



Politechnika Gdańska
Wydział Elektroniki,
Telekomunikacji i Informatyki



Katedra:	Architektury Systemów Komputerowych
Imię i nazwisko dyplomanta:	Tomasz Boiński
Nr albumu:	999999
Forma i poziom studiów:	Stacjonarne jednolite studia magisterskie
Kierunek studiów:	Informatyka

Praca dyplomowa magisterska

Temat pracy:

Szablon pracy magisterskiej i doktorskiej, pamiętaj by odkomentować właściwy plik ze stroną tytułową!

Kierujący pracą:

prof. dr hab. inż. Jan Iksiński

Zakres pracy:

Opis pracy (jedno zdanie)

Gdańsk, 2012

Spis treści

1	Przegląd literatury	1
1.1	Wielomiany	1
1.1.1	Definicja	1
1.1.2	Podstawowe działania na wielomianach	1
1.1.3	Twierdzenia i dowody	3
2	Wstęp	5
2.1	Idea dokumentu	5
2.2	Strony tytułowe	5
2.3	Poprawność dokumentu	6
2.4	Uwagi i poprawki	6
3	Podstawowe elementy	7
3.1	Sekcje dokumentu	7
3.1.1	Podpodrozdział	7
3.2	Podstawowe elementy typograficzne	7
3.2.1	Twarda spacja	7
3.2.2	Formatowanie tekstu	7
3.3	Podział linii i paragrafy	8
3.4	Środowisko matematyczne	10
3.4.1	Twierdzenia i dowody	10
3.5	Algorytmy	11
3.6	Wpisy bibliograficzne	13

Wszystko co zostało dodane we frontmatter zostanie wydzielone do odrębnie numerowanej sekcji. Numeracja zazwyczaj jest małymi, rzymskimi liczbami. Tutaj przykład jak tekst umieścić na dole strony z wyrównaniem do prawej. Inne układy można robić analogicznie.

Podziękowania

Tu możemy dać podziękowania. Jest to zrealizowane w postaci zwykłego, nienumerowanego rozdziału. W razie konieczności należy ten rozdział po prostu usunąć.

Rozdział 1

Przegląd literatury

1.1 Wielomiany

1.1.1 Definicja

Definicja 1 *Wielomianem zmiennej rzeczywistej x nazywamy wyrażenie:*

$$W(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_{n-1}x + a_n, \quad (1.1)$$

gdzie $a_0, a_1, a_2, \dots, a_{n-1}, a_n \in R, n \in N$

Liczby $a_0, a_1, a_2, \dots, a_{n-1}, a_n$ nazywamy współczynnikami wielomianu, natomiast n nazywamy stopniem wielomianu.

Szczególnym przypadkiem wielomianu jest jednomian.

Definicja 2 *Jednomianem zmiennej rzeczywistej x nazywamy wielomian, który posiada co najwyżej jeden wyraz niezerowy i określamy wzorem:*

$$W(x) = ax^n, \text{ gdzie } a \in R, n \in N \quad (1.2)$$

Można, więc rozumieć wielomian jako skończoną sumę jednomianów. Jednomian stopnia zerowego jest stałą, pojedyncza liczba rzeczywista, która w szczególności może być zerem.

1.1.2 Podstawowe działania na wielomianach

Na wielomianach, tak jak na liczbach możemy wykonywać podstawowe działania. Należą do nich: porównywanie, dodawanie, odejmowanie, mnożenie, dzielenie, a także obliczanie reszty z dzielenia oraz NWD (największego wspólnego dzielnika). Jako, że wielomian zmiennej x możemy traktować jak funkcję jednej zmiennej, możemy także policzyć z niego pochodne.

Porównywania wielomianów

Twierdzenie 1 *Dwa wielomiany uważamy za równe wtedy i tylko wtedy, gdy są tego samego stopnia, a ich kolejne współczynniki są równe.*

Przykład 1 *Mamy dane wielomian W_1 oraz wielomian W_2 .*

$$\begin{aligned} W_1(x) &= a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n, \\ W_2(x) &= b_0x^n + b_1x^{n-1} + \dots + b_{n-1}x + b_n, \end{aligned} \quad (1.3)$$

gdzie $a, b \in R, n \in N$

Wielomiany W_1 oraz W_2 są równe wtedy i tylko wtedy gdy $\forall i \in N \ a_i = b_i$.

Suma wielomianów

Twierdzenie 2 Aby dodać dwa wielomiany należy dodać ich wyrazy podobne.

Przykład 2 Mamy dane wielomian W_1 oraz wielomian W_2 .

$$\begin{aligned} W_1(x) &= a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n, \\ W_2(x) &= b_0x^n + b_1x^{n-1} + \dots + b_{n-1}x + b_n, \\ &\text{gdzie } a, b \in R, n \in N \end{aligned} \quad (1.4)$$

Zdefiniujemy trzeci wielomian: $W_3(x) = W_1(x) + W_2(x)$. Wówczas:

$$W_3(x) = (a_0 + b_0)x^n + (a_1 + b_1)x^{n-1} + \dots + (a_{n-1} + b_{n-1})x + a_n + b_n, \quad (1.5)$$

gdzie $a, b \in R, n \in N$

Na powyższym przykładzie łatwo zaobserwować, że stopień sumy dwóch wielomianów nie może być większy od większego ze stopni dodawanych wielomianów. W przypadku gdy oba te wielomiany są tego samego stopnia, o przeciwnym współczynniku przy najwyższej potędze, to stopień ten będzie mniejszy.

Twierdzenie 3

$$\deg(W_1 + W_2) \leq \max(\deg(W_1), \deg(W_2)) \quad (1.6)$$

Różnica wielomianów

Definicja 3 Wielomianem przeciwnym nazywamy wielomian, którego wszystkie współczynniki są przeciwne do danych.

Przykład 3 Mamy dany wielomian W_1 .

$$W_1(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n, \text{ gdzie } a, b \in R, n \in N \quad (1.7)$$

Zdefiniujemy drugi wielomian: $W_2(x) = -W_1(x)$. Wówczas:

$$W_1(x) = -a_0x^n + (-a_1)x^{n-1} + \dots + (-a_{n-1})x + (-a_n), \text{ gdzie } a, b \in R, n \in N \quad (1.8)$$

Twierdzenie 4 Aby obliczyć różnicę wielomianów W_1 i W_2 , należy dodać ze sobą wielomiany W_1 i $-W_2$, czyli wielomian przeciwny do wielomianu W_2

$$y'(v) = \begin{cases} y(v) - \alpha_y & \text{jeżeli } v \in A \\ y(v) + \alpha_y & \text{jeżeli } v \in C \\ y(v) & \text{w innych przypadkach} \end{cases}$$

Zapis ten, acz skomplikowany, pozwala na reprezentację złożonych reguł matematycznych w postaci ładnie ułożonych i wyrównanych wierszy. Reguły **left** oraz **right** pozwalają na utworzenie nawiasów klamrowych, których rozmiar będzie automatycznie dostosowywany do rozmiaru elementu, jakie mają zawierać.

Rozdział 2

Wstęp

2.1 Idea dokumentu

Dokument ten ma w założeniu pomóc w pisaniu pracy magisterskiej/rozprawy doktorskiej w narzędziu jakim jest Latex. Zdecydowanie polecam jego używanie, gdyż znacznie upraszcza formatowanie i skład tekstu, wykonując bardzo dużo pracy za autora. W dalszej części tego dokumentu na przykładach przedstawię kilka technik jak zrealizować rzeczy, na jakie natknąłem się w trakcie pisania mojej pracy magisterskiej oraz rozprawy doktorskiej.

Dokument ten nie ma na celu dostarczenie kompletnej dokumentacji Latex'a, czy też zaznajomić z nim osoby nie mającego żadnej wiedzy na jego temat. Nie jest to samouczek (takich jest cała masa w sieci, np. [6], polecam też zapoznanie się z dokumentacją pakietów, jakie zostały użyte w tym dokumencie), ale raczej zbiór ciekawych konstrukcji, jakie znalazłem w trakcie pisania swojej pracy magisterskiej i rozprawy doktorskiej. Jednakże tam gdzie uznam to za stosowne będę podawał rzeczy również oczywiste, by osoby nie mające wcześniej kontaktu z Latex'em mogły, na bazie tego opracowania, rozpocząć pracę nad swoim dokumentem. Domyślnie przygotowany jest dla dwustronnego wydruku w stylu książki. Wydruk jednostronny wymaga zmiany definicji klasy z **twoside** na **oneside** i poprawienia definicji nagłówków i stopek (linie 35 – 56 preambuły dokumentu). Polecam również zaglądać do kodu dokumentu. Sam dokument bardziej pokazywać będzie efekt końcowy, jaki można uzyskać. Samą treść najłatwiej podejrzeć już w pliku źródłowym.

Do pisania samego tekstu użyć można dowolnego edytora tekstu, pozwalającego na tworzenie zwykłych plików tekstowych. Polecam jednak jakieś zintegrowane środowisko umożliwiające szybką kompilację i podgląd wynikowego dokumentu. Osobiście używam narzędzia o nazwie **Kile** dostępnego praktycznie w każdej dystrybucji systemu Linux.

Zalecam przyjrzenie się również nagłówkowi dokumentu – zawiera on importy różnych przydatnych pakietów dodatkowych wraz z opisem. Większość z nich będzie opisana lub wykorzystana w tym dokumencie.

Gorąco zachęcam do korzystania z Latex'a. Przy odrobinie wprawy jest to narzędzie wysoce wygodne i wydajne, umożliwiające zdecydowanie prostsze tworzenie złożonych dokumentów niż dowolny WYSIWIG.

2.2 Strony tytułowe

Do dokumentu dołączyłem strony tytułowe podobne do oficjalnych formatek pracy magisterskiej (`mgr_titlepage.tex`) oraz rozprawy doktorskiej (`phd_titlepage.tex`). Pierwsza z nich, będąca bazą dla drugiej, opracowana została przez mgra inż. Michała Wójcika¹.

¹<http://mwojcik.eti.pg.gda.pl>

2.3 Poprawność dokumentu

Przed drukiem warto zawsze sprawdzić swój dokument pod względem formalnej poprawności. Służy do tego narzędzie **lacheck**. Na stronach Katedry Architektury Systemów Komputerowych znajduje się interfejs WWW do tego narzędzia napisany przez mgra inż. Rafała Knozę. Narzędzie to jest dostępne pod adresem <http://kask.eti.pg.gda.pl/lacheck/>.

2.4 Uwagi i poprawki

Zachęcam wszystkich do modyfikacji, poprawiania i rozbudowy niniejszego dokumentu. Z Waszą pomocą z biegiem czasu dokument ten ma szansę stać się kompletnym przewodnikiem do Latexa pomocnym w trakcie pisania pracy magisterskiej czy doktorskiej. Wszelkie zmiany przesłane do mnie opublikuję na stronie katedralnej, oczywiście z zaznaczeniem współautorów.

Rozdział 3

Podstawowe elementy

To jest rozdział.

3.1 Sekcje dokumentu

W Latexu w klasie dokumentów **book** wyróżniamy rozdziały (**chapter**), podrozdziały **section**, podpodrozdziały **subsection**, podpodpodrozdziały **subsubsection** i paragrafy (**paragraph**). Podpodpodrozdziały i paragrafy domyślnie nie są numerowane ani nie występują w spisie treści. Zachowanie to można zmienić poprzez funkcję **setcounter** umieszczaną w preambule. Wykomentowany przykład można znaleźć w kodzie tego dokumentu.

Obeccnie znajdujemy się na poziomie podrozdziału. Pozostałe przykłady poniżej.

3.1.1 Podpodrozdział

To jest podpodrozdział.

Podpodpodrozdział

To jest podpodpodrozdział. On nie jest domyślnie numerowany.

Paragraf A to jest paragraf. On również nie jest domyślnie numerowany.

3.2 Podstawowe elementy typograficzne

3.2.1 Twarda spacja

Twarda spacja jest bardzo istotnym elementem, gdyż zabrania Latex'owi łamanie linii w miejscu jej wystąpienia, a tym samym pozwoli na niejako „sklejenie” wyrazów ze sobą. Dzięki temu możemy uniknąć tzw. sierot (pojedynczych znaków na końcu wiersza). W Latex twardą spację umieszcza się wstawiając znak tyldy (~). Zapisujemy to więc np. tak: „dokument, w~którym”.

3.2.2 Formatowanie tekstu

Aby zapewnić poprawny wygląd tekstu należy pamiętać o kilku rzeczach:

- Linia poprzedzona procentem to komentarz.
- Poprzedzaniu spacji występującej po kropce kończącej skrót znakiem ucieczki, odstęp będzie wtedy taki, jak odstęp między wyrazami a nie między zdaniami. Przykładowo zapis „np. tekst” vs. „np. tekst”. Ten drugi jest poprawny, a zapisany został tak: „np.\ tekst”.
- Skróty pisane wielkimi literami kończące zdanie powinny posiadać \@ przed kropką kończącą zdanie, np. OCS\@. Spowoduje to potraktowanie spacji jako spacji międzyzdaniowej z nie międzywyrazowej.
- Cudzysłowie zawsze tworzymy używając podwójnego przecinka jako symbolu otwierającego cudzysłów, oraz podwójnego apostrofu zamykającego cudzysłów.
- Kursywę uzyskujemy za pomocą słowa kluczowego \textit, co w efekcie daje *tekst kursywą*. Pogrubiony **używamy słowa kluczowego \textbf**. Każdorazowo tekst mający być napisany danym krojem otaczamy nawiasami klamrowymi.
- Myślnik (–) tworzymy poprzez umieszczenie bezpośrednio po sobie dwu kresek (minusy). Różnica między nimi jest zasadnicza. Pojedynczy myślnik generuje krótką kreskę (–), podwójny długą (—), potrójny najdłuższą (≡).
- Odwołania do różnych elementów dokumentu robimy poprzez słowo kluczowe **ref()**. Jako jego parametr wstawiamy nazwę zdefiniowaną za pomocą słowa kluczowego **label()**. Należy pamiętać, że odwołanie zwraca jedynie numer elementu, słowo opisowe, jak np. rozdział czy rysunek należy dodać samodzielnie. Polecam tutaj przyjąć jakąś konwencję i się jej trzymać w całym dokumencie. Tak samo należy postępować w przypadku etykiet.
- Latex doskonale radzi sobie z dzieleniem wyrazów na końcach linii, jednak czasami zachodzi konieczność wymuszenia podziału w określonym miejscu. W tym celu należy zastosować konstrukcję \-. Latex takiego ukośnika nie wydrukuję dopóty, dopóki rzeczywiście w tym miejscu nie zostanie wykonane przeniesienie części wyrazu. Możliwe jest dodanie wielu podziałów w jednym wyrazie. Użyte wtedy zostanie to, które spowoduje wygenerowania „najładniejszego” tekstu.

3.3 Podział linii i paragrafy

Nowy paragraf rozpoczyna się poprzez wstawienie jednej wolnej linii. Latex automatycznie wygeneruje wcięcie. Należy pamiętać, że pierwszy paragraf, zgodnie ze standardami drukarskimi, nie ma wcięcia! Możemy tym sterować za pomocą poleceń **noindent** (brak wcięcia) oraz **indent** (dodatkowe wcięcie).

Jeżeli chcemy po prostu zrobić nową linię, bez tworzenia nowego paragrafu używamy konstrukcji \\. Efekt będzie taki, że paragraf będzie kontynuowany w nowej linii. Nie spowoduje to jednak rozciągnięcia poprzedniej linii. Zostanie ona przerwana tam gdzie tego sobie zażyczymy i kontynuowana w nowej linii.

A co w przypadku, gdy chcemy z jakiegoś powodu przerwać linię, ale wymusić justowanie tekstu? Weźmy dla przykładu fragment:

Trzecim istotnym aspektem jest stosowana w trakcie wytwarzania ontologii metodologia pracy [4, 5]. Zastosowanie jednej z uznawanych metodologii, takich jak Methontology, NeOn czy metodologia opracowana przez Noy i McGuinness, znacząco wpływa na jakość uzyskanego produktu. Wspomniane metodologie w dużej mierze uwzględniają potrzebę przyszłej integracji wiedzy, a w połączeniu z narzędziami typu Protégé czy OCS [1, 2, 3], pozwalają na tworzenie spójnych i formalnie oraz logicznie poprawnych ontologii.

Tekst zostaje bardzo brzydko złamany w środku odnośników do cytowań. Użycie podwójnego po słowie „metodologia” w pierwszym zdaniu ukośnika da nam natomiast taki efekt:

Trzecim istotnym aspektem jest stosowana w trakcie wytwarzania ontologii metodologia pracy [4, 5]. Zastosowanie jednej z uznawanych metodologii, takich jak Methontology, NeOn czy metodologia opracowana przez Noy i McGuinness, znacząco wpływa na jakość uzyskanego produktu. Wspomniane metodologie w dużej mierze uwzględniają potrzebę przyszłej integracji wiedzy, a w połączeniu z narzędziami typu Protégé czy OCS [1, 2, 3], pozwalają na tworzenie spójnych i formalnie oraz logicznie poprawnych ontologii.

Też nie ładnie, gdyż linijka jest niewyjustowana. Z pomocą przychodzi nam tutaj komenda **linebreak[]**, gdzie w nawiasie kwadratowym podajemy liczbę od 1 do 4 określającą jak bardzo zależy nam na tym, by linia została złamana w tym miejscu (4 to najwyższa wartość). Efekt jest następujący:

Trzecim istotnym aspektem jest stosowana w trakcie wytwarzania ontologii metodologia pracy [4, 5]. Zastosowanie jednej z uznawanych metodologii, takich jak Methontology, NeOn czy metodologia opracowana przez Noy i McGuinness, znacząco wpływa na jakość uzyskanego produktu. Wspomniane metodologie w dużej mierze uwzględniają potrzebę przyszłej integracji wiedzy, a w połączeniu z narzędziami typu Protégé czy OCS [1, 2, 3], pozwalają na tworzenie spójnych i formalnie oraz logicznie poprawnych ontologii.

Jeżeli z jakiegoś powodu potrzebujemy nową linię to używamy komendy **newpage**.

Tekst występujący po niej znajdzie się na nowej stronie. Rozdziały itp. automatycznie generują nową stronę, przy czym w układzie dwustronnym nowy rozdział zawsze zaczyna się od nieparzystej strony.

3.4 Środowisko matematyczne

Środowisko matematyczne otwieramy i zamykamy znakiem $\$$. Niektóre funkcje można używać tylko wewnątrz takiego środowiska. Przykładem niech będzie funkcja **mathcal** zamieniająca duże litery w symbole o charakterystycznym kroju, stosowanym do opisywania stałych, np. \mathcal{O} czy $\mathcal{R}(\mathcal{D}, \mathcal{P}, \mathcal{T}, \mathcal{S}, \mathcal{U}, \mathcal{I})$. Pamiętać należy, że zamienione zostaną wszystkie litery w wyrażeniu występującym wewnątrz nawiasów klamrowych.

Niektóre konstrukcje, np. równania, automatycznie włączają tryb matematyczny. Równania dobrze jest opisać, przykład przedstawia Równanie 3.1.

$$\mathcal{O}(\mathcal{K}, \mathcal{B}, \mathcal{C}, \mathcal{R}) \quad (3.1)$$

gdzie:

\mathcal{K} - zbiór klas wchodzących w skład ontologii,

\mathcal{B} - zbiór bytów wchodzących w skład ontologii,

\mathcal{C} - zbiór komentarzy przypisanych do klas \mathcal{K} i bytów \mathcal{B} wchodzących w skład ontologii,

\mathcal{R} - zbiór relacji wiążących elementy ontologii.

W równaniach możemy stosować różne dodatkowe symbole oraz np. wyrównywać je do określonego miejsca. Służy do tego blok typu **split**, a sam punkt wyrównania określony jest ampersandem (&). Przykład zastosowania prezentuje równanie 3.2.

$$\begin{aligned} \forall x_1 \leq y_1, x_2 \leq y_2 : f(x_1 + x_2, y_1 + y_2) \\ = \frac{y_1}{y_1 + y_2} f(x_1, y_1) + \frac{y_2}{y_1 + y_2} f(x_2, y_2) \end{aligned} \quad (3.2)$$

3.4.1 Twierdzenia i dowody

Linie 75 – 93 nagłówka dokumentu definiują nowe nazwy sekcji twierdzeń i dowodów, oraz znacznik końca dowodu (taki czarny kwadracik). Dzięki nim można uzyskać ładnie wyglądające twierdzenia jak poniżej (Twierdzenie 6). Zauważmy, że równanie dowodu nie jest równaniem numerowanym. Wszędzie tam, gdzie nie chcemy by rozdział czy dowolna inna sekcja była numerowana należy w jej nazwie użyć gwiazdki, np. `\begin{equation*}`.

Twierdzenie 6 *Podobieństwo pomiędzy pojęciami A i B opisane jest stosunkiem ilości informacji niezbędnej do opisanie ich wspólności znaczeniowej oraz ilością informacji niezbędnej do ich opisanie (Równanie 3.3).*

$$sim_{lin}(A, B) = \frac{\log P(common(A, B))}{\log P(description(A, B))} \quad (3.3)$$

Dowód

$$\begin{aligned}
 f(x, y) &= f(x + 0, y + (y - x)) \\
 &= \frac{x}{y} * f(x, x) + \frac{y - x}{x} * f(0, y - x) \\
 &= \frac{x}{y} * 1 + \frac{y - x}{x} * 0 \\
 &= \frac{x}{y} \quad \blacksquare
 \end{aligned}$$

Inna ciekawa konstrukcja wykorzystująca tryb matematyczny do zapisania pewnego stwierdzenia:

Niech $A \subseteq T$, $C = N_y(A) \neq W$, a $\alpha_y = \min_{a \in A, b \notin C} \{y(a) + y(b) - q(a, b)\}$ oraz

$$y'(v) = \begin{cases} y(v) - \alpha_y & \text{jeżeli } v \in A \\ y(v) + \alpha_y & \text{jeżeli } v \in C \\ y(v) & \text{w innych przypadkach} \end{cases}$$

Zapis ten, acz skomplikowany, pozwala na reprezentację złożonych reguł matematycznych w postaci ładnie ułożonych i wyrównanych wierszy. Reguły **left** oraz **right** pozwalają na utworzenie nawiasów klamrowych, których rozmiar będzie automatycznie dostosowywany do rozmiaru elementu, jakie mają zawierać.

chapterRysunki, tabele i inne konstrukcje sectionRysunki Obrazki wstawiamy do dokumentu za pośrednictwem sekcji **figure**. Przykładowy wyśrodkowany rysunek z etykietą i opisem to Rys. 3.1.



Rysunek 3.1: Opis do obrazka.

3.5 Algorytmy

Pakiety **algorithmic** oraz **algorithm** dostarczają środowiska do definiowania algorytmów. Można za ich pomocą definiować zarówno pseudokod jak i algorytmy ogólne. Pseudokod przedstawia Algorytm 1. Przykładową ogólną reprezentację algorytmu Levenshteina obrazuje Algorytm 2.

Podobnie jak w przypadku rysunków, algorytmy są automatycznie rozlokowywane w tekście przez system LaTeX.

Algorytm 1 Pseudokod prezentujący jakiś bezsensowny algorytm

```
1: program FancyProgram {  
2:   function superFunkcja(List listA) {  
3:     for all element : listA.getElements() do  
4:       int result = storeElement(element);  
5:       if result == SUCCESS then  
6:         double number = result*element;  
7:       else if result == FAILURE then  
8:         double number = result/element;  
9:       end if  
10:    end for  
11:  }  
12:  input List elements;  
13:  output number;  
14:  superFunkcja(elements);  
15:  return number;  
16: }
```

Algorytm 2 Algorytm Levenshteina

1. Niech $n = \text{długość}(s)$ a $m = \text{długość}(t)$.
 2. Jeżeli $n == 0$ zwróć m i zakończ działanie.
 3. Jeżeli $m == 0$ zwróć n i zakończ działanie.
 4. Utwórz macierz d o $m + 1$ wierszach indeksowanych od 0 do m oraz $n + 1$ kolumnach indeksowanych od 0 do n , pierwszy wiersz zainicjuj wartościami od 0 do n a pierwszą kolumnę od 0 do m .
 5. Porównaj każdy znak pierwszego ciągu, indeksowany za pomocą i , z każdym znakiem drugiego ciągu, indeksowanym za pomocą j . Dla każdego porównania:
 - (a) jeżeli $s[i]$ jest identyczne z $t[j]$, $\text{koszt} = 0$, w przeciwnym wypadku $\text{koszt} = 1$,
 - (b) ustaw wartość komórki $d[i, j]$ macierzy na wartość minimalną spośród:
 - $d[i - 1, j] + 1$,
 - $d[i, j - 1] + 1$,
 - $d[i - 1, j - 1] + \text{koszt}$.
 6. Odczytaj odległość $\text{dist}_{lev}(s, t)$ z komórki $d[n, m]$.
-

3.6 Wpisy bibliograficzne

Wpisy bibliograficzne przechowujemy w odrębnym pliku z rozszerzeniem `.bib`. Przykładowy plik został dołączony do tego dokumentu. Do pliku takiego należy dodawać odpowiedni sformatowane wpisy. Latex automatycznie posortuje je po nazwiskach autorów oraz do finalnego dokumentu dołączy tylko te wpisy, które posiadają odwołania w tekście! Można więc stworzyć kompletną bazę publikacji, a Latex użyje tylko to co potrzeba. Dodatkowo, dzięki użyciu pakietu **natbib** z parametrem **sort** (patrz preambuła dokumentu), numerki w odwołaniach również zostaną posortowane, niezależnie od kolejności podania odwołań. Przykład podano w rozdziale 3.3.

Spis rysunków

3.1	Opis do obrazka.	11
-----	--------------------------	----

Spis tablic

Spis algorytmów

1	Pseudokod prezentujący jakiś bezsensowny algorytm	12
2	Algorytm Levenshteina	12

Bibliografia

- [1] T. Boiński. Architektura portalu dziedzinowego. *Praca zbiorowa Katedry Architektury Systemów Komputerowych KASKBOOK*, pages 81–92, 2008.
- [2] T. Boiński, Ł. Budnik, A. Jakowski, J. Mroziński, and K. Mazurkiewicz. OCS – domain oriented ontology creation system. *Polish Journal of Environmental Studies*, 18(3B):35–38, 2009.
- [3] T. Boiński, A. Jaworska, R. Kleczkowski, P. Kunowski, and J. Szamański. Zespołowa budowa ontologii z wykorzystaniem systemu OCS oraz edytora Protégé. *Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej*, 19:101–105, 2010.
- [4] T. Boiński, P. Orłowski, and P. Szpryngier. Inżynieria ontologii dla potrzeb integracji systemów. *Praca zbiorowa Katedry Architektury Systemów Komputerowych KASKBOOK (w publikacji)*, 2012.
- [5] T. Boiński, P. Orłowski, J. Szymański, and H. Krawczyk. Security ontology construction and integration. In *Proceedings of KEOD2011 of the 3rd International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, pages 369–374. INSTICC, 2011.
- [6] Wikibooks. LaTeX. <http://pl.wikibooks.org/wiki/LaTeX>, 2012. [Online; przeglądano 19-03-2012].