***Zaawansowane algorytmy i struktury danych   
Informatyka st. II sem. 1  
Egzamin – prezentacja – dodatek***

***Dawid Bitner***

Spis treści

[Schematy blokowe 3](#_Toc74077877)

[Kompresja algorytmem LZ77 4](#_Toc74077878)

[Kompresja algorytmem LZ78 5](#_Toc74077879)

[Kompresja algorytmem LZW 6](#_Toc74077880)

[Schematy użytkowe 7](#_Toc74077881)

[Kompresja algorytmem LZ77 7](#_Toc74077882)

[Dekompresja algorytmem LZ77 8](#_Toc74077883)

[Kompresja algorytmem LZ78 9](#_Toc74077884)

[Dekompresja algorytmem LZ78 10](#_Toc74077885)

[Kompresja algorytmem LZW 11](#_Toc74077886)

# Schematy blokowe

Wszystkie schematy zaprezentowane poniżej są w pełni edytowalne w pliku .docx. Pseudokod dotyczący algorytmów znajduje się w sprawozdaniu. Ten dokument pełni funkcję dodatku do sprawozdania.

## Kompresja algorytmem LZ77

koniec

Fałsz

Prawda

dołącz s z tyłu okna buforu

Nie

Tak

s := usuń l + 1 znak od początku wejścia

odrzuć l + 1 znak z przodu okna buforu

Wyprowadź parę <d, l, c>

c := znak następujący po prefiksie na wejściu

d := pierwszy znak wejścia

l := długość prefiksu

d := 0

l := długość prefiksu

d := odległość do początku prefiksu

Czy prefix nie jest pusty

prefiks := najdłuższy prefiks wejścia rozpoczynającego się w oknie

Wejście nie jest puste

Pobierz dane wejściowe

start

## Kompresja algorytmem LZ78

Fałsz

Opróżnij słownik

start

Nie pobrano wszystkich symboli komunikatu

koniec

Wyprowadź parę <indeks frazy f, s>

Nie

Tak

Fałsz

Prawda

f := fraza pusta

Wstaw frazę f|s do słownika

Wyprowadź parę <indeks frazy f, s>

f := f|s

czy f|s znajduje się w słowniku

Pobierz symbol s

f := fraza pusta

## Kompresja algorytmem LZW

Fałsz

Opróżnij słownik

start

koniec

Wyprowadź parę <indeks frazy f, s>

Nie

Tak

Fałsz

Prawda

f := fraza pusta

Wstaw frazę f|s do słownika

Wyprowadź parę <indeks frazy f, s>

f := f|s

If f|s znajduje się w słowniku

Pobierz symbol s

Nie pobrano wszystkich symboli komunikatu

f := fraza pusta

# Schematy użytkowe

Poniżej zaprezentowano sposoby kompresji i dekompresji za pomocą algorytmów Lempel-Ziv (LZ77, LZ78), oraz dodatkowo sposób kompresji za pomocą algorytmu Lempel-Ziv-Welch (LZW).

W każdym z przykładów posłużyłem się ciągiem znaków: **aaabbbbbbaaaabbbb.**

## Kompresja algorytmem LZ77

Dla przykładu ustalamy słownik (lewa strona) i bufor (prawa strona) na 8.

01234567 | 01234567  
aaaaaaaa | aaabbbbb -> a (A)  
**aaa**aaaaa | **aaa**bbbbb -> <0, 3, b> <- 4 (B)  
aaaaaaa**b** | **bbbbb**aaa -> <7, 5, a> <- 6 (C)  
abbbbbb**a** | **aaa**bbbb -> <7, 3, b> <- 4 (D)  
**bbb**aaaab | **bbb** -> <0, 2, b> <- 3 (E)

Wynik działania: a <0, 3, b> <7, 5, a> <7, 3, b> <0, 2, b>

1. W pierwszej kolejności wypełniamy cały słownik pierwszym znakiem z ciągu do zaszyfrowania. W typ przypadku jest to litera „a”.
2. W buforze odnajdujemy pierwszy ciąg stałych znaków. Jest to „aaa”. Odpowiada on ciągowi w słowniku, który zaczyna się od pozycji nr 0, którą wstawiamy na pierwszym miejscu w nawiasie, następnie sprawdzamy w którym miejscu w buforze występuje najbliższy znak różniący się od „a” – jest to „b” na pozycji nr 3. Liczbę 3 wpisujemy na drugim miejscu do nawiasu. Na ostatnim miejscu w nawiasie umieszczamy znak „b”. Przesuwamy w lewo o 4 (3 + 1) pozycje zestaw naszych znaków, dopisujemy do buforu 4 kolejne znaki z ciągu, który ma zostać poddany kompresji.
3. Znak „b” występuje na pozycji 7 w słowniku. Ciąg znaków w buforze zawiera się w pozycjach od 0 do 4. Kolejny różny znak występuje na pozycji 5. Zapisujemy dane do nawiasów. Ciąg przesuwamy jak w powyższym punkcie o 5 (4 + 1) znaków.
4. Znak „a” występuje na pozycji 7 w słowniku – z przeskokiem do pozycji 0. Ciąg znaków w buforze zawiera się w pozycjach od 0 do 2. Kolejny różny znak występuje na pozycji 3. Zapisujemy dane do nawiasów. Ciąg przesuwamy jak w powyższym punkcie o 4 (3 + 1) znaków. Bufor nie zostaje w pełni wypełniony, ponieważ w startowym ciągu znaków nie pozostała wystarczająca liczba znaków.
5. Znak „b” występuje na pozycjach od 0 do 2 w słowniku. Ciąg znaków w buforze zawiera się w pozycjach od 0 do 2. Kolejny różny znak występuje na pozycji 3. Zapisujemy dane do nawiasów. Ciąg przesuwamy jak w powyższym punkcie o 4 (3 + 1) znaków. Bufor nie zostaje w pełni wypełniony, ponieważ w startowym ciągu znaków nie pozostała wystarczająca liczba znaków.

## 

## Dekompresja algorytmem LZ77

Dekompresujemy ten sam przykład:  
a <0, 3, b> <7, 5, a> <7, 3, b> <0, 2, b>

01234567 | 01234567  
aaaaaaaa | a (A)  
aaaaaaaa | aaab <0, 3, b> <- 4 (B)  
aaaaaaab | bbbbba <7, 5, a> <- 6 (C)  
abbbbbba | aaab <7, 3, b> <- 4 (D)  
bbbaaaab | bbb <0, 2, b> <- 3 (E)

Wynik odczytujemy od góry: aaab bbbbba aaab bbb

1. W pierwszej kolejności wypełniamy cały słownik pierwszym znakiem z ciągu do zaszyfrowania. W typ przypadku jest to litera „a”. Odczytując „a”.
2. <0, 3, b> wypełniamy bufor w następujący sposób: od pozycji 0 do pozycji 2 (3 – 1) wypełniamy poprzednim znakiem: „a”. Na pozycji 3 wpisujemy znak b.
3. Wracając do poprzedniego kroku. Przesuwamy się, podobnie jak przy kompresji o 4 (3 + 1) znaków. <7, 5, a>. W słowniku na pozycji 7 występuje „b”. W buforze wypisujemy „b” do pozycji 4. Na pozycji 5 wpisujemy „a”.
4. Wracając do poprzedniego kroku. Przesuwamy się, podobnie jak przy kompresji o 6 (5 + 1) znaków. <7, 3, b>. W słowniku na pozycji 7 występuje „a”. W buforze wypisujemy „a” do pozycji 2. Na pozycji 3 wpisujemy „b”.
5. Wracając do poprzedniego kroku. Przesuwamy się, podobnie jak przy kompresji o 4 (3 + 1) znaków. <0, 2, b>. W słowniku na pozycji 0 występuje „b”. W buforze wypisujemy „b” do pozycji 1. Na pozycji 2 wpisujemy „b”. Jako że wpisujemy ten sam znak to następuje zakończenie algorytmu.

## Kompresja algorytmem LZ78

Dla przypomnienia, posługujemy się następującym ciągiem. **aaabbbbbbaaaabbbb.**

|  |  |  |
| --- | --- | --- |
| Index | Kodowany ciąg | Wynik |
| 1 | a | (0, a) |
| 2 | aa | (1, a) |
| 3 | b | (0, b) |
| 4 | bb | (3, b) |
| 5 | bbb | (4, b) |
| 6 | aaa | (2, a) |
| 7 | bbbb | (5, b) |

Wynik: (0, a) (1, a) (0, b) (3, b) (4, b) (2, a) (5, b).

1. Wybieramy pierwszy znak ciągu – „a”. W wyniku wpisujemy (0, a) – ponieważ słowo nie było wcześniej indeksowane – dlatego 0, drugą wartością wpisaną jest „a” – dodany znak do słowa – w tym przypadku słowo bazowe było puste („”).  
   **~~a~~**aabbbbbbaaaabbbb – skreślamy dodany znak do słownika.
2. Kolejnym słowem z nieskreślonego ciągu, który nie występuje w słowniku jest „aa”. Aby osiągnąć takie słowo musimy pobrać zawartość słowa o indeksie 1 i rozszerzyć je o „a”. Stąd zapis (1, a).  
   **~~aaa~~**bbbbbbaaaabbbb - wykreślamy ciąg.
3. Kolejnym znakiem jest „b” który w tworzonym słowniku nie występuje. W wyniku wpisujemy (0, b) – ponieważ słowo nie było wcześniej indeksowane – dlatego 0, drugą wartością wpisaną jest „b” – dodany znak do słowa – w tym przypadku słowo bazowe było puste („”).  
   **~~aaab~~**bbbbbaaaabbbb – skreślamy dodany znak do słownika.
4. Kolejnym słowem z nieskreślonego ciągu, który nie występuje w słowniku jest „bb”. Aby osiągnąć takie słowo musimy pobrać zawartość słowa o indeksie 3 i rozszerzyć je o „b”. Stąd zapis (3, b).

**~~aaabbb~~**bbbaaaabbbb - wykreślamy ciąg.

1. Kolejnym słowem z nieskreślonego ciągu, który nie występuje w słowniku jest „bbb”. Aby osiągnąć takie słowo musimy pobrać zawartość słowa o indeksie 4 i rozszerzyć je o „b”. Stąd zapis (4, b).

**~~aaabbbbbb~~**aaaabbbb - wykreślamy ciąg.

1. Kolejnym słowem z nieskreślonego ciągu, który nie występuje w słowniku jest „aaa”. Aby osiągnąć takie słowo musimy pobrać zawartość słowa o indeksie 2 i rozszerzyć je o „a”. Stąd zapis (2, a).

**~~aaabbbbbbaaaa~~**bbbb - wykreślamy ciąg.

1. Kolejnym słowem z nieskreślonego ciągu, który nie występuje w słowniku jest „bbbb”. Aby osiągnąć takie słowo musimy pobrać zawartość słowa o indeksie 5 i rozszerzyć je o „b”. Stąd zapis (5, b).

**~~aaabbbbbbaaaabbbb~~** - wykreślamy ciąg.

## Dekompresja algorytmem LZ78

Dekompresujemy przykład z kompresji algorytmem LZ77.   
Przykład: (0, a) (1, a) (0, b) (3, b) (4, b) (2, a) (5, b).

|  |  |  |
| --- | --- | --- |
| Zakodowany ciąg | Odkodowany ciąg | Index |
| (0, a) | a | 1 |
| (1, a) | aa | 2 |
| (0, b) | b | 3 |
| (3, b) | bb | 4 |
| (4, b) | bbb | 5 |
| (2, a) | aaa | 6 |
| (5, b) | bbbb | 7 |

Wynik: a aa b bb bbb aaa bbbb

1. Odczytujemy pierwszy ciąg: (0, a) – indeks zerowy oznacza, że poprzedni ciąg był pusty, rozszerzamy go o „a”. Słowo otrzymuje indeks 1. W wyniku otrzymujemy „a”.
2. (1, a) – rozszerzamy słowo o indeksie 1 o znak „a”. Słowo otrzymuje indeks 2. W wyniku otrzymujemy „aa”.
3. (0, b) – oznacza pierwsze wystąpienie ciągu „b”. indeks zerowy oznacza, że poprzedni ciąg był pusty, rozszerzamy go o „b”. Słowo otrzymuje indeks 3. W wyniku otrzymujemy „b”.
4. (3, b) – rozszerzamy słowo o indeksie 3 o znak „b”. Słowo otrzymuje indeks 4. W wyniku otrzymujemy „bb”.
5. (4, b) – rozszerzamy słowo o indeksie 4 o znak „b”. Słowo otrzymuje indeks 5. W wyniku otrzymujemy „bbb”.
6. (2, a) – rozszerzamy słowo o indeksie 2 o znak „a”. Słowo otrzymuje indeks 6. W wyniku otrzymujemy „aaa”.
7. (5, b) – rozszerzamy słowo o indeksie 5 o znak „b”. Słowo otrzymuje indeks 7. W wyniku otrzymujemy „bbbb”.

## Kompresja algorytmem LZW

Dla przypomnienia, posługujemy się następującym ciągiem. **aaabbbbbbaaaabbbb.**Algorytm jest bardzo podobny do algorytmu LZ78 – z tym, że ograniczamy się do indeksów.

|  |  |  |
| --- | --- | --- |
| encode | decode | |
| index | entry |
| - | 1 | a |
| - | 2 | b |
| 1 | 3 | aa |
| 3 | 4 | aab |
| 2 | 5 | bb |
| 5 | 6 | bbb |
| 6 | 7 | bbba |
| 3 | 8 | aaa |
| 4 | 9 | aabb |
| 6 | 10 | - |

Wynik: 1 3 2 5 6 3 4 6

1. Alfabetycznie – indeksujemy pierwszy znak, czyli „a” – pod 1.
2. Alfabetycznie – indeksujemy drugi znak, czyli „b” – pod 2.
3. „a” w słowniku występuje, więc rozszerzamy o drugi znak: czyli „aa”. aaabbbbbbaaaabbbb.
4. Przesuwamy zaznaczenie w prawo i rozszerzamy o jeden znak: aaabbbbbbaaaabbbb.
5. Pozostawiamy ostatni znak i rozszerzamy zaznaczenie o jeden. Otrzymujemy: aaabbbbbbaaaabbbb.
6. Pozostawiamy ostatni znak i rozszerzamy zaznaczenie o jeden. Otrzymujemy: aaabbbbbbaaaabbbb.
7. Pozostawiamy ostatni znak i rozszerzamy zaznaczenie o jeden i dodatkowo rozszerzamy jeszcze raz o jeden, ponieważ bez rozszerzenia otrzymalibyśmy słowo „bbb” – które już istnieje, więc: aaabbbbbbaaaabbbb.
8. Pozostawiamy ostatni znak i rozszerzamy zaznaczenie o dwa, ponieważ rozszerzając o jeden otrzymamy „aa” – a takie słowo już występuje w naszym słowniku. Otrzymujemy: aaabbbbbbaaaabbbb.
9. Pozostawiamy ostatni znak i rozszerzamy zaznaczenie o dwa, ponieważ rozszerzając o jeden otrzymamy „aab” – a takie słowo już występuje w naszym słowniku. Otrzymujemy: aaabbbbbbaaaabbbb.
10. Pozostaje nam ciąg: aaabbbbbbaaaabbbb – który występuje już pod indeksem 6 – opisujemy do wyniku.