

Kompresja bezstratna.
Kody Tunstalla. Kodowanie
arytmetyczne
Kodowanie słownikowe i predykcyjne.
Kodowanie i bezpieczeństwo informacji

Filip Zagórski

8 maja 2012

Kody Tunstalla

- Wszystkie słowa kodowe mają tą samą długość ale jeden kod może kodować różną liczbę liter alfabetu wejściowego.

Kody Tunstalla

- Wszystkie słowa kodowe mają tą samą długość ale jeden kod może kodować różną liczbę liter alfabetu wejściowego.
- Chcemy zmaksymalizować średnią liczbę symboli z pierwotnego alfabetu reprezentowanych przez słowa kodowe.

Kody Tunstalla

- Wszystkie słowa kodowe mają tę samą długość ale jeden kod może kodować różną liczbę liter alfabetu wejściowego.
- Chcemy zmaksymalizować średnią liczbę symboli z pierwotnego alfabetu reprezentowanych przez słowa kodowe.
- Alfabet wejściowy: litery a_1, a_2, \dots, a_N z prawdopodobieństwami p_1, p_2, \dots, p_N (N symboli).

Kody Tunstalla

- Wszystkie słowa kodowe mają tę samą długość ale jeden kod może kodować różną liczbę liter alfabetu wejściowego.
- Chcemy zmaksymalizować średnią liczbę symboli z pierwotnego alfabetu reprezentowanych przez słowa kodowe.
- Alfabet wejściowy: litery a_1, a_2, \dots, a_N z prawdopodobieństwami p_1, p_2, \dots, p_N (N symboli).
- Kody są długości n bitów.

Algorytm tworzenia kodów Tunstalla

- Przyporządkowujemy symbolom alfabetu N różnych słów kodowych (o długości n).

Algorytm tworzenia kodów Tunstalla

- Przyporządkowujemy symbolom alfabetu N różnych słów kodowych (o długości n).
- Dopóki liczba niewykorzystanych słów kodowych jest większa niż $N - 1$:

Algorytm tworzenia kodów Tunstalla

- Przyporządkowujemy symbolom alfabetu N różnych słów kodowych (o długości n).
- Dopóki liczba niewykorzystanych słów kodowych jest większa niż $N - 1$:
 - wybierz słowo kodowe e odpowiadające ciągowi o największym prawdopodobieństwie;

Algorytm tworzenia kodów Tunstalla

- Przyporządkowujemy symbolom alfabetu N różnych słów kodowych (o długości n).
- Dopóki liczba niewykorzystanych słów kodowych jest większa niż $N - 1$:
 - wybierz słowo kodowe e odpowiadające ciągowi o największym prawdopodobieństwie;
 - usuń e z kodu;

Algorytm tworzenia kodów Tunstalla

- Przyporządkowujemy symbolom alfabetu N różnych słów kodowych (o długości n).
- Dopóki liczba niewykorzystanych słów kodowych jest większa niż $N - 1$:
 - wybierz słowo kodowe e odpowiadające ciągowi o największym prawdopodobieństwie;
 - usuń e z kodu;
 - dodaj do kodu ciągi powstałe z dodania a_1, \dots, a_n jako sufiksów ciągu odpowiadającego kodowi e (przypisz im odpowiednie prawdopodobieństwa).

Algorytm tworzenia kodów Tunstalla

- Przyporządkowujemy symbolom alfabetu N różnych słów kodowych (o długości n).
- Dopóki liczba niewykorzystanych słów kodowych jest większa niż $N - 1$:
 - wybierz słowo kodowe e odpowiadające ciągowi o największym prawdopodobieństwie;
 - usuń e z kodu;
 - dodaj do kodu ciągi powstałe z dodania a_1, \dots, a_n jako sufiksów ciągu odpowiadającego kodowi e (przypisz im odpowiednie prawdopodobieństwa).

Co najmniej jedno słowo kodowe zostanie niewykorzystane

Przykład

- Weźmy alfabet a, b, c ($N = 3$)
z prawdopodobieństwami $P(a) = 0.6$, $P(b) = 0.3$
i $P(c) = 0.1$.

Przykład

- Weźmy alfabet a, b, c ($N = 3$)
z prawdopodobieństwami $P(a) = 0.6$, $P(b) = 0.3$
i $P(c) = 0.1$.
- Ustalmy $n = 3$ (8 słów kodowych).

Przykład

- Weźmy alfabet a, b, c ($N = 3$)
z prawdopodobieństwami $P(a) = 0.6$, $P(b) = 0.3$
i $P(c) = 0.1$.
- Ustalmy $n = 3$ (8 słów kodowych).
- Mamy 3 słowa kodowe odpowiadające $a \rightarrow 0.6$,
 $b \rightarrow 0.3$ i $c \rightarrow 0.1$.

Przykład

- Weźmy alfabet a, b, c ($N = 3$)
z prawdopodobieństwami $P(a) = 0.6$, $P(b) = 0.3$
i $P(c) = 0.1$.
- Ustalmy $n = 3$ (8 słów kodowych).
- Mamy 3 słowa kodowe odpowiadające $a \rightarrow 0.6$,
 $b \rightarrow 0.3$ i $c \rightarrow 0.1$.
- Zastępujemy a i otrzymujemy 5 kodów: $aa \rightarrow 0.36$,
 $ab \rightarrow 0.18$, $ac \rightarrow 0.06$, $b \rightarrow 0.3$ $c \rightarrow 0.1$.

Przykład

- Weźmy alfabet a, b, c ($N = 3$)
z prawdopodobieństwami $P(a) = 0.6$, $P(b) = 0.3$
i $P(c) = 0.1$.
- Ustalmy $n = 3$ (8 słów kodowych).
- Mamy 3 słowa kodowe odpowiadające $a \rightarrow 0.6$,
 $b \rightarrow 0.3$ i $c \rightarrow 0.1$.
- Zastępujemy a i otrzymujemy 5 kodów: $aa \rightarrow 0.36$,
 $ab \rightarrow 0.18$, $ac \rightarrow 0.06$, $b \rightarrow 0.3$ i $c \rightarrow 0.1$.
- Zastępujemy aa i otrzymujemy 7 kodów:
 $aaa \rightarrow 0.216$, $aab \rightarrow 0.108$, $aac \rightarrow 0.036$,
 $ab \rightarrow 0.18$, $ac \rightarrow 0.06$, $b \rightarrow 0.3$ i $c \rightarrow 0.1$.

Przykład

- Weźmy alfabet a, b, c ($N = 3$)
z prawdopodobieństwami $P(a) = 0.6$, $P(b) = 0.3$
i $P(c) = 0.1$.
- Ustalmy $n = 3$ (8 słów kodowych).
- Mamy 3 słowa kodowe odpowiadające $a \rightarrow 0.6$,
 $b \rightarrow 0.3$ i $c \rightarrow 0.1$.
- Zastępujemy a i otrzymujemy 5 kodów: $aa \rightarrow 0.36$,
 $ab \rightarrow 0.18$, $ac \rightarrow 0.06$, $b \rightarrow 0.3$ i $c \rightarrow 0.1$.
- Zastępujemy aa i otrzymujemy 7 kodów:
 $aaa \rightarrow 0.216$, $aab \rightarrow 0.108$, $aac \rightarrow 0.036$,
 $ab \rightarrow 0.18$, $ac \rightarrow 0.06$, $b \rightarrow 0.3$ i $c \rightarrow 0.1$.

aaa	aab	aac	ab	ac	b	c	???
000	001	010	011	100	101	110	111

Przykład

aaa	aab	aac	ab	ac	b	c	???
000	001	010	011	100	101	110	111

Przykład

aaa	aab	aac	ab	ac	b	c	???
000	001	010	011	100	101	110	111

- Zakodujemy tekst *abcaabb*aa.

Przykład

aaa	aab	aac	ab	ac	b	c	???
000	001	010	011	100	101	110	111

- Zakodujemy tekst *abcaabbaa*.
- Otrzymujemy: 001110001101??.

Przykład

aaa	aab	aac	ab	ac	b	c	???
000	001	010	011	100	101	110	111

- Zakodujemy tekst *abcaabbaa*.
- Otrzymujemy: 001110001101??.
- Na końcu tekstu może pojawić się blok dla którego nie ma słowa kodowego, wtedy wysyłamy specjalny kod i normalne kody liter.

Przykład

aaa	aab	aac	ab	ac	b	c	???
000	001	010	011	100	101	110	111

- Zakodujemy tekst *abcaabbaa*.
- Otrzymujemy: 001110001101??.
- Na końcu tekstu może pojawić się blok dla którego nie ma słowa kodowego, wtedy wysyłamy specjalny kod i normalne kody liter.
- 001110001101111*kod(a)kod(a)*

Średnia długość

- Średnia ilość bitów na jeden symbol wejściowy.

$$\sum_{i=1}^{2^n-1} P(e_i) \frac{n}{|e_i|},$$

gdzie e_i - słowo odpowiadające i -temu kodowi.

Średnia długość

- Średnia ilość bitów na jeden symbol wejściowy.

$$\sum_{i=1}^{2^n-1} P(e_i) \frac{n}{|e_i|},$$

gdzie e_i - słowo odpowiadające i -temu kodowi.

- Dla przykładu z poprzedniego slajdu:

$$\begin{aligned} & (0.216 + 0.108 + 0.036) \frac{3}{3} + \\ & (0.18 + 0.06) \frac{3}{2} + \\ & (0.3 + 0.1) \frac{3}{1} = 1.92 \end{aligned}$$

Średnia długość

- Średnia ilość bitów na jeden symbol wejściowy.

$$\sum_{i=1}^{2^n-1} P(e_i) \frac{n}{|e_i|},$$

gdzie e_i - słowo odpowiadające i -temu kodowi.

- Dla przykładu z poprzedniego slajdu:

$$(0.216 + 0.108 + 0.036) \frac{3}{3} +$$

$$(0.18 + 0.06) \frac{3}{2} +$$

$$(0.3 + 0.1) \frac{3}{1} = 1.92$$

- Średnia długość kodu Huffmana dla tego przypadku to 1.4.

Kody Tunstalla - podsumowanie

- Zmienna długość bloków wejściowych, stała długość wyjściowych.

Kody Tunstalla - podsumowanie

- Zmienna długość bloków wejściowych, stała długość wyjściowych.
- Kompresja i odporność na przekłamania.

Kody Tunstalla - podsumowanie

- Zmienna długość bloków wejściowych, stała długość wyjściowych.
- Kompresja i odporność na przekłamania.
- Jednoznaczność kodowania i dekodowania.

Kodowanie arytmetyczne

- Tekst wejściowy zostaje odwzorowany na liczbę z przedziału $[0, 1)$.

Kodowanie arytmetyczne

- Tekst wejściowy zostaje odwzorowany na liczbę z przedziału $[0, 1)$.
- Kod tekstu to liczba n - długość tekstu i liczba z (znacznik) reprezentowany z odpowiednio dobraną dokładnością.

Kodowanie arytmetyczne

- Tekst wejściowy zostaje odwzorowany na liczbę z przedziału $[0, 1)$.
- Kod tekstu to liczba n - długość tekstu i liczba z (znacznik) reprezentowany z odpowiednio dobraną dokładnością.
- Elementom alfabetu a_1, \dots, a_N z prawdopodobieństwami p_1, \dots, p_N przyporządkowujemy przedział $[F(i), F(i + 1))$, gdzie $F(i) = \sum_{j=1}^{i-1} p_j$.

- Mamy ciąg liter $x_1 x_2 \dots x_m$ z alfabetu a_1, \dots, a_N .

- Mamy ciąg liter $x_1 x_2 \dots x_m$ z alfabetu a_1, \dots, a_N .
- Na początku przedział $[l, p) = [0, 1)$.

- Mamy ciąg liter $x_1 x_2 \dots x_m$ z alfabetu a_1, \dots, a_N .
- Na początku przedział $[l, p) = [0, 1)$.
- Dla $i = 1, 2, \dots, m$:

- Mamy ciąg liter $x_1 x_2 \dots x_m$ z alfabetu a_1, \dots, a_N .
- Na początku przedział $[l, p) = [0, 1)$.
- Dla $i = 1, 2, \dots, m$:
 - Niech $x_i = a_j$.

- Mamy ciąg liter $x_1 x_2 \dots x_m$ z alfabetu a_1, \dots, a_N .
- Na początku przedział $[l, p) = [0, 1)$.
- Dla $i = 1, 2, \dots, m$:
 - Niech $x_i = a_j$.
 - Wtedy $d \leftarrow p - l$, $p \leftarrow l + F(j + 1)d$ i $l \leftarrow l + F(j)d$.

- Mamy ciąg liter $x_1 x_2 \dots x_m$ z alfabetu a_1, \dots, a_N .
- Na początku przedział $[l, p) = [0, 1)$.
- Dla $i = 1, 2, \dots, m$:
 - Niech $x_i = a_j$.
 - Wtedy $d \leftarrow p - l$, $p \leftarrow l + F(j+1)d$ i $l \leftarrow l + F(j)d$.
- Znacznik to dowolna liczba z przedziału $[l, p)$, np.
 $z = (l + p)/2$.

Przykład

- Weźmy alfabet a, b, c z prawdopodobieństwami 0.7, 0.1, 0.2.

Przykład

- Weźmy alfabet a, b, c z prawdopodobieństwami 0.7, 0.1, 0.2.
- Zakodujmy tekst abc .

Przykład

- Weźmy alfabet a, b, c z prawdopodobieństwami 0.7, 0.1, 0.2.
- Zakodujmy tekst abc .
- Na początku mamy przedział: $[0, 1)$.

Przykład

- Weźmy alfabet a, b, c z prawdopodobieństwami 0.7, 0.1, 0.2.
- Zakodujemy tekst abc .
- Na początku mamy przedział: $[0, 1)$.
- $F(1) = 0$, $F(2) = 0.7$, $F(3) = 0.8$, $F(4) = 1$.

Przykład

- Weźmy alfabet a, b, c z prawdopodobieństwami 0.7, 0.1, 0.2.
- Zakodujemy tekst abc .
- Na początku mamy przedział: $[0, 1)$.
- $F(1) = 0$, $F(2) = 0.7$, $F(3) = 0.8$, $F(4) = 1$.
- Kodujemy a i otrzymujemy przedział $[0, 0.7)$.

Przykład

- Weźmy alfabet a, b, c z prawdopodobieństwami 0.7, 0.1, 0.2.
- Zakodujemy tekst abc .
- Na początku mamy przedział: $[0, 1)$.
- $F(1) = 0, F(2) = 0.7, F(3) = 0.8, F(4) = 1$.
- Kodujemy a i otrzymujemy przedział $[0, 0.7)$.
- Kodujemy b i otrzymujemy przedział $[0.49, 0.56)$.

Przykład

- Weźmy alfabet a, b, c z prawdopodobieństwami 0.7, 0.1, 0.2.
- Zakodujemy tekst abc .
- Na początku mamy przedział: $[0, 1)$.
- $F(1) = 0, F(2) = 0.7, F(3) = 0.8, F(4) = 1$.
- Kodujemy a i otrzymujemy przedział $[0, 0.7)$.
- Kodujemy b i otrzymujemy przedział $[0.49, 0.56)$.
- Kodujemy c i otrzymujemy przedział $[0.546, 0.56)$.

Przykład

- Weźmy alfabet a, b, c z prawdopodobieństwami 0.7, 0.1, 0.2.
- Zakodujemy tekst abc .
- Na początku mamy przedział: $[0, 1)$.
- $F(1) = 0, F(2) = 0.7, F(3) = 0.8, F(4) = 1$.
- Kodujemy a i otrzymujemy przedział $[0, 0.7)$.
- Kodujemy b i otrzymujemy przedział $[0.49, 0.56)$.
- Kodujemy c i otrzymujemy przedział $[0.546, 0.56)$.
- Za znacznik możemy przyjąć 0.553.

- Dla ustalonej długości tekstu n , każdy ciąg jest odwzorowany na przedział rozłączny z przedziałami odpowiadającymi innym ciągom. Gwarantuje to jednoznaczność kodowania.

- Dla ustalonej długości tekstu n , każdy ciąg jest odwzorowany na przedział rozłączny z przedziałami odpowiadającymi innym ciągom. Gwarantuje to jednoznaczność kodowania.
- Wygenerowanie znacznika dla konkretnego ciągu nie wymaga wyznaczania bądź pamiętania znaczników innych ciągów.

- Dostajemy n - długość tekstu i z - znacznik tekstu.

Dekodowanie

- Dostajemy n - długość tekstu i z - znacznik tekstu.
- $l \leftarrow 0$ i $p \leftarrow 1$.

Dekodowanie

- Dostajemy n - długość tekstu i z - znacznik tekstu.
- $l \leftarrow 0$ i $p \leftarrow 1$.
- Dla $i = 1, 2, \dots, n$:

- Dostajemy n - długość tekstu i z - znacznik tekstu.
- $l \leftarrow 0$ i $p \leftarrow 1$.
- Dla $i = 1, 2, \dots, n$:
 - Wybieramy j takie, że
$$l + F(j)(p - l) \leq z < l + F(j + 1)(p - l);$$

- Dostajemy n - długość tekstu i z - znacznik tekstu.
- $l \leftarrow 0$ i $p \leftarrow 1$.
- Dla $i = 1, 2, \dots, n$:
 - Wybieramy j takie, że
$$l + F(j)(p - l) \leq z < l + F(j + 1)(p - l);$$
 - Przyjmujemy, że $x_i = a_j$;

- Dostajemy n - długość tekstu i z - znacznik tekstu.
- $l \leftarrow 0$ i $p \leftarrow 1$.
- Dla $i = 1, 2, \dots, n$:
 - Wybieramy j takie, że
$$l + F(j)(p - l) \leq z < l + F(j + 1)(p - l);$$
 - Przyjmujemy, że $x_i = a_j$;
 - $d \leftarrow p - l$, $p \leftarrow l + F(j + 1)d$ i $l \leftarrow l + F(j)d$.

- Dostajemy n - długość tekstu i z - znacznik tekstu.
- $l \leftarrow 0$ i $p \leftarrow 1$.
- Dla $i = 1, 2, \dots, n$:
 - Wybieramy j takie, że
$$l + F(j)(p - l) \leq z < l + F(j + 1)(p - l);$$
 - Przyjmujemy, że $x_i = a_j$;
 - $d \leftarrow p - l$, $p \leftarrow l + F(j + 1)d$ i $l \leftarrow l + F(j)d$.
- Ciąg oryginalny to x_1, \dots, x_n .

Przykład

- $P(a) = 0.7, P(b) = 0.1, P(c) = 0.2, z = 0.55$ i $n = 3$.

Przykład

- $P(a) = 0.7, P(b) = 0.1, P(c) = 0.2, z = 0.55$ i $n = 3$.
- $F(1) = 0, F(2) = 0.7, F(3) = 0.8$ i $F(4) = 1$.

Przykład

- $P(a) = 0.7, P(b) = 0.1, P(c) = 0.2, z = 0.55$ i $n = 3$.
- $F(1) = 0, F(2) = 0.7, F(3) = 0.8$ i $F(4) = 1$.
- $l = 0$ i $p = 1$.

Przykład

- $P(a) = 0.7, P(b) = 0.1, P(c) = 0.2, z = 0.55$ i $n = 3$.
- $F(1) = 0, F(2) = 0.7, F(3) = 0.8$ i $F(4) = 1$.
- $l = 0$ i $p = 1$.
- Dla a mamy $0 \leq 0.55 < 0.7$, stąd $x_1 = a, l = 0$ i $p = 0.7$.

Przykład

- $P(a) = 0.7, P(b) = 0.1, P(c) = 0.2, z = 0.55$ i $n = 3$.
- $F(1) = 0, F(2) = 0.7, F(3) = 0.8$ i $F(4) = 1$.
- $l = 0$ i $p = 1$.
- Dla a mamy $0 \leq 0.55 < 0.7$, stąd $x_1 = a, l = 0$ i $p = 0.7$.
- Dla b mamy $0.49 \leq 0.55 < 0.56$, stąd $x_2 = b, l = 0.49$ i $p = 0.56$.

Przykład

- $P(a) = 0.7, P(b) = 0.1, P(c) = 0.2, z = 0.55$ i $n = 3$.
- $F(1) = 0, F(2) = 0.7, F(3) = 0.8$ i $F(4) = 1$.
- $l = 0$ i $p = 1$.
- Dla a mamy $0 \leq 0.55 < 0.7$, stąd $x_1 = a, l = 0$ i $p = 0.7$.
- Dla b mamy $0.49 \leq 0.55 < 0.56$, stąd $x_2 = b, l = 0.49$ i $p = 0.56$.
- Dla c mamy $0.546 \leq 0.55 < 0.56$, stąd $x_3 = c, l = 0.546$ i $p = 0.56$.

Przykład

- $P(a) = 0.7, P(b) = 0.1, P(c) = 0.2, z = 0.55$ i $n = 3$.
- $F(1) = 0, F(2) = 0.7, F(3) = 0.8$ i $F(4) = 1$.
- $l = 0$ i $p = 1$.
- Dla a mamy $0 \leq 0.55 < 0.7$, stąd $x_1 = a, l = 0$ i $p = 0.7$.
- Dla b mamy $0.49 \leq 0.55 < 0.56$, stąd $x_2 = b, l = 0.49$ i $p = 0.56$.
- Dla c mamy $0.546 \leq 0.55 < 0.56$, stąd $x_3 = c, l = 0.546$ i $p = 0.56$.
- Odkodowany ciąg to *abc*

- Jak reprezentować znacznik (liczba rzeczywista) aby był jak najkrótszy.

- Jak reprezentować znacznik (liczba rzeczywista) aby był jak najkrótszy.
- Niech $x = x_1, \dots, x_n$ będzie ciągiem danych o prawdopodobieństwie wystąpienia $P(x) = \prod_{i=1}^n P(x_i)$. Zaokrąglenie z' znacznika z do

$$m(x) = \left\lceil \log \frac{1}{P(X)} \right\rceil + 1$$

bitów gwarantuje jednoznaczność kodowania.

- Jak reprezentować znacznik (liczba rzeczywista) aby był jak najkrótszy.
- Niech $x = x_1, \dots, x_n$ będzie ciągiem danych o prawdopodobieństwie wystąpienia $P(x) = \prod_{i=1}^n P(x_i)$. Zaokrąglenie z' znacznika z do

$$m(x) = \left\lceil \log \frac{1}{P(X)} \right\rceil + 1$$

bitów gwarantuje jednoznaczność kodowania.

- Kod arytmetyczny dla ustalonej długości tekstu jest kodem prefiksowym.

Przykład

- $P(a) = 0.7, P(b) = 0.1, P(c) = 0.2.$

Przykład

- $P(a) = 0.7, P(b) = 0.1, P(c) = 0.2.$
- Kod dla tekstu abc to $0.553 = (0.100011011)_2.$

Przykład

- $P(a) = 0.7, P(b) = 0.1, P(c) = 0.2$.
- Kod dla tekstu abc to $0.553 = (0.100011011)_2$.
- $P(abc) = 0.014$.

Przykład

- $P(a) = 0.7, P(b) = 0.1, P(c) = 0.2.$
- Kod dla tekstu abc to $0.553 = (0.100011011)_2.$
- $P(abc) = 0.014.$
- $\lceil \log_{0.014} 1 \rceil + 1 = 8$

Przykład

- $P(a) = 0.7, P(b) = 0.1, P(c) = 0.2$.
- Kod dla tekstu abc to $0.553 = (0.100011011)_2$.
- $P(abc) = 0.014$.
- $\lceil \log_{0.014} 1 \rceil + 1 = 8$
- Czyli do zakodowania tekstu wystarczy wysłać 10001101.

- Dla długich ciągów potrzebujemy długich liczb i przetwarzanie wymaga przeczytania całego ciągu.

- Dla długich ciągów potrzebujemy długich liczb i przetwarzanie wymaga przeczytania całego ciągu.
- Można zmodyfikować algorytm do pracy przyrostowej - znacznik powstaje etapami i można wysyłać go fragmentami.

Kodowanie ze skalowaniem

Po zakodowaniu kolejnej litery:

- Jeśli $[l, p) \subseteq [0, 0.5)$:

Kodowanie ze skalowaniem

Po zakodowaniu kolejnej litery:

- Jeśli $[l, p) \subseteq [0, 0.5)$:
 - zamień $[l, p)$ na $[2l, 2p)$;

Kodowanie ze skalowaniem

Po zakodowaniu kolejnej litery:

- Jeśli $[l, p) \subseteq [0, 0.5)$:
 - zamień $[l, p)$ na $[2l, 2p)$;
 - dołącz do kodu słowo 01^{licznik} ;

Kodowanie ze skalowaniem

Po zakodowaniu kolejnej litery:

- Jeśli $[l, p) \subseteq [0, 0.5)$:
 - zamień $[l, p)$ na $[2l, 2p)$;
 - dołącz do kodu słowo 01^{licznik} ;
 - $\text{licznik} \leftarrow 0$.

Kodowanie ze skalowaniem

Po zakodowaniu kolejnej litery:

- Jeśli $[l, p) \subseteq [0, 0.5)$:
 - zamień $[l, p)$ na $[2l, 2p)$;
 - dołącz do kodu słowo 01^{licznik} ;
 - $\text{licznik} \leftarrow 0$.
- Jeśli $[l, p) \subseteq [0.5, 1)$:

Kodowanie ze skalowaniem

Po zakodowaniu kolejnej litery:

- Jeśli $[l, p) \subseteq [0, 0.5)$:
 - zamień $[l, p)$ na $[2l, 2p)$;
 - dołącz do kodu słowo 01^{licznik} ;
 - $\text{licznik} \leftarrow 0$.
- Jeśli $[l, p) \subseteq [0.5, 1)$:
 - zamień $[l, p)$ na $[2l - 1, 2p - 1)$;

Kodowanie ze skalowaniem

Po zakodowaniu kolejnej litery:

- Jeśli $[l, p) \subseteq [0, 0.5)$:
 - zamień $[l, p)$ na $[2l, 2p)$;
 - dołącz do kodu słowo 01^{licznik} ;
 - $\text{licznik} \leftarrow 0$.
- Jeśli $[l, p) \subseteq [0.5, 1)$:
 - zamień $[l, p)$ na $[2l - 1, 2p - 1)$;
 - dołącz do kodu słowo 10^{licznik} ;

Kodowanie ze skalowaniem

Po zakodowaniu kolejnej litery:

- Jeśli $[l, p) \subseteq [0, 0.5)$:
 - zamień $[l, p)$ na $[2l, 2p)$;
 - dołącz do kodu słowo 01^{licznik} ;
 - $\text{licznik} \leftarrow 0$.
- Jeśli $[l, p) \subseteq [0.5, 1)$:
 - zamień $[l, p)$ na $[2l - 1, 2p - 1)$;
 - dołącz do kodu słowo 10^{licznik} ;
 - $\text{licznik} \leftarrow 0$.

Kodowanie ze skalowaniem

Po zakodowaniu kolejnej litery:

- Jeśli $[l, p) \subseteq [0, 0.5)$:
 - zamień $[l, p)$ na $[2l, 2p)$;
 - dołącz do kodu słowo 01^{licznik} ;
 - $\text{licznik} \leftarrow 0$.
- Jeśli $[l, p) \subseteq [0.5, 1)$:
 - zamień $[l, p)$ na $[2l - 1, 2p - 1)$;
 - dołącz do kodu słowo 10^{licznik} ;
 - $\text{licznik} \leftarrow 0$.
- Jeśli $l < 0.5 < p$ i $[l, p) \subseteq [0.25, 0.75)$:

Kodowanie ze skalowaniem

Po zakodowaniu kolejnej litery:

- Jeśli $[l, p) \subseteq [0, 0.5)$:
 - zamień $[l, p)$ na $[2l, 2p)$;
 - dołącz do kodu słowo 01^{licznik} ;
 - $\text{licznik} \leftarrow 0$.
- Jeśli $[l, p) \subseteq [0.5, 1)$:
 - zamień $[l, p)$ na $[2l - 1, 2p - 1)$;
 - dołącz do kodu słowo 10^{licznik} ;
 - $\text{licznik} \leftarrow 0$.
- Jeśli $l < 0.5 < p$ i $[l, p) \subseteq [0.25, 0.75)$:
 - zamień $[l, p)$ na $[2l - 0.5, 2p - 0.5)$;

Kodowanie ze skalowaniem

Po zakodowaniu kolejnej litery:

- Jeśli $[l, p) \subseteq [0, 0.5)$:
 - zamień $[l, p)$ na $[2l, 2p)$;
 - dołącz do kodu słowo 01^{licznik} ;
 - $\text{licznik} \leftarrow 0$.
- Jeśli $[l, p) \subseteq [0.5, 1)$:
 - zamień $[l, p)$ na $[2l - 1, 2p - 1)$;
 - dołącz do kodu słowo 10^{licznik} ;
 - $\text{licznik} \leftarrow 0$.
- Jeśli $l < 0.5 < p$ i $[l, p) \subseteq [0.25, 0.75)$:
 - zamień $[l, p)$ na $[2l - 0.5, 2p - 0.5)$;
 - $\text{licznik} \leftarrow \text{licznik} + 1$.

Kodowanie ze skalowaniem

Po zakodowaniu kolejnej litery:

- Jeśli $[l, p) \subseteq [0, 0.5)$:
 - zamień $[l, p)$ na $[2l, 2p)$;
 - dołącz do kodu słowo 01^{licznik} ;
 - $\text{licznik} \leftarrow 0$.
- Jeśli $[l, p) \subseteq [0.5, 1)$:
 - zamień $[l, p)$ na $[2l - 1, 2p - 1)$;
 - dołącz do kodu słowo 10^{licznik} ;
 - $\text{licznik} \leftarrow 0$.
- Jeśli $l < 0.5 < p$ i $[l, p) \subseteq [0.25, 0.75)$:
 - zamień $[l, p)$ na $[2l - 0.5, 2p - 0.5)$;
 - $\text{licznik} \leftarrow \text{licznik} + 1$.

Analogicznie można zmodyfikować procedurę dekodowania aby dekodowanie odbywało się na podstawie otrzymywanych fragmentów.

Dalsze możliwe usprawnienia

- Przejście z arytmetyki zmiennoprzecinkowej na arytmetykę całkowitoliczbową: zastąpienie przedziału $[0, 1)$ przez przedział liczb całkowitych $[0, 2^m - 1]$.

Dalsze możliwe usprawnienia

- Przejście z arytmetyki zmiennoprzecinkowej na arytmetykę całkowitoliczbową: zastąpienie przedziału $[0, 1)$ przez przedział liczb całkowitych $[0, 2^m - 1]$.
- Problem: Jak dobrać m aby uniknąć błędów zaokrągleń.

Algorytmy słownikowe - Motywacje

- Dane, które kompresujemy, bardzo często nie tworzą ciągu wartości niezależnych (kolejne symbole są zależne od poprzednich).
- Pewne ciągi (słowa) powtarzają się bardzo często.

Słowniki statyczne

- Mamy ustalony stały słownik.
- Tekst kodujemy jako ciąg pozycji ze słownika.
- Co z elementami których nie ma w słowniku?
Można np. umieścić pojedyncze litery w słowniku.
- Niestety słowniki statyczne są wrażliwe na zmianę charakteru danych.

Przykład: kodowanie tekstów programów - statyczny słownik słów kluczowych.

Kodowanie digramowe

- Ustalamy wielkość słownika np. 256 (2^8)
- Słownik składa się ze wszystkich liter i tylu par liter (digramów), ile się jeszcze zmieści (najbardziej prawdopodobne pary).
- Na przykład w kodzie ASCII używamy przeważnie tylko 95 znaków (znaki drukowalne) i pozostałe 161 możemy zastąpić najczęściej występującymi parami.

Słowniki dynamiczne

- Słownik tworzymy w trakcie kodowania.
- Słownik dostosowuje się do charakteru danych.
- Dekoder może go odtworzyć w oparciu o odkodowaną część danych (nie trzeba go dołączać do danych).

- **Idea:** słownikiem jest zakodowana/odkodowana część tekstu.
- Dla zakodowanej części długości n i niezakodowanej długości m , czyli dla ciągu $x_1 \dots x_n x_{n+1} \dots x_{n+m}$ szukamy najdłuższego podśłowa w ciągu $x_1 \dots x_n$ będące prefiksem ciągu $x_{n+1} \dots x_{n+m}$ (dopasowanie).
- Jako kod podajemy trzy liczby:
 - wielkość przesunięcia w lewo
 - ilość kopiowanych znaków
 - kod pierwszej niepasującej litery.
- Jeśli pojawia się nowa litera której nie można znaleźć w zakodowanej części to wysyłamy trójkę $(0, 0, kod)$

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- wabba-wabba-wabba-woo-woo-woo

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- wabba-wabba-wabba-woo-woo-woo (0,0,k(w))

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- wabba-wabba-wabba-woo-woo-woo
- wabba-wabba-wabba-woo-woo-woo

(0,0,k(w))

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- wabba-wabba-wabba-woo-woo-woo (0,0,k(w))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(a))

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- wabba-wabba-wabba-woo-woo-woo (0,0,k(w))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(a))
- wabba-wabba-wabba-woo-woo-woo

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- wabba-wabba-wabba-woo-woo-woo (0,0,k(w))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(b))

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- wabba-wabba-wabba-woo-woo-woo (0,0,k(w))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(b))
- wabba-wabba-wabba-woo-woo-woo

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- wabba-wabba-wabba-woo-woo-woo (0,0,k(w))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(b))
- wabba-wabba-wabba-woo-woo-woo (1,1,k(a))

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- wabba-wabba-wabba-woo-woo-woo (0,0,k(w))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(b))
- wabba-wabba-wabba-woo-woo-woo (1,1,k(a))
- wabba-wabba-wabba-woo-woo-woo

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- wabba-wabba-wabba-woo-woo-woo (0,0,k(w))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(b))
- wabba-wabba-wabba-woo-woo-woo (1,1,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(-))

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- wabba-wabba-wabba-woo-woo-woo (0,0,k(w))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(b))
- wabba-wabba-wabba-woo-woo-woo (1,1,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(-))
- wabba-wabba-wabba-woo-woo-woo

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- wabba-wabba-wabba-woo-woo-woo (0,0,k(w))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(b))
- wabba-wabba-wabba-woo-woo-woo (1,1,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(-))
- wabba-wabba-wabba-woo-woo-woo (6,7,k(a))

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- wabba-wabba-wabba-woo-woo-woo (0,0,k(w))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(b))
- wabba-wabba-wabba-woo-woo-woo (1,1,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(-))
- wabba-wabba-wabba-woo-woo-woo (6,7,k(a))
- wabba-wabba-wabba-woo-woo-woo

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- wabba-wabba-wabba-woo-woo-woo (0,0,k(w))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(b))
- wabba-wabba-wabba-woo-woo-woo (1,1,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(-))
- wabba-wabba-wabba-woo-woo-woo (6,7,k(a))
- wabba-wabba-wabba-woo-woo-woo (6,5,k(o))

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- wabba-wabba-wabba-woo-woo-woo (0,0,k(w))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(b))
- wabba-wabba-wabba-woo-woo-woo (1,1,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(-))
- wabba-wabba-wabba-woo-woo-woo (6,7,k(a))
- wabba-wabba-wabba-woo-woo-woo (6,5,k(o))
- wabba-wabba-wabba-woo-woo-woo

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- wabba-wabba-wabba-woo-woo-woo (0,0,k(w))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(b))
- wabba-wabba-wabba-woo-woo-woo (1,1,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(-))
- wabba-wabba-wabba-woo-woo-woo (6,7,k(a))
- wabba-wabba-wabba-woo-woo-woo (6,5,k(o))
- wabba-wabba-wabba-woo-woo-woo (1,1,k(-))

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- wabba-wabba-wabba-woo-woo-woo (0,0,k(w))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(b))
- wabba-wabba-wabba-woo-woo-woo (1,1,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(-))
- wabba-wabba-wabba-woo-woo-woo (6,7,k(a))
- wabba-wabba-wabba-woo-woo-woo (6,5,k(o))
- wabba-wabba-wabba-woo-woo-woo (1,1,k(-))
- wabba-wabba-wabba-woo-woo-woo

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- wabba-wabba-wabba-woo-woo-woo (0,0,k(w))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(b))
- wabba-wabba-wabba-woo-woo-woo (1,1,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(-))
- wabba-wabba-wabba-woo-woo-woo (6,7,k(a))
- wabba-wabba-wabba-woo-woo-woo (6,5,k(o))
- wabba-wabba-wabba-woo-woo-woo (1,1,k(-))
- wabba-wabba-wabba-woo-woo-woo (4,6,k(o))

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- $(0,0,k(w))$

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- $(0,0,k(w))$ w

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- $(0,0,k(w))$ **w**
- $(0,0,k(a))$

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- $(0,0,k(w))$ **w**
- $(0,0,k(a))$ **wa**

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- $(0,0,k(w))$ **w**
- $(0,0,k(a))$ **wa**
- $(0,0,k(b))$

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- $(0,0,k(w))$ **w**
- $(0,0,k(a))$ **wa**
- $(0,0,k(b))$ **wab**

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- $(0,0,k(w))$ **w**
- $(0,0,k(a))$ **wa**
- $(0,0,k(b))$ **wab**
- $(1,1,k(a))$

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- $(0,0,k(w))$ **w**
- $(0,0,k(a))$ **wa**
- $(0,0,k(b))$ **wab**
- $(1,1,k(a))$ **wabba**

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- $(0,0,k(w))$ **w**
- $(0,0,k(a))$ **wa**
- $(0,0,k(b))$ **wab**
- $(1,1,k(a))$ **wabba**
- $(0,0,k(-))$

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- $(0,0,k(w))$ **w**
- $(0,0,k(a))$ **wa**
- $(0,0,k(b))$ **wab**
- $(1,1,k(a))$ **wabba**
- $(0,0,k(-))$ **wabba-**

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- $(0,0,k(w))$ **w**
- $(0,0,k(a))$ **wa**
- $(0,0,k(b))$ **wab**
- $(1,1,k(a))$ **wabba**
- $(0,0,k(-))$ **wabba-**
- $(6,7,k(a))$

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- $(0,0,k(w))$ w
- $(0,0,k(a))$ wa
- $(0,0,k(b))$ wab
- $(1,1,k(a))$ wabba
- $(0,0,k(-))$ wabba-
- $(6,7,k(a))$ wabba-wabba-wa

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- $(0,0,k(w))$ w
- $(0,0,k(a))$ wa
- $(0,0,k(b))$ wab
- $(1,1,k(a))$ wabba
- $(0,0,k(-))$ wabba-
- $(6,7,k(a))$ wabba-wabba-wa
- $(6,5,k(o))$

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- $(0,0,k(w))$ w
- $(0,0,k(a))$ wa
- $(0,0,k(b))$ wab
- $(1,1,k(a))$ wabba
- $(0,0,k(-))$ wabba-
- $(6,7,k(a))$ wabba-wabba-wa
- $(6,5,k(o))$ wabba-wabba-wabba-wo

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- $(0,0,k(w))$ **w**
- $(0,0,k(a))$ **wa**
- $(0,0,k(b))$ **wab**
- $(1,1,k(a))$ **wabba**
- $(0,0,k(-))$ **wabba-**
- $(6,7,k(a))$ **wabba-wabba-wa**
- $(6,5,k(o))$ **wabba-wabba-wabba-wo**
- $(1,1,k(-))$

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- $(0,0,k(w))$ w
- $(0,0,k(a))$ wa
- $(0,0,k(b))$ wab
- $(1,1,k(a))$ wabba
- $(0,0,k(-))$ wabba-
- $(6,7,k(a))$ wabba-wabba-wa
- $(6,5,k(o))$ wabba-wabba-wabba-wo
- $(1,1,k(-))$ wabba-wabba-wabba-woo-

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- $(0,0,k(w))$ w
- $(0,0,k(a))$ wa
- $(0,0,k(b))$ wab
- $(1,1,k(a))$ wabba
- $(0,0,k(-))$ wabba-
- $(6,7,k(a))$ wabba-wabba-wa
- $(6,5,k(o))$ wabba-wabba-wabba-wo
- $(1,1,k(-))$ wabba-wabba-wabba-woo-
- $(4,6,k(o))$

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ ($2^k - 1$ i 2^k).
- $(0,0,k(w))$ **w**
- $(0,0,k(a))$ **wa**
- $(0,0,k(b))$ **wab**
- $(1,1,k(a))$ **wabba**
- $(0,0,k(-))$ **wabba-**
- $(6,7,k(a))$ **wabba-wabba-wa**
- $(6,5,k(o))$ **wabba-wabba-wabba-wo**
- $(1,1,k(-))$ **wabba-wabba-wabba-woo-**
- $(4,6,k(o))$ **wabba-wabba-wabba-woo-woo-woo**

LZ77 - podsumowanie

- Mamy bufor słownika i bufor kodowania - razem nazywamy to oknem.
- **Co można usprawnić:** Zamiast 3 liczb wysyłać tylko dwie: kopiowany ciąg (bez kodu ostatniej litery) lub 0 i kod nowej litery.
- Krotki kodowane przy pomocy algorytmu Huffmana (adaptacyjne)
- Zastosowanie: zip, gzip, png, arj, rar, ...

Ziv, Lempel, LZ78

- Powtórzenia nie muszą być na małej odległości - wada LZ77.
- Słownik to zbiór numerowanych słów.
- Zawartość słownika jest tworzona w oparciu o zakodowaną część tekstu.
- Kodowanie to ciąg indeksów ze słownika tworzący kodowany tekst.

LZ78 - Algorytm

- Na początku słownik jest pusty (ewentualnie ma jeden wpis $0 - \epsilon$).
- Szukamy w słowniku najdłuższego prefiksu tekstu i wysyłamy numer tego słowa oraz kod następnej litery. Do słownika dodajemy kolejne nowe słowo (to które wysłaliśmy). Jeśli w słowniku nie ma takiego prefiksu to wysyłamy 0 i kod pierwszej litery.
- Czasami na samym końcu wysyłamy krótszy prefiks i ostatnią literę.

wabba-wabba-wabba-woo-woo-woo

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a bba-wabba-wabba-woo-woo-woo

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a bba-wabba-wabba-woo-woo-woo
- $(0, k(b))$ 3 - b

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a bba-wabba-wabba-woo-woo-woo
- $(0, k(b))$ 3 - b ba-wabba-wabba-woo-woo-woo

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a bba-wabba-wabba-woo-woo-woo
- $(0, k(b))$ 3 - b ba-wabba-wabba-woo-woo-woo
- $(3, k(a))$ 4 - ba

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a bba-wabba-wabba-woo-woo-woo
- $(0, k(b))$ 3 - b ba-wabba-wabba-woo-woo-woo
- $(3, k(a))$ 4 - ba -wabba-wabba-woo-woo-woo

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a bba-wabba-wabba-woo-woo-woo
- $(0, k(b))$ 3 - b ba-wabba-wabba-woo-woo-woo
- $(3, k(a))$ 4 - ba -wabba-wabba-woo-woo-woo
- $(0, k(-))$ 5 - -

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a bba-wabba-wabba-woo-woo-woo
- $(0, k(b))$ 3 - b ba-wabba-wabba-woo-woo-woo
- $(3, k(a))$ 4 - ba -wabba-wabba-woo-woo-woo
- $(0, k(-))$ 5 - - wabba-wabba-woo-woo-woo

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a bba-wabba-wabba-woo-woo-woo
- $(0, k(b))$ 3 - b ba-wabba-wabba-woo-woo-woo
- $(3, k(a))$ 4 - ba -wabba-wabba-woo-woo-woo
- $(0, k(-))$ 5 - - wabba-wabba-woo-woo-woo
- $(1, k(a))$ 6 - wa

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a bba-wabba-wabba-woo-woo-woo
- $(0, k(b))$ 3 - b ba-wabba-wabba-woo-woo-woo
- $(3, k(a))$ 4 - ba -wabba-wabba-woo-woo-woo
- $(0, k(-))$ 5 - - wabba-wabba-woo-woo-woo
- $(1, k(a))$ 6 - wa bba-wabba-woo-woo-woo

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a bba-wabba-wabba-woo-woo-woo
- $(0, k(b))$ 3 - b ba-wabba-wabba-woo-woo-woo
- $(3, k(a))$ 4 - ba -wabba-wabba-woo-woo-woo
- $(0, k(-))$ 5 - - wabba-wabba-woo-woo-woo
- $(1, k(a))$ 6 - wa bba-wabba-woo-woo-woo
- $(3, k(b))$ 7 - bb

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a bba-wabba-wabba-woo-woo-woo
- $(0, k(b))$ 3 - b ba-wabba-wabba-woo-woo-woo
- $(3, k(a))$ 4 - ba -wabba-wabba-woo-woo-woo
- $(0, k(-))$ 5 - - wabba-wabba-woo-woo-woo
- $(1, k(a))$ 6 - wa bba-wabba-woo-woo-woo
- $(3, k(b))$ 7 - bb a-wabba-woo-woo-woo

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a bba-wabba-wabba-woo-woo-woo
- $(0, k(b))$ 3 - b ba-wabba-wabba-woo-woo-woo
- $(3, k(a))$ 4 - ba -wabba-wabba-woo-woo-woo
- $(0, k(-))$ 5 - - wabba-wabba-woo-woo-woo
- $(1, k(a))$ 6 - wa bba-wabba-woo-woo-woo
- $(3, k(b))$ 7 - bb a-wabba-woo-woo-woo
- $(2, k(-))$ 8 - a-

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a bba-wabba-wabba-woo-woo-woo
- $(0, k(b))$ 3 - b ba-wabba-wabba-woo-woo-woo
- $(3, k(a))$ 4 - ba -wabba-wabba-woo-woo-woo
- $(0, k(-))$ 5 - - wabba-wabba-woo-woo-woo
- $(1, k(a))$ 6 - wa bba-wabba-woo-woo-woo
- $(3, k(b))$ 7 - bb a-wabba-woo-woo-woo
- $(2, k(-))$ 8 - a- wabba-woo-woo-woo

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a bba-wabba-wabba-woo-woo-woo
- $(0, k(b))$ 3 - b ba-wabba-wabba-woo-woo-woo
- $(3, k(a))$ 4 - ba -wabba-wabba-woo-woo-woo
- $(0, k(-))$ 5 - - wabba-wabba-woo-woo-woo
- $(1, k(a))$ 6 - wa bba-wabba-woo-woo-woo
- $(3, k(b))$ 7 - bb a-wabba-woo-woo-woo
- $(2, k(-))$ 8 - a- wabba-woo-woo-woo
- $(6, k(b))$ 9 - wab

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a bba-wabba-wabba-woo-woo-woo
- $(0, k(b))$ 3 - b ba-wabba-wabba-woo-woo-woo
- $(3, k(a))$ 4 - ba -wabba-wabba-woo-woo-woo
- $(0, k(-))$ 5 - - wabba-wabba-woo-woo-woo
- $(1, k(a))$ 6 - wa bba-wabba-woo-woo-woo
- $(3, k(b))$ 7 - bb a-wabba-woo-woo-woo
- $(2, k(-))$ 8 - a- wabba-woo-woo-woo
- $(6, k(b))$ 9 - wab ba-woo-woo-woo

wabba-wabba-wabba-woo-woo-woo

- (0,k(w)) 1 - w abba-wabba-wabba-woo-woo-woo
- (0,k(a)) 2 - a bba-wabba-wabba-woo-woo-woo
- (0,k(b)) 3 - b ba-wabba-wabba-woo-woo-woo
- (3,k(a)) 4 - ba -wabba-wabba-woo-woo-woo
- (0,k(-)) 5 - - wabba-wabba-woo-woo-woo
- (1,k(a)) 6 - wa bba-wabba-woo-woo-woo
- (3,k(b)) 7 - bb a-wabba-woo-woo-woo
- (2,k(-)) 8 - a- wabba-woo-woo-woo
- (6,k(b)) 9 - wab ba-woo-woo-woo
- (4,k(-)) 10 - ba-

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a bba-wabba-wabba-woo-woo-woo
- $(0, k(b))$ 3 - b ba-wabba-wabba-woo-woo-woo
- $(3, k(a))$ 4 - ba -wabba-wabba-woo-woo-woo
- $(0, k(-))$ 5 - - wabba-wabba-woo-woo-woo
- $(1, k(a))$ 6 - wa bba-wabba-woo-woo-woo
- $(3, k(b))$ 7 - bb a-wabba-woo-woo-woo
- $(2, k(-))$ 8 - a- wabba-woo-woo-woo
- $(6, k(b))$ 9 - wab ba-woo-woo-woo
- $(4, k(-))$ 10 - ba- woo-woo-woo

wabba-wabba-wabba-woo-woo-woo

- (0,k(w)) 1 - w abba-wabba-wabba-woo-woo-woo
- (0,k(a)) 2 - a bba-wabba-wabba-woo-woo-woo
- (0,k(b)) 3 - b ba-wabba-wabba-woo-woo-woo
- (3,k(a)) 4 - ba -wabba-wabba-woo-woo-woo
- (0,k(-)) 5 - - wabba-wabba-woo-woo-woo
- (1,k(a)) 6 - wa bba-wabba-woo-woo-woo
- (3,k(b)) 7 - bb a-wabba-woo-woo-woo
- (2,k(-)) 8 - a- wabba-woo-woo-woo
- (6,k(b)) 9 - wab ba-woo-woo-woo
- (4,k(-)) 10 - ba- woo-woo-woo
- (1,k(o)) 11 - wo

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a bba-wabba-wabba-woo-woo-woo
- $(0, k(b))$ 3 - b ba-wabba-wabba-woo-woo-woo
- $(3, k(a))$ 4 - ba -wabba-wabba-woo-woo-woo
- $(0, k(-))$ 5 - - wabba-wabba-woo-woo-woo
- $(1, k(a))$ 6 - wa bba-wabba-woo-woo-woo
- $(3, k(b))$ 7 - bb a-wabba-woo-woo-woo
- $(2, k(-))$ 8 - a- wabba-woo-woo-woo
- $(6, k(b))$ 9 - wab ba-woo-woo-woo
- $(4, k(-))$ 10 - ba- woo-woo-woo
- $(1, k(o))$ 11 - wo o-woo-woo

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a bba-wabba-wabba-woo-woo-woo
- $(0, k(b))$ 3 - b ba-wabba-wabba-woo-woo-woo
- $(3, k(a))$ 4 - ba -wabba-wabba-woo-woo-woo
- $(0, k(-))$ 5 - - wabba-wabba-woo-woo-woo
- $(1, k(a))$ 6 - wa bba-wabba-woo-woo-woo
- $(3, k(b))$ 7 - bb a-wabba-woo-woo-woo
- $(2, k(-))$ 8 - a- wabba-woo-woo-woo
- $(6, k(b))$ 9 - wab ba-woo-woo-woo
- $(4, k(-))$ 10 - ba- woo-woo-woo
- $(1, k(o))$ 11 - wo o-woo-woo
- $(0, k(o))$ 12 - o

wabba-wabba-wabba-woo-woo-woo

- (0,k(w)) 1 - w abba-wabba-wabba-woo-woo-woo
- (0,k(a)) 2 - a bba-wabba-wabba-woo-woo-woo
- (0,k(b)) 3 - b ba-wabba-wabba-woo-woo-woo
- (3,k(a)) 4 - ba -wabba-wabba-woo-woo-woo
- (0,k(-)) 5 - - wabba-wabba-woo-woo-woo
- (1,k(a)) 6 - wa bba-wabba-woo-woo-woo
- (3,k(b)) 7 - bb a-wabba-woo-woo-woo
- (2,k(-)) 8 - a- wabba-woo-woo-woo
- (6,k(b)) 9 - wab ba-woo-woo-woo
- (4,k(-)) 10 - ba- woo-woo-woo
- (1,k(o)) 11 - wo o-woo-woo
- (0,k(o)) 12 - o -woo-woo

wabba-wabba-wabba-woo-woo-woo

- $(0, k(w))$ 1 - w abba-wabba-wabba-woo-woo-woo
- $(0, k(a))$ 2 - a bba-wabba-wabba-woo-woo-woo
- $(0, k(b))$ 3 - b ba-wabba-wabba-woo-woo-woo
- $(3, k(a))$ 4 - ba -wabba-wabba-woo-woo-woo
- $(0, k(-))$ 5 - - wabba-wabba-woo-woo-woo
- $(1, k(a))$ 6 - wa bba-wabba-woo-woo-woo
- $(3, k(b))$ 7 - bb a-wabba-woo-woo-woo
- $(2, k(-))$ 8 - a- wabba-woo-woo-woo
- $(6, k(b))$ 9 - wab ba-woo-woo-woo
- $(4, k(-))$ 10 - ba- woo-woo-woo
- $(1, k(o))$ 11 - wo o-woo-woo
- $(0, k(o))$ 12 - o -woo-woo
- $(5, k(w))$ 13 - -w

wabba-wabba-wabba-woo-woo-woo

- (0,k(w)) 1 - w abba-wabba-wabba-woo-woo-woo
- (0,k(a)) 2 - a bba-wabba-wabba-woo-woo-woo
- (0,k(b)) 3 - b ba-wabba-wabba-woo-woo-woo
- (3,k(a)) 4 - ba -wabba-wabba-woo-woo-woo
- (0,k(-)) 5 - - wabba-wabba-woo-woo-woo
- (1,k(a)) 6 - wa bba-wabba-woo-woo-woo
- (3,k(b)) 7 - bb a-wabba-woo-woo-woo
- (2,k(-)) 8 - a- wabba-woo-woo-woo
- (6,k(b)) 9 - wab ba-woo-woo-woo
- (4,k(-)) 10 - ba- woo-woo-woo
- (1,k(o)) 11 - wo o-woo-woo
- (0,k(o)) 12 - o -woo-woo
- (5,k(w)) 13 - -w oo-woo

wabba-wabba-wabba-woo-woo-woo

- (0,k(w)) 1 - w abba-wabba-wabba-woo-woo-woo
- (0,k(a)) 2 - a bba-wabba-wabba-woo-woo-woo
- (0,k(b)) 3 - b ba-wabba-wabba-woo-woo-woo
- (3,k(a)) 4 - ba -wabba-wabba-woo-woo-woo
- (0,k(-)) 5 - - wabba-wabba-woo-woo-woo
- (1,k(a)) 6 - wa bba-wabba-woo-woo-woo
- (3,k(b)) 7 - bb a-wabba-woo-woo-woo
- (2,k(-)) 8 - a- wabba-woo-woo-woo
- (6,k(b)) 9 - wab ba-woo-woo-woo
- (4,k(-)) 10 - ba- woo-woo-woo
- (1,k(o)) 11 - wo o-woo-woo
- (0,k(o)) 12 - o -woo-woo
- (5,k(w)) 13 - -w oo-woo
- (12,k(o)) 14 - oo

wabba-wabba-wabba-woo-woo-woo

- (0,k(w)) 1 - w abba-wabba-wabba-woo-woo-woo
- (0,k(a)) 2 - a bba-wabba-wabba-woo-woo-woo
- (0,k(b)) 3 - b ba-wabba-wabba-woo-woo-woo
- (3,k(a)) 4 - ba -wabba-wabba-woo-woo-woo
- (0,k(-)) 5 - - wabba-wabba-woo-woo-woo
- (1,k(a)) 6 - wa bba-wabba-woo-woo-woo
- (3,k(b)) 7 - bb a-wabba-woo-woo-woo
- (2,k(-)) 8 - a- wabba-woo-woo-woo
- (6,k(b)) 9 - wab ba-woo-woo-woo
- (4,k(-)) 10 - ba- woo-woo-woo
- (1,k(o)) 11 - wo o-woo-woo
- (0,k(o)) 12 - o -woo-woo
- (5,k(w)) 13 - -w oo-woo
- (12,k(o)) 14 - oo -woo

wabba-wabba-wabba-woo-woo-woo

- (0,k(w)) 1 - w abba-wabba-wabba-woo-woo-woo
- (0,k(a)) 2 - a bba-wabba-wabba-woo-woo-woo
- (0,k(b)) 3 - b ba-wabba-wabba-woo-woo-woo
- (3,k(a)) 4 - ba -wabba-wabba-woo-woo-woo
- (0,k(-)) 5 - - wabba-wabba-woo-woo-woo
- (1,k(a)) 6 - wa bba-wabba-woo-woo-woo
- (3,k(b)) 7 - bb a-wabba-woo-woo-woo
- (2,k(-)) 8 - a- wabba-woo-woo-woo
- (6,k(b)) 9 - wab ba-woo-woo-woo
- (4,k(-)) 10 - ba- woo-woo-woo
- (1,k(o)) 11 - wo o-woo-woo
- (0,k(o)) 12 - o -woo-woo
- (5,k(w)) 13 - -w oo-woo
- (12,k(o)) 14 - oo -woo
- (13,k(o)) 15 - -wo

wabba-wabba-wabba-woo-woo-woo

- (0,k(w)) 1 - w abba-wabba-wabba-woo-woo-woo
- (0,k(a)) 2 - a bba-wabba-wabba-woo-woo-woo
- (0,k(b)) 3 - b ba-wabba-wabba-woo-woo-woo
- (3,k(a)) 4 - ba -wabba-wabba-woo-woo-woo
- (0,k(-)) 5 - - wabba-wabba-woo-woo-woo
- (1,k(a)) 6 - wa bba-wabba-woo-woo-woo
- (3,k(b)) 7 - bb a-wabba-woo-woo-woo
- (2,k(-)) 8 - a- wabba-woo-woo-woo
- (6,k(b)) 9 - wab ba-woo-woo-woo
- (4,k(-)) 10 - ba- woo-woo-woo
- (1,k(o)) 11 - wo o-woo-woo
- (0,k(o)) 12 - o -woo-woo
- (5,k(w)) 13 - -w oo-woo
- (12,k(o)) 14 - oo -woo
- (13,k(o)) 15 - -wo o

wabba-wabba-wabba-woo-woo-woo

- (0,k(w)) 1 - w abba-wabba-wabba-woo-woo-woo
- (0,k(a)) 2 - a bba-wabba-wabba-woo-woo-woo
- (0,k(b)) 3 - b ba-wabba-wabba-woo-woo-woo
- (3,k(a)) 4 - ba -wabba-wabba-woo-woo-woo
- (0,k(-)) 5 - - wabba-wabba-woo-woo-woo
- (1,k(a)) 6 - wa bba-wabba-woo-woo-woo
- (3,k(b)) 7 - bb a-wabba-woo-woo-woo
- (2,k(-)) 8 - a- wabba-woo-woo-woo
- (6,k(b)) 9 - wab ba-woo-woo-woo
- (4,k(-)) 10 - ba- woo-woo-woo
- (1,k(o)) 11 - wo o-woo-woo
- (0,k(o)) 12 - o -woo-woo
- (5,k(w)) 13 - -w oo-woo
- (12,k(o)) 14 - oo -woo
- (13,k(o)) 15 - -wo o
- (0,k(o))

Odkodowywanie

- $(0, k(w))$

Odkodowywanie

- $(0, k(w))$ 1 - w w

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$ 6 - wa wabba-wa

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$ 6 - wa wabba-wa
- $(3, k(b))$

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$ 6 - wa wabba-wa
- $(3, k(b))$ 7 - bb wabba-wabb

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$ 6 - wa wabba-wa
- $(3, k(b))$ 7 - bb wabba-wabb
- $(2, k(-))$

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$ 6 - wa wabba-wa
- $(3, k(b))$ 7 - bb wabba-wabb
- $(2, k(-))$ 8 - a- wabba-wabba-

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$ 6 - wa wabba-wa
- $(3, k(b))$ 7 - bb wabba-wabb
- $(2, k(-))$ 8 - a- wabba-wabba-
- $(6, k(b))$

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$ 6 - wa wabba-wa
- $(3, k(b))$ 7 - bb wabba-wabb
- $(2, k(-))$ 8 - a- wabba-wabba-
- $(6, k(b))$ 9 - wab wabba-wabba-wab

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$ 6 - wa wabba-wa
- $(3, k(b))$ 7 - bb wabba-wabb
- $(2, k(-))$ 8 - a- wabba-wabba-
- $(6, k(b))$ 9 - wab wabba-wabba-wab
- $(4, k(-))$

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$ 6 - wa wabba-wa
- $(3, k(b))$ 7 - bb wabba-wabb
- $(2, k(-))$ 8 - a- wabba-wabba-
- $(6, k(b))$ 9 - wab wabba-wabba-wab
- $(4, k(-))$ 10 - ba- wabba-wabba-wabba-

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$ 6 - wa wabba-wa
- $(3, k(b))$ 7 - bb wabba-wabb
- $(2, k(-))$ 8 - a- wabba-wabba-
- $(6, k(b))$ 9 - wab wabba-wabba-wab
- $(4, k(-))$ 10 - ba- wabba-wabba-wabba-
- $(1, k(o))$

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$ 6 - wa wabba-wa
- $(3, k(b))$ 7 - bb wabba-wabb
- $(2, k(-))$ 8 - a- wabba-wabba-
- $(6, k(b))$ 9 - wab wabba-wabba-wab
- $(4, k(-))$ 10 - ba- wabba-wabba-wabba-
- $(1, k(o))$ 11 - wo wabba-wabba-wabba-wo

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$ 6 - wa wabba-wa
- $(3, k(b))$ 7 - bb wabba-wabb
- $(2, k(-))$ 8 - a- wabba-wabba-
- $(6, k(b))$ 9 - wab wabba-wabba-wab
- $(4, k(-))$ 10 - ba- wabba-wabba-wabba-
- $(1, k(o))$ 11 - wo wabba-wabba-wabba-wo
- $(0, k(o))$

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$ 6 - wa wabba-wa
- $(3, k(b))$ 7 - bb wabba-wabb
- $(2, k(-))$ 8 - a- wabba-wabba-
- $(6, k(b))$ 9 - wab wabba-wabba-wab
- $(4, k(-))$ 10 - ba- wabba-wabba-wabba-
- $(1, k(o))$ 11 - wo wabba-wabba-wabba-wo
- $(0, k(o))$ 12 - o wabba-wabba-wabba-woo

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$ 6 - wa wabba-wa
- $(3, k(b))$ 7 - bb wabba-wabb
- $(2, k(-))$ 8 - a- wabba-wabba-
- $(6, k(b))$ 9 - wab wabba-wabba-wab
- $(4, k(-))$ 10 - ba- wabba-wabba-wabba-
- $(1, k(o))$ 11 - wo wabba-wabba-wabba-wo
- $(0, k(o))$ 12 - o wabba-wabba-wabba-woo
- $(5, k(w))$

Odkodowywanie

- (0,k(w)) 1 - w w
- (0,k(a)) 2 - a wa
- (0,k(b)) 3 - b wab
- (3,k(a)) 4 - ba wabba
- (0,k(-)) 5 - - wabba-
- (1,k(a)) 6 - wa wabba-wa
- (3,k(b)) 7 - bb wabba-wabb
- (2,k(-)) 8 - a- wabba-wabba-
- (6,k(b)) 9 - wab wabba-wabba-wab
- (4,k(-)) 10 - ba- wabba-wabba-wabba-
- (1,k(o)) 11 - wo wabba-wabba-wabba-wo
- (0,k(o)) 12 - o wabba-wabba-wabba-woo
- (5,k(w)) 13 - -w wabba-wabba-wabba-woo-w

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$ 6 - wa wabba-wa
- $(3, k(b))$ 7 - bb wabba-wabb
- $(2, k(-))$ 8 - a- wabba-wabba-
- $(6, k(b))$ 9 - wab wabba-wabba-wab
- $(4, k(-))$ 10 - ba- wabba-wabba-wabba-
- $(1, k(o))$ 11 - wo wabba-wabba-wabba-wo
- $(0, k(o))$ 12 - o wabba-wabba-wabba-woo
- $(5, k(w))$ 13 - -w wabba-wabba-wabba-woo-w
- $(12, k(o))$

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$ 6 - wa wabba-wa
- $(3, k(b))$ 7 - bb wabba-wabb
- $(2, k(-))$ 8 - a- wabba-wabba-
- $(6, k(b))$ 9 - wab wabba-wabba-wab
- $(4, k(-))$ 10 - ba- wabba-wabba-wabba-
- $(1, k(o))$ 11 - wo wabba-wabba-wabba-wo
- $(0, k(o))$ 12 - o wabba-wabba-wabba-woo
- $(5, k(w))$ 13 - -w wabba-wabba-wabba-woo-w
- $(12, k(o))$ 14 - oo wabba-wabba-wabba-woo-woo

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$ 6 - wa wabba-wa
- $(3, k(b))$ 7 - bb wabba-wabb
- $(2, k(-))$ 8 - a- wabba-wabba-
- $(6, k(b))$ 9 - wab wabba-wabba-wab
- $(4, k(-))$ 10 - ba- wabba-wabba-wabba-
- $(1, k(o))$ 11 - wo wabba-wabba-wabba-wo
- $(0, k(o))$ 12 - o wabba-wabba-wabba-woo
- $(5, k(w))$ 13 - -w wabba-wabba-wabba-woo-w
- $(12, k(o))$ 14 - oo wabba-wabba-wabba-woo-woo
- $(13, k(o))$

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$ 6 - wa wabba-wa
- $(3, k(b))$ 7 - bb wabba-wabb
- $(2, k(-))$ 8 - a- wabba-wabba-
- $(6, k(b))$ 9 - wab wabba-wabba-wab
- $(4, k(-))$ 10 - ba- wabba-wabba-wabba-
- $(1, k(o))$ 11 - wo wabba-wabba-wabba-wo
- $(0, k(o))$ 12 - o wabba-wabba-wabba-woo
- $(5, k(w))$ 13 - -w wabba-wabba-wabba-woo-w
- $(12, k(o))$ 14 - oo wabba-wabba-wabba-woo-woo
- $(13, k(o))$ 15 - -wo wabba-wabba-wabba-woo-woo-wo

Odkodowywanie

- $(0, k(w))$ 1 - w w
- $(0, k(a))$ 2 - a wa
- $(0, k(b))$ 3 - b wab
- $(3, k(a))$ 4 - ba wabba
- $(0, k(-))$ 5 - - wabba-
- $(1, k(a))$ 6 - wa wabba-wa
- $(3, k(b))$ 7 - bb wabba-wabb
- $(2, k(-))$ 8 - a- wabba-wabba-
- $(6, k(b))$ 9 - wab wabba-wabba-wab
- $(4, k(-))$ 10 - ba- wabba-wabba-wabba-
- $(1, k(o))$ 11 - wo wabba-wabba-wabba-wo
- $(0, k(o))$ 12 - o wabba-wabba-wabba-woo
- $(5, k(w))$ 13 - -w wabba-wabba-wabba-woo-w
- $(12, k(o))$ 14 - oo wabba-wabba-wabba-woo-woo
- $(13, k(o))$ 15 - -wo wabba-wabba-wabba-woo-woo-wo
- $(0, k(o))$ wabba-wabba-wabba-woo-woo-woo

Ziv, Lempel, Welch - LZW

- Poprawki do LZ78 - rezygnacja z drugiego elementu pary.
- Potrzebny słownik początkowy zawierający wszystkie używane litery.
- Słownik konstruujemy tak samo jak w LZ78.

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- wabba-wabba-wabba-woo-woo-woo

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - .
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - .
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo
- 1 8 - ba

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - .
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo
- 1 8 - ba a-wabba-wabba-woo-woo-woo

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo
- 1 8 - ba a-wabba-wabba-woo-woo-woo
- 0 9 - a-

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo
- 1 8 - ba a-wabba-wabba-woo-woo-woo
- 0 9 - a- -wabba-wabba-woo-woo-woo

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo
- 1 8 - ba a-wabba-wabba-woo-woo-woo
- 0 9 - a- -wabba-wabba-woo-woo-woo
- 4 10 - -w

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo
- 1 8 - ba a-wabba-wabba-woo-woo-woo
- 0 9 - a- -wabba-wabba-woo-woo-woo
- 4 10 - -w wabba-wabba-woo-woo-woo

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - .
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo
- 1 8 - ba a-wabba-wabba-woo-woo-woo
- 0 9 - a- -wabba-wabba-woo-woo-woo
- 4 10 - -w wabba-wabba-woo-woo-woo
- 5 11 - wab

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - .
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo
- 1 8 - ba a-wabba-wabba-woo-woo-woo
- 0 9 - a- -wabba-wabba-woo-woo-woo
- 4 10 - -w wabba-wabba-woo-woo-woo
- 5 11 - wab bba-wabba-woo-woo-woo

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - .
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo
- 1 8 - ba a-wabba-wabba-woo-woo-woo
- 0 9 - a- -wabba-wabba-woo-woo-woo
- 4 10 - -w wabba-wabba-woo-woo-woo
- 5 11 - wab bba-wabba-woo-woo-woo
- 7 12 - bba

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - .
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo
- 1 8 - ba a-wabba-wabba-woo-woo-woo
- 0 9 - a- -wabba-wabba-woo-woo-woo
- 4 10 - -w wabba-wabba-woo-woo-woo
- 5 11 - wab bba-wabba-woo-woo-woo
- 7 12 - bba a-wabba-woo-woo-woo

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo
- 1 8 - ba a-wabba-wabba-woo-woo-woo
- 0 9 - a- -wabba-wabba-woo-woo-woo
- 4 10 - -w wabba-wabba-woo-woo-woo
- 5 11 - wab bba-wabba-woo-woo-woo
- 7 12 - bba a-wabba-woo-woo-woo
- 9 13 - a-w

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo
- 1 8 - ba a-wabba-wabba-woo-woo-woo
- 0 9 - a- -wabba-wabba-woo-woo-woo
- 4 10 - -w wabba-wabba-woo-woo-woo
- 5 11 - wab bba-wabba-woo-woo-woo
- 7 12 - bba a-wabba-woo-woo-woo
- 9 13 - a-w wabba-woo-woo-woo

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo
- 1 8 - ba a-wabba-wabba-woo-woo-woo
- 0 9 - a- -wabba-wabba-woo-woo-woo
- 4 10 - -w wabba-wabba-woo-woo-woo
- 5 11 - wab bba-wabba-woo-woo-woo
- 7 12 - bba a-wabba-woo-woo-woo
- 9 13 - a-w wabba-woo-woo-woo
- 11 14 - wabb

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo
- 1 8 - ba a-wabba-wabba-woo-woo-woo
- 0 9 - a- -wabba-wabba-woo-woo-woo
- 4 10 - -w wabba-wabba-woo-woo-woo
- 5 11 - wab bba-wabba-woo-woo-woo
- 7 12 - bba a-wabba-woo-woo-woo
- 9 13 - a-w wabba-woo-woo-woo
- 11 14 - wabb ba-woo-woo-woo

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo
- 1 8 - ba a-wabba-wabba-woo-woo-woo
- 0 9 - a- -wabba-wabba-woo-woo-woo
- 4 10 - -w wabba-wabba-woo-woo-woo
- 5 11 - wab bba-wabba-woo-woo-woo
- 7 12 - bba a-wabba-woo-woo-woo
- 9 13 - a-w wabba-woo-woo-woo
- 11 14 - wabb ba-woo-woo-woo
- 8 15 - ba-

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo
- 1 8 - ba a-wabba-wabba-woo-woo-woo
- 0 9 - a- -wabba-wabba-woo-woo-woo
- 4 10 - -w wabba-wabba-woo-woo-woo
- 5 11 - wab bba-wabba-woo-woo-woo
- 7 12 - bba a-wabba-woo-woo-woo
- 9 13 - a-w wabba-woo-woo-woo
- 11 14 - wabb ba-woo-woo-woo
- 8 15 - ba- -woo-woo-woo

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo
- 1 8 - ba a-wabba-wabba-woo-woo-woo
- 0 9 - a- -wabba-wabba-woo-woo-woo
- 4 10 - -w wabba-wabba-woo-woo-woo
- 5 11 - wab bba-wabba-woo-woo-woo
- 7 12 - bba a-wabba-woo-woo-woo
- 9 13 - a-w wabba-woo-woo-woo
- 11 14 - wabb ba-woo-woo-woo
- 8 15 - ba- -woo-woo-woo
- 10 16 - -wo

Kodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- wabba-wabba-wabba-woo-woo-woo
- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo
- 1 8 - ba a-wabba-wabba-woo-woo-woo
- 0 9 - a- -wabba-wabba-woo-woo-woo
- 4 10 - -w wabba-wabba-woo-woo-woo
- 5 11 - wab bba-wabba-woo-woo-woo
- 7 12 - bba a-wabba-woo-woo-woo
- 9 13 - a-w wabba-woo-woo-woo
- 11 14 - wabb ba-woo-woo-woo
- 8 15 - ba- -woo-woo-woo
- 10 16 - -wo oo-woo-woo

- 2 17 - oo

● 2 17 - oo 0-WOO-WOO

Kodowanie

- 2 17 - oo 0-WOO-WOO
- 2 18 - o-

Kodowanie

- 2 17 - oo O-WOO-WOO
- 2 18 - o- -WOO-WOO

Kodowanie

- 2 17 - oo O-WOO-WOO
- 2 18 - o- -WOO-WOO
- 10 19 - -wo

Kodowanie

- 2 17 - oo O-WOO-WOO
- 2 18 - o- -WOO-WOO
- 10 19 - -wo OO-WOO

Kodowanie

- 2 17 - oo O-WOO-WOO
- 2 18 - o- -WOO-WOO
- 10 19 - -wo OO-WOO
- 17 20 - oo-

Kodowanie

- 2 17 - oo 0-WOO-WOO
- 2 18 - o- -WOO-WOO
- 10 19 - -wo OO-WOO
- 17 20 - oo- -WOO

Kodowanie

- 2 17 - oo 0-WOO-WOO
- 2 18 - o- -WOO-WOO
- 10 19 - -wo oo-WOO
- 17 20 - oo- -WOO
- 19 21 - -woo

Kodowanie

- 2 17 - oo 0-WOO-WOO
- 2 18 - o- -WOO-WOO
- 10 19 - -wo oo-WOO
- 17 20 - oo- -WOO
- 19 21 - -woo o

Kodowanie

- 2 17 - oo 0-WOO-WOO
- 2 18 - o- -WOO-WOO
- 10 19 - -wo oo-WOO
- 17 20 - oo- -WOO
- 19 21 - -woo o
- 2

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1 wabb 7 - bb, 8 - b?

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1 wabb 7 - bb, 8 - b?
- 0

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1 wabb 7 - bb, 8 - b?
- 0 wabba 8 - ba, 9 - a?

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1 wabb 7 - bb, 8 - b?
- 0 wabba 8 - ba, 9 - a?
- 4

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1 wabb 7 - bb, 8 - b?
- 0 wabba 8 - ba, 9 - a?
- 4 wabba- 9 - a-, 10 - -?

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1 wabb 7 - bb, 8 - b?
- 0 wabba 8 - ba, 9 - a?
- 4 wabba- 9 - a-, 10 - -?
- 5

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1 wabb 7 - bb, 8 - b?
- 0 wabba 8 - ba, 9 - a?
- 4 wabba- 9 - a-, 10 - -?
- 5 wabba-wa 10 - -w, 11 - wa?

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1 wabb 7 - bb, 8 - b?
- 0 wabba 8 - ba, 9 - a?
- 4 wabba- 9 - a-, 10 - -?
- 5 wabba-wa 10 - -w, 11 - wa?
- 7

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1 wabb 7 - bb, 8 - b?
- 0 wabba 8 - ba, 9 - a?
- 4 wabba- 9 - a-, 10 - -?
- 5 wabba-wa 10 - -w, 11 - wa?
- 7 wabba-wabb 11 - wab, 12 - bb?

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1 wabb 7 - bb, 8 - b?
- 0 wabba 8 - ba, 9 - a?
- 4 wabba- 9 - a-, 10 - -?
- 5 wabba-wa 10 - -w, 11 - wa?
- 7 wabba-wabb 11 - wab, 12 - bb?
- 9

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1 wabb 7 - bb, 8 - b?
- 0 wabba 8 - ba, 9 - a?
- 4 wabba- 9 - a-, 10 - -?
- 5 wabba-wa 10 - -w, 11 - wa?
- 7 wabba-wabb 11 - wab, 12 - bb?
- 9 wabba-wabba- 12 - bba, 13 - a?

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1 wabb 7 - bb, 8 - b?
- 0 wabba 8 - ba, 9 - a?
- 4 wabba- 9 - a-, 10 - -?
- 5 wabba-wa 10 - -w, 11 - wa?
- 7 wabba-wabb 11 - wab, 12 - bb?
- 9 wabba-wabba- 12 - bba, 13 - a?
- 11

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1 wabb 7 - bb, 8 - b?
- 0 wabba 8 - ba, 9 - a?
- 4 wabba- 9 - a-, 10 - -?
- 5 wabba-wa 10 - -w, 11 - wa?
- 7 wabba-wabb 11 - wab, 12 - bb?
- 9 wabba-wabba- 12 - bba, 13 - a-?
- 11 wabba-wabba-wab 13 - a-w, 14 - wab?

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1 wabb 7 - bb, 8 - b?
- 0 wabba 8 - ba, 9 - a?
- 4 wabba- 9 - a-, 10 - -?
- 5 wabba-wa 10 - -w, 11 - wa?
- 7 wabba-wabb 11 - wab, 12 - bb?
- 9 wabba-wabba- 12 - bba, 13 - a-?
- 11 wabba-wabba-wab 13 - a-w, 14 - wab?
- 8

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1 wabb 7 - bb, 8 - b?
- 0 wabba 8 - ba, 9 - a?
- 4 wabba- 9 - a-, 10 - -?
- 5 wabba-wa 10 - -w, 11 - wa?
- 7 wabba-wabb 11 - wab, 12 - bb?
- 9 wabba-wabba- 12 - bba, 13 - a?
- 11 wabba-wabba-wab 13 - a-w, 14 - wab?
- 8 wabba-wabba-wabba 14 - wabb, 15 - ba?

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1 wabb 7 - bb, 8 - b?
- 0 wabba 8 - ba, 9 - a?
- 4 wabba- 9 - a-, 10 - -?
- 5 wabba-wa 10 - -w, 11 - wa?
- 7 wabba-wabb 11 - wab, 12 - bb?
- 9 wabba-wabba- 12 - bba, 13 - a?
- 11 wabba-wabba-wab 13 - a-w, 14 - wab?
- 8 wabba-wabba-wabba 14 - wabb, 15 - ba?
- 10

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1 wabb 7 - bb, 8 - b?
- 0 wabba 8 - ba, 9 - a?
- 4 wabba- 9 - a-, 10 - -?
- 5 wabba-wa 10 - -w, 11 - wa?
- 7 wabba-wabb 11 - wab, 12 - bb?
- 9 wabba-wabba- 12 - bba, 13 - a?
- 11 wabba-wabba-wab 13 - a-w, 14 - wab?
- 8 wabba-wabba-wabba 14 - wabb, 15 - ba?
- 10 wabba-wabba-wabba-w 15 - ba-, 16 - -w?

Dekodowanie

- Słownik początkowy: 0 - a, 1 - b, 2 - o, 3 - w, 4 - -.
- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1 wabb 7 - bb, 8 - b?
- 0 wabba 8 - ba, 9 - a?
- 4 wabba- 9 - a-, 10 - -?
- 5 wabba-wa 10 - -w, 11 - wa?
- 7 wabba-wabb 11 - wab, 12 - bb?
- 9 wabba-wabba- 12 - bba, 13 - a?
- 11 wabba-wabba-wab 13 - a-w, 14 - wab?
- 8 wabba-wabba-wabba 14 - wabb, 15 - ba?
- 10 wabba-wabba-wabba-w 15 - ba-, 16 - -w?
- ...

Trudny przypadek

- Słowo abababababab... i słownik 1 - a i 2 - b.

Trudny przypadek

- Słowo abababababab... i słownik 1 - a i 2 - b.
- 1 3 - ab, 2 4 - ba, 3 5 - aba, 5 6 - abab, ...

Trudny przypadek

- Słowo abababababab... i słownik 1 - a i 2 - b.
- 1 3 - ab, 2 4 - ba, 3 5 - aba, 5 6 - abab, ...
- Dekodowanie:

Trudny przypadek

- Słowo abababababab... i słownik 1 - a i 2 - b.
- 1 3 - ab, 2 4 - ba, 3 5 - aba, 5 6 - abab, ...
- Dekodowanie:
 - 1 a 3 - a?

Trudny przypadek

- Słowo abababababab... i słownik 1 - a i 2 - b.
- 1 3 - ab, 2 4 - ba, 3 5 - aba, 5 6 - abab, ...
- Dekodowanie:
 - 1 a 3 - a?
 - 2 ab 3 - ab, 4 - b?

Trudny przypadek

- Słowo abababababab... i słownik 1 - a i 2 - b.
- 1 3 - ab, 2 4 - ba, 3 5 - aba, 5 6 - abab, ...
- Dekodowanie:
 - 1 a 3 - a?
 - 2 ab 3 - ab, 4 - b?
 - 3 abab 4 - ba, 5 - ab?

Trudny przypadek

- Słowo abababababab... i słownik 1 - a i 2 - b.
- 1 3 - ab, 2 4 - ba, 3 5 - aba, 5 6 - abab, ...
- Dekodowanie:
 - 1 a 3 - a?
 - 2 ab 3 - ab, 4 - b?
 - 3 abab 4 - ba, 5 - ab?
 - 5 ababab?

Trudny przypadek

- Słowo abababababab... i słownik 1 - a i 2 - b.
- 1 3 - ab, 2 4 - ba, 3 5 - aba, 5 6 - abab, ...
- Dekodowanie:
 - 1 a 3 - a?
 - 2 ab 3 - ab, 4 - b?
 - 3 abab 4 - ba, 5 - ab?
 - 5 ababab?
- Ten problem można rozwiązać przez porównanie z algorytmem kodowania i wywnioskować, że 5 - aba.

Zastosowanie LZW

- Compress - kompresja w UNIX-ie.
- GIF
- Kompresja w modemie - V.42 bis

Algorytmy predykcyjne - Motywacje

- W tekstach naturalnych symbole bardzo często zależą od siebie.
- W językach naturalnych na podstawie już przeczytanych symboli można z bardzo dużym prawdopodobieństwem przewidzieć następny symbol (mocna zależność od historii).
- Często zamiast kompresować ciąg wejściowy kompresujemy różnicę między tym ciągiem a ciągiem generowanym przez pewien „zgadywacz”.

Przykład wykorzystania kontekstu

	A	B	C
A	0.1	0.3	0.6
B	0.6	0.1	0.3
C	0.3	0.6	0.1

- Łatwo sprawdzić, że $P(A) = P(B) = P(C) = \frac{1}{3}$, stąd normalny kod Huffmana będzie miał średnią długość $\frac{5}{3}$.
- Co będzie jeśli wykorzystamy informację o tym po jakiej literze występuje kolejna, czyli stworzymy trzy kodowania Huffmana, kolejno dla liter występujących po A, B, C?
- Łatwo policzyć, że wtedy średnia długość kodu spadnie do $\frac{13}{10}$.
- Gdybyśmy wydłużyli historię (kontekst) moglibyśmy uzyskać większy stopień kompresji, ale ilość kodów rosłaby szybko.

Predykcja z częściowym dopasowaniem (PPM)

- Algorytm dynamiczny wykorzystujący kontekst.
- Specjalny symbol wyjścia oznaczający brak istniejącego kontekstu danej długości (<esc>).
- Ustalamy maksymalny rozmiar kontekstu.
- Dla danej litery szukamy maksymalnego kontekstu, jeśli nie wysyłamy symbol wyjścia i sprawdzamy krótszy kontekst. Jeśli istnieje to wysyłamy odpowiedni kod a ilość użycia litery w tym kontekście zwiększamy o 1.
- Jeśli symbol pojawił się po raz pierwszy dodajemy go bez kontekstu, (kontekst -1) z prawdopodobieństwem równym dla każdej litery.

Przykład

- Kodujemy tekst: *this-is-the-tithe*.
- Przyjmujemy, że najdłuższy kontekst ma długość 2 (mamy konteksty długości -1, 0, 1 i 2).
- Po zakodowaniu *this-is* mamy następujące tabele kontekstów:

Przykład

- Kodujemy tekst: *this-is-the-tithe*.
- Przyjmujemy, że najdłuższy kontekst ma długość 2 (mamy konteksty długości -1, 0, 1 i 2).
- Po zakodowaniu *this-is* mamy następujące tabele kontekstów:

Przykład

- Kodujemy tekst: *this-is-the-tithe*.
- Przyjmujemy, że najdłuższy kontekst ma długość 2 (mamy konteksty długości -1, 0, 1 i 2).
- Po zakodowaniu *this-is* mamy następujące tabele kontekstów:

Przykład

- Kodujemy tekst: *this-is-the-tithe*.
- Przyjmujemy, że najdłuższy kontekst ma długość 2 (mamy konteksty długości -1, 0, 1 i 2).
- Po zakodowaniu *this-is* mamy następujące tabele kontekstów:

Kontekst długości -1

Litera	Licznik
t	1
h	1
i	1
s	1
-	1

Przykład

Kontekst długości 0

Litera	Licznik
<esc>	1
t	1
h	1
i	2
s	1
-	1

Przykład

Kontekst długości 1

Kontekst	Litera	Licznik
t	<esc>	1
	h	1
h	<esc>	1
	i	1
i	<esc>	1
	s	2
s	<esc>	1
	-	1
-	<esc>	1
	i	1

Przykład

Kontekst długości 2

Kontekst	Litera	Licznik
th	<esc>	1
	i	1
hi	<esc>	1
	s	1
is	<esc>	1
	-	1
s-	<esc>	1
	i	1
-i	<esc>	1
	s	1

- Dla każdego kontekstu możemy teraz utworzyć dynamiczne kody Huffmana.
- Albo wykorzystać dany kontekst z ilością wystąpień jako prawdopodobieństwa do podziału odcinka w kodowaniu arytmetycznym.
- Najczęściej za maksymalną długość kontekstu przyjmuje się 5.

CALIC - Context Adaptative Lossless Image Compression

		NN	NNE
	NW	N	NE
WW	W	X	

- Sprawdzamy czy w sąsiedztwie są krawędzie pionowe lub poziome, w tym celu liczymy wartości pomocnicze

$$d_h = |W - WW| + |N - NW| + |NE - N|$$

$$d_v = |W - NW| + |N - NN| + |NE - NNE|$$

CALIC - Context Adaptative Lossless Image Compression

- Pseudokod algorytmu:

```
if  $d_h - d_v > 80$  then  $\hat{X} \leftarrow N$ 
else if  $d_v - d_h > 80$  then  $\hat{X} \leftarrow W$ 
else
     $\hat{X} \leftarrow (N + W)/2 + (NE - NW)/4$ 
    if  $d_h - d_v > 32$  then  $\hat{X} \leftarrow (\hat{X} + N)/2$ 
    else if  $d_v - d_h > 32$  then  $\hat{X} \leftarrow (\hat{X} + W)/2$ 
    else if  $d_h - d_v > 8$  then  $\hat{X} \leftarrow (3\hat{X} + N)/4$ 
    else if  $d_v - d_h > 8$  then  $\hat{X} \leftarrow (3\hat{X} + W)/4$ 
```

- Predykcję \hat{X} można jeszcze bardziej uszczegółowić.
- Kodujemy ciąg różnic $X - \hat{X}$.

JPEG-LS (bezstratny)

NW	N
W	X

7 schematów predykcji:

1 $\hat{X} = W$

JPEG-LS (bezstratny)

NW	N
W	X

7 schematów predykcji:

1 $\hat{X} = W$

2 $\hat{X} = N$

JPEG-LS (bezstratny)

NW	N
W	X

7 schematów predykcji:

- 1 $\hat{X} = W$
- 2 $\hat{X} = N$
- 3 $\hat{X} = NW$

JPEG-LS (bezstratny)

NW	N
W	X

7 schematów predykcji:

1 $\hat{X} = W$

2 $\hat{X} = N$

3 $\hat{X} = NW$

4 $\hat{X} = N + W - NW$

JPEG-LS (bezstratny)

NW	N
W	X

7 schematów predykcji:

- 1 $\hat{X} = W$
- 2 $\hat{X} = N$
- 3 $\hat{X} = NW$
- 4 $\hat{X} = N + W - NW$
- 5 $\hat{X} = N + (W - NW)/2$

JPEG-LS (bezstratny)

NW	N
W	X

7 schematów predykcji:

- 1 $\hat{X} = W$
- 2 $\hat{X} = N$
- 3 $\hat{X} = NW$
- 4 $\hat{X} = N + W - NW$
- 5 $\hat{X} = N + (W - NW)/2$
- 6 $\hat{X} = W + (N - NW)/2$

JPEG-LS (bezstratny)

NW	N
W	X

7 schematów predykcji:

- 1 $\hat{X} = W$
- 2 $\hat{X} = N$
- 3 $\hat{X} = NW$
- 4 $\hat{X} = N + W - NW$
- 5 $\hat{X} = N + (W - NW)/2$
- 6 $\hat{X} = W + (N - NW)/2$
- 7 $\hat{X} = (N + W)/2$

JPEG-LS (bezstratny)

NW	N
W	X

Nowy standard:

if $NW \geq \max(W, N)$ then $\hat{X} \leftarrow \max(W, N)$
else if $NW \leq \max(W, N)$ then $\hat{X} \leftarrow \min(W, N)$
else $\hat{X} \leftarrow W + N - NW$

Wykorzystanie poziomów rozdzielczości

- Kodujemy obraz wysyłając najpierw średni kolor kwadratów $2^k \times 2^k$ a następnie różnice między tą średnią a kwadratami o rozmiarach $2^{k-1} \times 2^{k-1}$.
- Kończymy na pikslach (kwadraty $2^0 \times 2^0$).
- Różnice nie są dużymi liczbami i łatwo poddają się kompresji.

Kodowanie obrazów czarno-białych (faksy)

- Linie takich obrazów zawierają na przemian bloki białe i czarne.
- Możemy przesyłać więc tylko długości takich bloków (zakładając, że pierwszy jest biały).
- Dodatkowo każdą liczbę przedstawiamy jako parę m, t gdzie $l = 64m + t$ dla $t = 0, \dots, 63$ i $m = 0, \dots, 27$.
- Dodatkowo możemy kodować nie linię a jej różnicę z linią poprzednią (bardzo skuteczne w faksach).

Kompresja bezstratna – podsumowanie

- Techniki oparte na prawdopodobieństwach: kody Huffmana, Tunstalla, kodowanie arytmetyczne.
- Metody słownikowe: LZ77, LZ78, LZW.
- Kodowanie predykcyjne: PPM, CALIC, JPEG-LS, run-length encoding.
- Stosowanie kilku metod po kolei.