracy opcjonalnie dołączamy płytę *CD / DVD / Blu-Ray* 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).

2.3 Struktura pracy

W pracy należy zwykle 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"). Tytuły (pod)rozdziałów nie powinny być jednak bardzo długie (orientacyjnie: powinny mieścić się w jednej linii).
2. W spisie treści nie pojawią 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. 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.

5. Każdy rozdział i sekcja powinien posiadać treść własną, przynajmniej wprowadzenie i przestawienie, co w rozdziale/sekcji się znajduje. Orientacyjnie powinny to być przynajmniej 3 zdania.

2.3.1 Przykładowa subsection

Subsection **(sekcja - poziom 3)** wygląda jak wyżej. Jak widać: jest numerowane trzema liczbami.

Przykładowa subsubsection

Subsubsection **(sekcja - poziom 4)** wygląda jak wyżej. Można dodefiniować jego numerowanie, ale raczej nie będzie to czytelne (cztery liczby).

Przykładowy paragraph Z kolei *paragraph* (najdrobniejsza domyślna sekcja - poziom 5) wygląda jak obok i nie ma po nim domyślnego przeniesienia do kolejnej linii (*line feed*). Taki jest standard LaTeX, acz może to budzić estetycznie lekki niepokój.

2.4 Ogólne zalecenia pisarskie

1. Spacje wielokrotne czy wielokrotne tabulacje znaczą w LaTeX-u tyle samo, co pojedyncza spacja.

   Jedna lub więcej pusta linia to zwykle zakończenie akapitu z przeniesieniem do nowego wiersza. Można też wymusić przeniesienie do nowego wiersza
   podwójnym *backslaszem* (\\). Używać tego należy z umiarem, ale można sobie formatować kod dokumentu na różne wygodne sposoby, jak np. naprzemienne wcięcia akapitów bez szkody dla finalnego wyglądu, itp.

2. pojęcia i słowa z języka obcego wyróżniamy zwykle poprzez *kursywę*;

3. słowa kluczowe z języka programowania, nazwy klas, metod, funkcji i zmiennych, czy też znaczniki HTML wyróżniamy zwykle przez **wytłuszczenie**, np. **MyClass**, <**h2**>;

4. nazwy funkcji i metod powinny mieć przy sobie nawias, aby łatwo dało się je zidentyfikować, np. **RetrieveData()**, **toString()**;

5. raczej nie używamy do wyróżniania KAPITALIKÓW (czyli tzw. CAPSÓW);

6. nazwy kontrolek GUI, plików, tabel baz danych itp. programistycznych elementów (z wyjątkiem słów kluczowych, nazw klas, funkcji, metod i zmiennych) wyróżniamy *kursywą*;

7. nazwy własne (szczególnie z obcego języka) lub skróty tych nazw proponuję pisać kursywą, jest to czytelniejsze, np. *Visual Studio, Java, SSN, ERP, CSS, AJAX*, etc.

8. pierwsze wystąpienie ważnego terminu można zaakcentować wytłuszczając go;

9. ewentualne cytaty (np. przepisy prawne) proponuję umieszczać w „cudzysłowie" i *kursywą*;

10. ważne klasyfikacje podkreślamy **pogrubieniem**, ułatwia to czytelnikowi zrozumienie istoty podziału; podobnie postępujemy z wyliczanymi etapami procesu czy algorytmu;

11. każdy rysunek powinien być czytelny (rozmiar czcionki legendy powinien być zbliżony do rozmiaru czcionki zwykłego tekstu, etykiety osi, jednostki, itp.), zapowiedziany i skomentowany w tekście; rysunek powinien być kontrastowy, na białym lub przynajmniej jasnym tle; najlepiej też używać rysunków w formatach wektorowych jak: wmf, ps, eps, svg, zwykle też pdf; rastrowe obrazy powinny być w odpowiedniej rozdzielczości/rozmiarze.

12. LaTeXrzuca obrazkami jak Anita Włodarczyk młotem, więc odnosimy się do nich w kodzie wyłącznie bezpośrednio przez *\ref* do zdefiniowanej *label*, np. „Rysunek \ref{fig:chords3} przedstawia schemat...". Odnośniki typu „poniżej", „powyżej" łatwo mogą zostać pozbawione sensu w trakcie dalszej edycji. Są w miarę bezpieczne jedynie dla niewielkich, nieczęstych rysunków poprzedzielanych sporą ilością tekstu.

13. wykresy wklejane np. z *MS Excela* czy R powinny mieć sensowne zakresy i osie (np. zmienne procentowe powinny być prezentowane w zakresie [0;100], a nie np. [-10;120] jak to się często *Excelowi* zdarza);

14. każdy rysunek, tabela czy listing kodu powinny być wspomniane i omówione w tekście pracy; rysunki obowiązkowo powinny mieć wskazane źródło (referencja do pozycji literatury, www lub „źródło własne" jeżeli zostały utworzone przez autora pracy); źródło tabel należy wskazać o ile nie są one tworem autora;

15. w tabelach stosujemy sensowne formatowanie liczb, zależnie od kontekstu, np. jedna, dwie czy trzy cyfry po przecinku;

16. **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 LaTeX pozostawił połączone twardą spacją wyrazy w tej samej linii. Aby wstawić twardą spację w LaTeX-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, od, do, za, nie, się). Twarda spacja ma stałą szerokość, równą nominalnej wielkości zwykłej spacji, stąd generowany odstęp może niekiedy wydawać się niestety nieestetyczny.

17. najczęściej stawiamy przecinek przed „jako", „z czym", „które", „jakie", „od których", „od jakich", „tak więc", „na co", „ale";

18. nie stawia się spacji przed znakiem przestankowym: znaki interpunkcyjne (kropkę, przecinek, średnik, wykrzyknik, znak zapytania itp.) stawia się zawsze bezpośrednio po wyrazie;

19. przed „(" stawiamy spację, po - nie; dla „)" odwrotnie;

20. 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]

    W akapicie powyżej mamy też przykład przypisu dolnego (**footnote**).

    Zwykle w LaTeX-u używamy półpauzy (dwa minusy otoczone spacjami) lub łącznika (jeden minus nie otaczany spacjami);

21. wypunktowania rozpoczynane wielką literą („wielozdaniowe") kończymy zawsze kropką;

22. wypunktowania rozpoczynane małą literą (jednozdaniowe lub równoważniki zdań) kończymy średnikiem lub przecinkiem (wszystkie jednakowo), dopiero ostatnie (jak te właśnie) kończymy kropką.

[1] Za: `http://yestok.pl/gen/y07.php`

2.5 Wybrane zasady interpunkcji

Przecinek stawiamy m.in. przed spójnikami: *a, aczkolwiek, ale, jednak, lecz, natomiast, tymczasem, wszakże, zaś, za to, dlatego, więc, zatem, toteż, czyli, mianowicie, ponieważ, to jest.*

„KTÓRY to zaimek wprowadzający zdania podrzędne, a więc „zaimek przecinkolubny". Przecinek wstawiamy przed całym zdaniem podrzędnym, a więc i przed zaimkiem KTÓRY – ale nie zawsze tuż przed samym słowem KTÓRY! Jeśli słowo KTÓRY jest poprzedzone przyimkiem, to PRZECINEK musi pojawić się PRZED tym PRZYIMKIEM, np. Stał tam pomnik, wokół którego ustawiono ławeczki (nie: Stał tam pomnik wokół, którego ustawiono...); Dom, koło którego posadzono te drzewa, już dawno się zawalił. Jeśli słowo KTÓRY jest poprzedzone wyrażeniem: na podstawie, za pomocą, na mocy, to przecinek stawiamy przed całostką: na podstawie którego, za pomocą którego, na mocy którego, np. Jest to utwór, na podstawie którego nakręcono trzy filmy, w tym jeden animowany (przestawny szyk: Jest to utwór, na którego podstawie nakręcono trzy filmy – też jest poprawny);"[2]

2.6 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;

„dzień dzisiejszy" – pleonazm.

2.7 Częste błędy ortograficzne

- „~~napewno~~",
- „~~wogóle/wogule~~",
- „~~na prawdę~~",
- „~~narazie~~",

[2] Za `https://nck.pl/projekty-kulturalne/projekty/ojczysty-dodaj-do-ulubionych/ciekawostki-jezykowe/ktory-i-przecinek`

- „~~po za tym/pozatym~~”,
- „~~na codzień~~”,
- „~~nacodzień~~”,
- „~~orginlany~~”,
- „~~muj~~”,
- „~~wziąść~~”,
- „~~conajmniej~~”,
- „~~niewiem~~”,
- ,,~~z przed/sprzed~~”,
- „~~jusz~~”,
- „~~złodzieji~~”.

Przekreślenie uzyskane z pomocą pakietu ulem oraz komendy **sout**.

2.8 Korekta

W końcowej fazie edycji konieczna jest dokładna korekta całości pracy pod kątem „literówek” (szczególnie wyrazów poprawnych słownikowo lecz nie gramatycznie, których nie wykryje sprawdzanie słownikowe), zapomnianych ogonków, „zjedzonych” lub zdublowanych wyrazów, oraz gramatyki i interpunkcji (zasady: `http://so.pwn.pl/zasady.php?id=629734`). Błędy ortograficzne (test błędów: gżegżółka, grzegżółka, gżegżułka, grzegrzuka) powinny być podkreślane w kodzie na czerwono przez słownik *Overleafa*, niestety tylko z wybranego języka.

Rozdział 3

Przykłady typowego formatowania dokumentu w LaTeX-u

Niniejszy rozdział zawiera przykłady typowego formatowania dokumentu LaTeX-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 tendencje to 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. Wolfram[1] przypuszcza, że wiele – o ile nie wszystkie – automaty komórkowe klasy czwartej są zdolne do obliczeń. Zostało to udowodnione dla reguły 110 i reguł Conweya.

3.2 Rysunki w pracy

Rysunki zasadniczo centrujemy na stronie i musimy uważać by nie wychodziły poza standardowe marginesy. Rozmiar rysunku ustalamy parametrem komendy **includegraphics**. Aby wyskalować obrazek na całą szerokość strony piszemy: **includegraphics [width=\linewidth]**. Można też skalować obrazek konkretniej, np.: **includegraphics [width=5cm]** lub **includegraphics [scale=0.4]**

Overleaf akceptuje wiele popularnych formatów graficznych zarówno wektorowych jak i rastrowych bez większych problemów. Najprostszą metodą jest skopiowanie całości kodu przykładowego rysunku (od **begin{figure}** do **end{figure}**) i podmienienie nazwy (ewentualnie ścieżki) do pliku, *label*, legendy oraz źródła. Pamiętajmy, że każdy rysunek powinien być wspomniany (**ref{}**) czy nawet omówiony w tekście pracy!

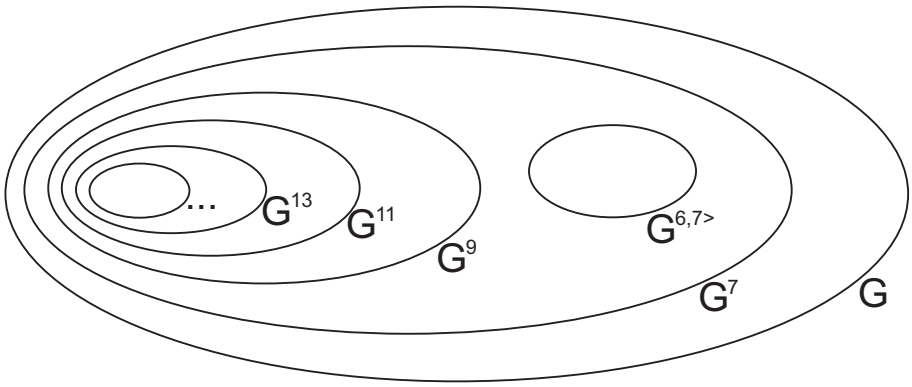
Poniżej przykład rysunku wektorowego (*eps*) - w *WinEdt* możliwe jest i *TeXify* (do *DVI*, później dopiero do *PDF*) oraz *PDFTeXify* (bezpośrednio do *PDF*), preferowane *TeXify* ponieważ *DVI* jest szybsze, w razie pozostawienia otwartego pliku *DVI* „uaktualnia" plik pokazując te same miejsce dokumentu, co jest bardzo wygodne. Ideałem jest utworzenie obrazu wektorowego i zapisanie go jako *eps* lub *ps*. Jeżeli mamy schemat wyrysowany kształtami w Wordzie czy innym programie wektorowym, prawdopodobnie możemy użyć (*BullZip PDF Printer -> Save as EPS*) do zapisania go jako eps. Mniej wygodną opcją jest wydrukowanie do pdf, gdzie musimy zwykle manualnie ustawić obcięcie marginesów

[1] Cellular Automata (1983) Stephen Wolfram

właściwej zawartości.

W ideale warto używać grafiki wektorowej do schematów, ale na pewno nie do zdjęć czy zrzutów ekranu. Te włączamy do dokumentu rastrowo, najlepiej jako png czy bmp (jpg mają zwykle nieprzyjemne zniekształcenia).

Na rysunku 3.1 pokazano przykładowy rysunek wektorowy (utworzony w *Corel Draw* i zapisany bezpośrednio jako *.eps*). Nawet przy dużym zbliżeniu zachowuje idealne kształty i kolory.



Rysunek 3.1: Podklasy funkcji harmonicznych

Źródło własne

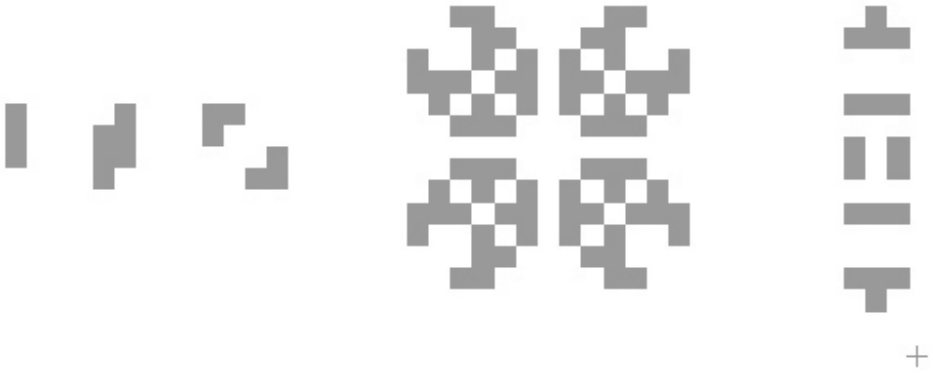
Rysunek 3.2 prezentuje dołączony plik rastrowy z kompresją jpeg (rastrowe są m.in. *jpg, png, bmp*). Niekiedy mogą powstać zniekształcenia wynikające ze skalowania lub kompresji, zalecany format to .png, bo jest kompresowany bezstratnie. Przy dużym zbliżeniu tego rysunku widoczne są niewielkie zniekształcenia i kształtów i kolorów.

Uwaga: kiedy dołączamy rastrowe obrazy, w *WinEdt* wersji 7.1 niemożliwa była opcja *TeXify* (do *PS*), trzeba było używać *PDFTeXify* (kompilacja bezpośrednio do *PDF*). Aktualna wersja programu może zachowywać się odmiennie.

3.3 Przykład tabeli

Tabela 3.1 prezentuje ustalanie *Note Importance*.

Proste tabele tworzymy ręcznie, bardziej skomplikowanie (łączenie kolumn, wierszy itp.) można wygenerować używając generatora kodu tabel LaTeX-a: `https://www.tablesgenerator.com/` albo plugina do Excela, który generuje z selekcji kod LaTeX-a tablicy: `http://www.ctan.org/tex-archive/support/excel2latex/`, opisany w `http://blog.modelworks.ch/?p=153`. W przypadki skomplikowanych tabel z dużą ilością tekstu w komórkach dla dobrych efektow trzeba się posiłkować pakietami LaTeX-a do obsługi tabel, jak np. *tabularx*.



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

Źródło `http://oscillators.org`

Tablica 3.1: Determining Note Importance

Źródło [1]

$\log_2 \frac{NoteLength}{HarmonicFragmentLength}$	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

3.4 Przykład formuły matematycznej

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

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

3.5 Kod programu jako *verbatim*

Poniżej kod programu z zachowanym formatowaniem oraz wcięciami z użyciem tabulacji czy spacji wielokrotnych (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();
};
```

Jest to środowisko **verbatim** zbliżone do standardowego tekstu, gdzie dozwolone jest łamanie strony.

3.6 Kod programu z użyciem pakietu *listing*

Jeżeli chcemy uzyskać listingi formatowane autonumerowane z numeracją wierszy i tłem można użyć np. pakietu **listing**: `https://www.overleaf.com/learn/latex/Code_listing#Captions_and_the_list_of_Listings`. Lekko zmodyfikowany przykładowy kod z wykorzystaniem tego pakietu przedstawiony jest na listingu 3.1.

Listing 3.1: Przykładowy kod w języku Python

```
1  import numpy as np
2
3  def incmatrix(genl1,genl2):
4      m = len(genl1)
5      n = len(genl2)
6      M = None #to become the incidence matrix
7      VT = np.zeros((n*m,1), int)  #dummy variable
8
9      #compute the bitwise xor matrix
10     M1 = bitxormatrix(genl1)
11     M2 = np.triu(bitxormatrix(genl2),1)
12     for i in range(m-1):
13         for j in range(i+1, m):
```

```
            [r,c] = np.where(M2 == M1[i,j])
            for k in range(len(r)):
                VT[(i)*n + r[k]] = 1;
                VT[(i)*n + c[k]] = 1;
                VT[(j)*n + r[k]] = 1;
                VT[(j)*n + c[k]] = 1;

                if M is None:
                    M = np.copy(VT)
                else:
                    M = np.concatenate((M, VT), 1)

                VT = np.zeros((n*m,1), int)

    return M
```

Listing 3.2: Przykładowa strona w formatowaniu HTML a'la NetBeans

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
   initial-scale=1.0">
  <title>Wyniki wyszukiwania</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <h1>Nazwa Twojego Sklepu</h1>
    <nav>
      <ul>
        <li><a href="#">Oferty</a></li>
        <li><a href="#">Moje konto</a></li>
        <li>
          <form action="/search" method="GET">
            <input type="text" name="query" placeholder="
   Szukaj...">
            <button type="submit">Szukaj</button>
          </form>
        </li>
      </ul>
```

```
    </nav>
  </header>

  <section id="search-results">
    <h2>Wyniki wyszukiwania dla: <span id="search-query"></span></h2>
    <!-- Tutaj będą wyświetlane wyniki wyszukiwania -->
  </section>

  <footer>
    <p>&copy; 2024 Nazwa Twojego Sklepu. Wszelkie prawa zastrzeżone.</p>
  </footer>
</body>
</html>
```

Listing 3.3: Przykład kodu XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note attribute="main" xmlns="http://www.w3.org/1999/xhtml">
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Do not forget me this weekend!</body>
</note>
```

3.7 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 jej podobnych źródeł o niepewnej reputacji i zmiennej treści) raczej nie podajemy jako bibliografii**, proszę szukać poważniejszych źródeł (dla definicji polecany jest Słownik PWN: `https://sjp.pwn.pl/`).

5. preferowane są źródła książkowe, za wyjątkiem języków programowania, pakietów czy technik, które niekiedy bardzo szybko się rozwijają.

Przykład odwołań do literatury: „Zagadnienie te zostało opisane w [2], szerzej w [3], bardzo szczegółowo w następujących trzech pracach [4, 1, 5], wreszcie podsumowane w [6].” Przykładowe referencje zdefiniowane tu w **biblio.bib** to różne formy publikacji źródeł (artykuły naukowe, książki itp.), a w szczególności przykładowy link do strony internetowej zdefiniowany jest jako **@MISC**. Należy zweryfikować finalny wygląd graficzny referencji, ponieważ niekiedy może być mocno nieprzewidywalny - uwaga na znaki specjalne LaTeX-a!. W szczególności by uniknąć zmiany liter na małe w tytułach **@ARTICLE** czy **@INPROCEEDINGS** można zastosować podwójny nawias klamrowy: **title = {{ ... }}**,.

3.8 Linki wewnętrzne i zewnętrzne

Pakiet *hyperref* automatycznie generuje linki w obrębie dokumentu do wszelkiej zawartości „referowalnej”, typu sekcje, rysunki, tabele, wzory referencje do bibliografii. Można też prosto definiować linki zewnętrzne za pomocą komendy `\url{...}`, które będą „klikalne” w docelowym dokumencie *pdf*. Dokumentacja pakietu: `http://en.wikibooks.org/wiki/LaTeX/Hyperlinks`.

Podsumowanie

Podsumowanie ma treści zbliżone do **Wstępu**, dodatkowo powinno się oceniać wykonaną pracę (spełnienie założeń i osiągnięcie celu pracy, ewentualnie co się nie udało zrealizować i dlaczego). 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 oczywiście więcej. W podsumowaniu pracy również używamy $\backslash ref\{...\}$, czyli precyzyjnych numerów rozdziałów („rozdział 3", sekcja 2.2, „we wstępie pracy", etc.) zamiast relatywnych określeń „następny", „kolejny" czy „ostatni".

I to tyle. „Koniec i bomba, a kto czytał ten trąba."[2] Powodzenia!

[2]Witold Gombrowicz, *Ferdydurke*

Bibliografia

[1] 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.

[2] 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.

[3] IMDB: Of Mice and Men. http://www.imdb.com/title/tt0105046/. [dostęp: 2025-05-08].

[4] K. Delac and M. Grgic. *Face Recognition*. I-Tech Education and Publishing, Vienna, Austria, 2007.

[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

Spis tablic

Listings