

Kompresa dat

Jan Outrata



KATEDRA INFORMATIKY
UNIVERZITA PALACKÉHO V OLMOUCI

přednášky



Kontextové (context-based) metody

- výskyt symbolu na vstupu není nezávislý na výskytu ostatních symbolů
- = pro modelování symbolu na vstupu využití kontextu délky k = posloupnost (bezprostředně) předchozích k symbolů → Markovův model k -tého řádu
- větší rozdíly v (podmíněných) pravděpodobnostech výskytu symbolů v kontextu jiných symbolů → lepší predikce symbolu (např. na konci slova) → vyšší míra komprese

Příklad

vstup *barbaraabarboraubaru*, $A = \{a, b, r, u, o\}$

$$f(a) = \frac{7}{20}, f(b) = \frac{5}{20}, f(r) = \frac{5}{20}, f(u) = \frac{2}{20}, f(o) = \frac{1}{20},$$

$$f(a|a) = \frac{1}{7}, f(b|a) = \frac{1}{7}, f(r|a) = \frac{4}{7}, f(u|a) = \frac{1}{7}, f(a|b) = \frac{4}{5}, f(o|b) = \frac{1}{5}, f(a|r) = \frac{2}{5},$$

$$f(b|r) = \frac{2}{5}, f(u|r) = \frac{1}{5}, f(b|u) = 1, f(r|o) = 1, \text{ostatní } f = 0$$

$$f(b|aa) = 1, f(a|ab) = 1, f(a|ar) = \frac{1}{4}, f(b|ar) = \frac{2}{4}, f(u|ar) = \frac{1}{4}, f(b|au) = 1,$$

$$f(r|ba) = 1, f(r|bo) = 1, f(a|ra) = \frac{1}{2}, f(u|ra) = \frac{1}{2}, f(a|rb) = \frac{1}{2}, f(o|rb) = \frac{1}{2},$$

$$f(a|ub) = 1, f(a|or) = 1, \text{ostatní } f = 0$$

- zvyšující se rozdíly v pravděpodobnostech výskytu symbolů s delším kontextem (menší počet symbolů v něm), ale počet kontextů délky k je $|A|^k$ (A abeceda symbolů)
- nezjišťovat pravděpodobnosti výskytu symbolu pro všechny kontexty dané délky, ale pouze pro kontexty na vstupu
- = adaptivní modelování pravděpodobností výskytu symbolů ve zkracujícím se kontextu jako posloupnosti bezprostředně předchozích symbolů
- Cleary, Witten, 1984
- speciální (escape) symbol ε abecedy značící neexistující/první výskyt symbolu na vstupu v aktuálním kontextu
- průběžné odhady (podmíněných) pravděpodobností $P(a_i|c^k)$ a distribučních funkcí/kumulovaných pravděpodobností $F_X(i|c^k)$ výskytu symbolů z abecedy $A|c^k \subseteq A \cup \{\varepsilon\} = \{a_1, a_2, \dots, a_n, \varepsilon\}$ jako náhodných proměnných $X(a_i|c^k) = i|c^k$ s frekvencemi/četnostmi $f(a_i|c^k) = \frac{n(a_i|c^k)}{\sum_{j=1}^{|A|c^k|} n(a_j|c^k)}$ výskytu symbolu $a_i \in A|c^k$ v kontextu c^k délky k
 - výskyt v kontextu $c^0 =$ výskyt (bez kontextu)
 - výskyt v kontextu $c^{-1} =$ neexistující/první výskyt, $A|c^{-1} = A$, $n(a_i|c^{-1}) = 1$ pro všechny $a_i \in A$

while načti ze vstupu symbol $a \in A$ **do**

$k \leftarrow k_{max} \leftarrow \min\{K, \text{počet načtených symbolů} - 1\}$;

while $k \geq 0 \wedge n(a|c^k) = 0$ pro (aktuální) kontext c^k symbolu a **do**

zapiš na výstup kód symbolu ε v kontextu c^k ;

$k \leftarrow k - 1$;

zapiš na výstup kód symbolu a v kontextu c^k ;

// algoritmus datové reprezentace $n(a|c^{k_{max}})$ nebo

$n(a|c^l) \leftarrow n(a|c^l) + 1$ pro všechny kontexty $c^l, k_{max} \geq l \geq 0$ symbolu a ;

- $f(\varepsilon|c^k)$? ze začátku (relativně) velká, s počtem zpracovaných symbolů klesající
 - metoda A (PPMA): $n(\varepsilon|c^k) = 1$
 - metoda B (PPMB): $n(\varepsilon|c^k) = |A|c^k \setminus \{\varepsilon\}|$ nebo 1 a $n(a|c^k) \leftarrow n(a|c^k) - 1$ pro $a \in A|c^k \setminus \{\varepsilon\}$ (při $n(a|c^k) = 0$ vyřazení a z $A|c^k \setminus \{\varepsilon\}$) – větší šance nového symbolu v kontextu s více symboly v kontextu?
 - metoda C (PPMC, Moffat): $n(\varepsilon|c^k) = |A|c^k \setminus \{\varepsilon\}|$ nebo 1 – v průměru nejvyšší míra komprese
 - PPMP: $f(\varepsilon|c^k) = \text{odhad } \frac{d_1}{d} - \frac{d_2}{d^2} + \frac{d_3}{d^3} - \dots$, d_i symbolů vyskytujících se na vstupu i -krát, na základě Poissonova rozdělení pravděpodobnosti výskytu každého z d symbolů na vstupu
 - PPMX: přibližná verze PPMP s odhadem $\frac{d_1}{d}$, PPMC při $d_1 = 0$ nebo $d_1 = d$

Příklad

vstup *barbaraabarboraubaru*, $A = \{a, b, r, u, o\}$, $K = 3$

$n(a|c^{-1}) = n(b|c^{-1}) = n(r|c^{-1}) = n(u|c^{-1}) = n(o|c^{-1}) = 1$, ostatní $n = 0$

vstup	<i>b</i>	<i>a</i>	<i>r</i>	<i>b</i>	
c^0	$n(b) = 1$	$n(a) = 1$	$n(r) = 1$	$n(b) = 2$	
c^1		$n(a b) = 1$	$n(r a) = 1$	$n(b r) = 1$	
c^2			$n(r ba) = 1$	$n(b ar) = 1$	
c^3				$n(b bar) = 1$	
výstup	$\varepsilon c^0 = \varepsilon, b c^{-1}$	$\varepsilon b, \varepsilon, a c^{-1}$	$\varepsilon ba, \varepsilon a, \varepsilon, r c^{-1}$	$\varepsilon bar, \varepsilon ar, \varepsilon r, b c^0 = b$	
	<i>a</i>	<i>r</i>	<i>a</i>	<i>a</i>	<i>b</i> ...
	$n(a) = 2$	$n(r) = 2$	$n(a) = 3$	$n(a) = 4$	$n(b) = 3$...
	$n(a b) = 2$	$n(r a) = 2$	$n(a r) = 1$	$n(a a) = 1$	$n(b a) = 1$...
	$n(a rb) = 1$	$n(r ba) = 2$	$n(a ar) = 1$	$n(a ra) = 1$	$n(b aa) = 1$...
	$n(a arb) = 1$	$n(r rba) = 1$	$n(a bar) = 1$	$n(a ara) = 1$	$n(b raa) = 1$...
	$\varepsilon arb, \varepsilon rb, a b$	$\varepsilon rba, r ba$	$\varepsilon bar, \varepsilon ar, \varepsilon r, a$	$\varepsilon ara, \varepsilon ra, \varepsilon a, a$	$\varepsilon raa, \varepsilon aa, \varepsilon a, b$...

$k \leftarrow k_{max} \leftarrow 0;$

while načti ze vstupu a dekoduj kód symbolu $a \in A|c^k$ v (aktuálním) kontextu c^k **do**

if $a = \varepsilon$ **then**

$k \leftarrow k - 1;$

else

 zapiš na výstup symbol $a \in A;$

 // algoritmus datové reprezentace $n(a|c^{k_{max}})$ nebo

$n(a|c^l) \leftarrow n(a|c^l) + 1$ pro všechny kontexty $c^l, k_{max} \geq l \geq 0$ symbolu $a;$

$k \leftarrow k_{max} \leftarrow \min\{K, \text{počet načtených kódů symbolů}\};$

- K ? co nejvyšší? – zvyšující se rozdíly v pravděpodobnostech výskytu symbolů s delším kontextem
 - vyšší pravděpodobnost kódování speciálního symbolu ε , delší kontexty bývají neaktuální – míra komprese nejprve s K prudce roste, ale pak mírně klesá \rightarrow v praxi 5 nebo 6 pro textová data
 - PPM*: v delších kontextech bývá výskyt jen jednoho symbolu = deterministický kontext $\rightarrow K$ = délka nejkratšího, jinak délka nejdelšího nedeterministického, varianta PPMZ (Bloom)
- kódování symbolů (adaptivním) aritmetickým kódováním s dynamickým pravděpodobnostním modelem podmíněným aktuálním kontextem c^k

Exclusion principle

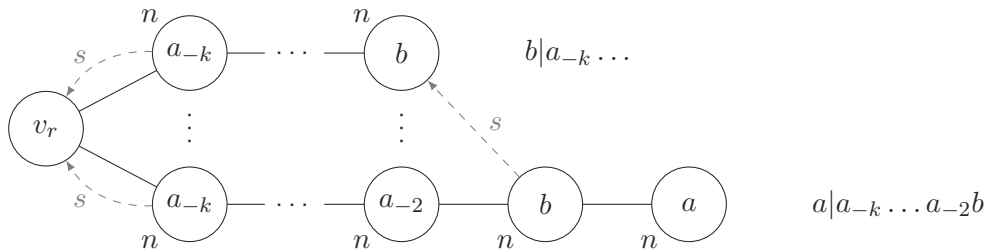
- menší abeceda znamená kratší kódy symbolů
- = když se načtený symbol a nevyskytl v (aktuálním) kontextu c^k délky $K \geq k \geq 0$, symboly vyskytující se v c^k , tzn. z $A|c^k \setminus \{\varepsilon\}$, lze vyřadit ze všech abeced $A|c^l$ kontextů c^l délky $-1 \leq l < k$ (protože kdyby byl býval některý načtený, zapsal by se na výstup v kontextu c^k jeho kód) – pouze pro kódování a , aktuální kontext se mění!

Příklad

při načtení prvního a nevyskytujícího se v c^0 lze vyřadit b z $A|c^{-1}$
prvního r ne v c^0 pak b, a z $A|c^{-1}$
třetího a ne v $c^3 = bar$ pak b z $A|ar, A|r, A|c^0, A|c^{-1}$
čtvrtého a ne v $c^1 = a$ pak r z $A|c^0, A|c^{-1}$
třetího b ne v $c^1 = a$ pak a, r z $A|c^0, A|c^{-1}$
...

Datová reprezentace $n(a|c^k)$, $a \in A$ pro všechny kontexty c^k , $K \geq k \geq 0 = n$ -ární strom
 $T = \langle V, E \rangle$ (n velikost A)

- listové uzly $v_l(a|c^{k_{max}}) \in V$ pro symboly $a \in A$ v kontextu $c^{k_{max}}$, $k_{max} = \max_{c^k} k$
 - vnitřní uzly $v(a_{-i}|c^{k-i}) \in V$ pro symboly a_{-i} , $i = k, \dots, 1$ v kontextu c^{k-i} ,
 $c^{k-i+1} = c^{k-i}a_{-i}$, $c^1 = a_{-k} +$ kořenový uzly $v_r \in V$
 - hrany $\langle v_r, v(a_{-k}|c^0) \rangle \in E$, $\langle v(a_{-i}|c^{k-i}), v(a_{-i+1}|c^{k-i+1}) \rangle \in E$ a
 $\langle v(a_{-1}|c^{k_{max}-1}), v_l(a|c^{k_{max}}) \rangle \in E$, nebo $\langle v_r, v_l(a|c^0) \rangle \in E$
- = trie = v uzlech část prvku (symbol a nebo a_{-i} kontextu c^k), ne celý prvek (kontext c^k)
- $s(v(b|c^k)) = v(b|c^{k-1})$, $k \geq 1$ a $s(v(b|c^0)) = v_r$ pro symbol $b \in A$



PPM (Prediction with partial match)



```
 $T \leftarrow \langle \{v_r\}, \emptyset \rangle;$   
 $v_p \leftarrow v_r;$   
// while ...  
 $v_q \leftarrow v_p, k \leftarrow k_{max};$   
while  $k \geq 0$  do  
  if  $k = K$  then  $v_p \leftarrow s(v_q)$  ;  
  if  $\langle v_q, v(a|c^k) \rangle \notin E$  then  
     $V \leftarrow V \cup \{v_l(a|c^k)\};$   
     $n(a|c^k) \leftarrow 1;$   
     $E \leftarrow E \cup \{\langle v_q, v_l(a|c^k) \rangle\};$   
     $v(a|c^k) \leftarrow v_l(a|c^k);$   
  else  
     $n(a|c^k) \leftarrow n(a|c^k) + 1;$   
  if  $v_q = v_p$  then  $v_p \leftarrow v(a|c^k)$  ;  
  if  $v_q \neq v_r$  then  $v_q \leftarrow s(v_q)$  ;  
   $k \leftarrow k - 1;$ 
```

Příklad

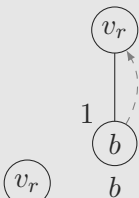
v_r

PPM (Prediction with partial match)



```
 $T \leftarrow \langle \{v_r\}, \emptyset \rangle;$   
 $v_p \leftarrow v_r;$   
// while ...  
 $v_q \leftarrow v_p, k \leftarrow k_{max};$   
while  $k \geq 0$  do  
  if  $k = K$  then  $v_p \leftarrow s(v_q)$  ;  
  if  $\langle v_q, v(a|c^k) \rangle \notin E$  then  
     $V \leftarrow V \cup \{v_l(a|c^k)\};$   
     $n(a|c^k) \leftarrow 1;$   
     $E \leftarrow E \cup \{\langle v_q, v_l(a|c^k) \rangle\};$   
     $v(a|c^k) \leftarrow v_l(a|c^k);$   
  else  
     $n(a|c^k) \leftarrow n(a|c^k) + 1;$   
  if  $v_q = v_p$  then  $v_p \leftarrow v(a|c^k)$  ;  
  if  $v_q \neq v_r$  then  $v_q \leftarrow s(v_q)$  ;  
   $k \leftarrow k - 1;$ 
```

Příklad

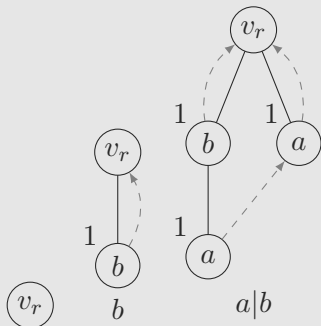


PPM (Prediction with partial match)



```
 $T \leftarrow \langle \{v_r\}, \emptyset \rangle;$   
 $v_p \leftarrow v_r;$   
// while ...  
 $v_q \leftarrow v_p, k \leftarrow k_{max};$   
while  $k \geq 0$  do  
  if  $k = K$  then  $v_p \leftarrow s(v_q)$  ;  
  if  $\langle v_q, v(a|c^k) \rangle \notin E$  then  
     $V \leftarrow V \cup \{v_l(a|c^k)\};$   
     $n(a|c^k) \leftarrow 1;$   
     $E \leftarrow E \cup \{\langle v_q, v_l(a|c^k) \rangle\};$   
     $v(a|c^k) \leftarrow v_l(a|c^k);$   
  else  
     $n(a|c^k) \leftarrow n(a|c^k) + 1;$   
  if  $v_q = v_p$  then  $v_p \leftarrow v(a|c^k)$  ;  
  if  $v_q \neq v_r$  then  $v_q \leftarrow s(v_q)$  ;  
   $k \leftarrow k - 1;$ 
```

Příklad

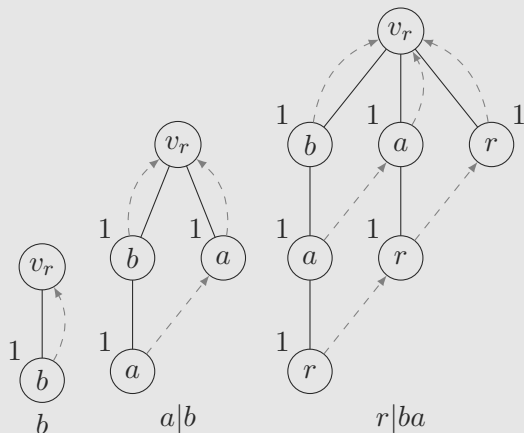


PPM (Prediction with partial match)



```
 $T \leftarrow \langle \{v_r\}, \emptyset \rangle;$   
 $v_p \leftarrow v_r;$   
// while ...  
 $v_q \leftarrow v_p, k \leftarrow k_{max};$   
while  $k \geq 0$  do  
  if  $k = K$  then  $v_p \leftarrow s(v_q)$  ;  
  if  $\langle v_q, v(a|c^k) \rangle \notin E$  then  
     $V \leftarrow V \cup \{v_l(a|c^k)\};$   
     $n(a|c^k) \leftarrow 1;$   
     $E \leftarrow E \cup \{\langle v_q, v_l(a|c^k) \rangle\};$   
     $v(a|c^k) \leftarrow v_l(a|c^k);$   
  else  
     $n(a|c^k) \leftarrow n(a|c^k) + 1;$   
  if  $v_q = v_p$  then  $v_p \leftarrow v(a|c^k)$  ;  
  if  $v_q \neq v_r$  then  $v_q \leftarrow s(v_q)$  ;  
   $k \leftarrow k - 1;$ 
```

Příklad



PPM (Prediction with partial match)

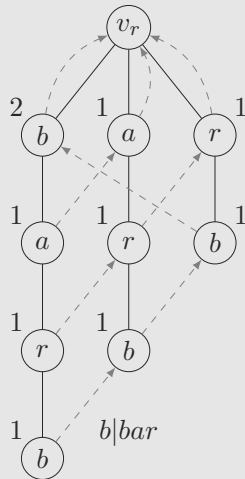
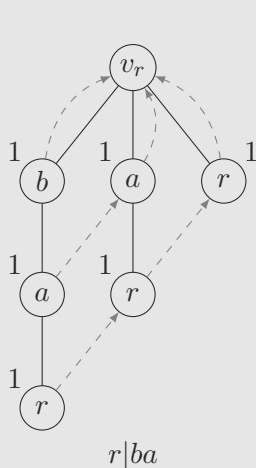


```

 $T \leftarrow \langle \{v_r\}, \emptyset \rangle;$ 
 $v_p \leftarrow v_r;$ 
// while ...
 $v_q \leftarrow v_p, k \leftarrow k_{max};$ 
while  $k \geq 0$  do
  if  $k = K$  then  $v_p \leftarrow s(v_q)$  ;
  if  $\langle v_q, v(a|c^k) \rangle \notin E$  then
     $V \leftarrow V \cup \{v_l(a|c^k)\};$ 
     $n(a|c^k) \leftarrow 1;$ 
     $E \leftarrow E \cup \{\langle v_q, v_l(a|c^k) \rangle\};$ 
     $v(a|c^k) \leftarrow v_l(a|c^k);$ 
  else
     $n(a|c^k) \leftarrow n(a|c^k) + 1;$ 
  if  $v_q = v_p$  then  $v_p \leftarrow v(a|c^k)$  ;
  if  $v_q \neq v_r$  then  $v_q \leftarrow s(v_q)$  ;
   $k \leftarrow k - 1;$ 

```

Příklad



PPM (Prediction with partial match)

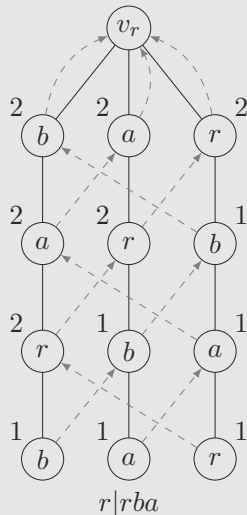
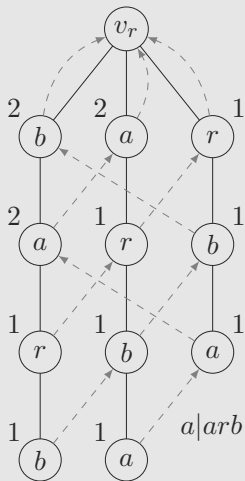


```

 $T \leftarrow \langle \{v_r\}, \emptyset \rangle;$ 
 $v_p \leftarrow v_r;$ 
// while ...
 $v_q \leftarrow v_p, k \leftarrow k_{max};$ 
while  $k \geq 0$  do
  if  $k = K$  then  $v_p \leftarrow s(v_q)$  ;
  if  $\langle v_q, v(a|c^k) \rangle \notin E$  then
     $V \leftarrow V \cup \{v_l(a|c^k)\};$ 
     $n(a|c^k) \leftarrow 1;$ 
     $E \leftarrow E \cup \{\langle v_q, v_l(a|c^k) \rangle\};$ 
     $v(a|c^k) \leftarrow v_l(a|c^k);$ 
  else
     $n(a|c^k) \leftarrow n(a|c^k) + 1;$ 
  if  $v_q = v_p$  then  $v_p \leftarrow v(a|c^k)$  ;
  if  $v_q \neq v_r$  then  $v_q \leftarrow s(v_q)$  ;
   $k \leftarrow k - 1;$ 

```

Příklad

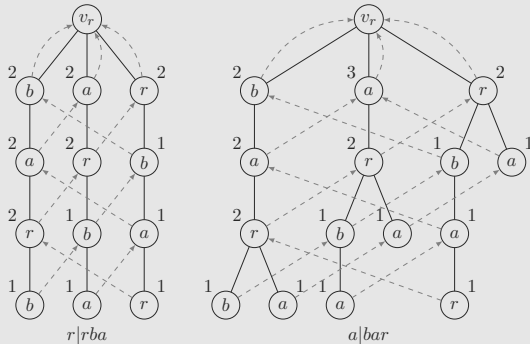


PPM (Prediction with partial match)



```
 $T \leftarrow \langle \{v_r\}, \emptyset \rangle;$   
 $v_p \leftarrow v_r;$   
// while ...  
 $v_q \leftarrow v_p, k \leftarrow k_{max};$   
while  $k \geq 0$  do  
  if  $k = K$  then  $v_p \leftarrow s(v_q)$  ;  
  if  $\langle v_q, v(a|c^k) \rangle \notin E$  then  
     $V \leftarrow V \cup \{v_l(a|c^k)\};$   
     $n(a|c^k) \leftarrow 1;$   
     $E \leftarrow E \cup \{\langle v_q, v_l(a|c^k) \rangle\};$   
     $v(a|c^k) \leftarrow v_l(a|c^k);$   
  else  
     $n(a|c^k) \leftarrow n(a|c^k) + 1;$   
  if  $v_q = v_p$  then  $v_p \leftarrow v(a|c^k)$  ;  
  if  $v_q \neq v_r$  then  $v_q \leftarrow s(v_q)$  ;  
   $k \leftarrow k - 1;$ 
```

Příklad

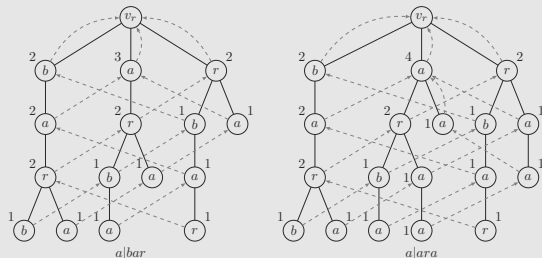


PPM (Prediction with partial match)



```
 $T \leftarrow \langle \{v_r\}, \emptyset \rangle;$   
 $v_p \leftarrow v_r;$   
// while ...  
 $v_q \leftarrow v_p, k \leftarrow k_{max};$   
while  $k \geq 0$  do  
  if  $k = K$  then  $v_p \leftarrow s(v_q)$  ;  
  if  $\langle v_q, v(a|c^k) \rangle \notin E$  then  
     $V \leftarrow V \cup \{v_l(a|c^k)\};$   
     $n(a|c^k) \leftarrow 1;$   
     $E \leftarrow E \cup \{\langle v_q, v_l(a|c^k) \rangle\};$   
     $v(a|c^k) \leftarrow v_l(a|c^k);$   
  else  
     $n(a|c^k) \leftarrow n(a|c^k) + 1;$   
  if  $v_q = v_p$  then  $v_p \leftarrow v(a|c^k)$  ;  
  if  $v_q \neq v_r$  then  $v_q \leftarrow s(v_q)$  ;  
   $k \leftarrow k - 1;$ 
```

Příklad



PPM (Prediction with partial match)



- pro každý symbol $a \in A$ na vstupu přidáno 0 až K uzlů stromu \rightarrow při velkém smazání a konstrukce nového z posledních několika symbolů (v praxi 2048)

- kontext nemusí být posloupnost bezprostředně předchozích symbolů na vstupu, jako u PPM
- kombinace modelů symbolu na vstupu využívajících různé kontexty = **context mixing**
- = adaptivní modelování pravděpodobností výskytu binárních symbolů (bitů, z binární zdrojové abecedy) kombinací modelů pro různé kontexty
- Matt Mahoney, 2002 a další (Osnach, Ratushnyak – PAQAR, Skibiński – PAsQDa, Taylor – RK) – několik verzí a mnoho variant (pro různé typy dat), free software, postupná vylepšení místo zcela nových metod z 80. až 90. let
- kontexty např. (posloupnosti):
 - bezprostředně předchozí bity, jako u PPM
 - bezprostředně předchozí (vícebitová) slova – slovo např. znak textu
 - nejvýznamnější bity bezprostředně předchozích slov – pro multimediální data (zvuk, obraz)
 - předchozí sousední slova vícedimenzionálních dat (stejná jako predikované ve stejné vzdálenosti) – např. obrazové body
 - vybraná předchozí slova (tzv. řídký kontext) – pro různé binární soubory, např. spustitelné, již (ztrátově) komprimované aj.

- i -tý model bitu na vstupu:

- počty $n_i(a)$, $a = \{\mathbf{0}, \mathbf{I}\}$ bitů v kontextu – verze 1 až 6, $n_i = n_i(\mathbf{0}) + n_i(\mathbf{I})$, částečná lokální adaptace: $n_i(a) \leftarrow n_i(a) + 1$ a $n_i(\mathbf{I} - a) \leftarrow \lfloor 1 + \frac{n_i(\mathbf{I} - a)}{2} \rfloor$ jestliže $n_i(\mathbf{I} - a) > 2$, a hodnota modelovaného bitu

- průběžné odhady (podmíněných) pravděpodobností $P_i(a|c)$, $a = \{\mathbf{0}, \mathbf{I}\}$ – od verze 7:

$$P_i(a|c) = \frac{n_i(a)}{n_i}$$

- průběžné odhady (podmíněných) pravděpodobností $P(a|c)$, $a = \{\mathbf{0}, \mathbf{I}\}$ kombinací modelů:

- $P(a|c) = \frac{s(a)}{s}$, $s = s(\mathbf{0}) + s(\mathbf{I})$, $s(a) = \epsilon + \sum_i w_i n_i(a)$ – verze 1 až 6, ϵ (experimentálně zjištěný) parametr pro nenulové $s(a)$, w_i váha i -tého modelu závisující na délce kontextu c – pevná (verze 1 až 3) nebo upravovaná pro minimalizaci chyby modelu a preferující přesnější modely (verze 4 až 6): $w_i \leftarrow \max[0, w_i + \frac{sn_i(\mathbf{I}) - s(\mathbf{I})n_i}{s(\mathbf{0})s(\mathbf{I})}(a - P(\mathbf{I}|c))]$

- neuronovou sítí – od verze 7: $P(\mathbf{I}|c) = \frac{1}{1 + e^{-\sum_i w_i x_i}}$, $x_i = \ln(\frac{P_i(\mathbf{I}|c)}{1 - P_i(\mathbf{I}|c)})$,

$$w_i \leftarrow w_i + \mu x_i (a - P(\mathbf{I})), \mu \text{ malá konstanta (míra adaptace)}$$

- kódování bitů adaptivním binárním aritmetickým kódováním (QM kódováním??) s dynamickým pravděpodobnostním modelem $\{P(\mathbf{0}|c), P(\mathbf{I}, c)\}$ – podobně jako v PPM

- ~ Burrows-Wheelerova transformace (BWT)
 - Michael Burrows, David J. Wheeler, 1994 (transformace Wheeler, 1983)
 - nepoužívá (podmíněně) pravděpodobnosti výskytu symbolů v kontextu jiných symbolů ani posloupností symbolů
 - blok b^k délky k : $c^{k-1}a$, a symbol na vstupu, kontext $c^{k-1} = a_{-k+1}a_{-k+2} \dots a_{-1} =$ posloupnost bezprostředně předchozích $k - 1$ symbolů symbolu a
- = permutace bloku b^k na blok obsahující podposloupnosti stejných symbolů (runs, viz dále)



- ~ Burrows-Wheelerova transformace (BWT)
 - Michael Burrows, David J. Wheeler, 1994 (transformace Wheeler, 1983)
 - nepoužívá (podmíněně) pravděpodobnosti výskytu symbolů v kontextu jiných symbolů ani posloupností symbolů
 - blok b^k délky k : $c^{k-1}a$, a symbol na vstupu, kontext $c^{k-1} = a_{-k+1}a_{-k+2} \dots a_{-1} =$ posloupnost bezprostředně předchozích $k - 1$ symbolů symbolu a
- = permutace bloku b^k na blok obsahující podposloupnosti stejných symbolů (runs, viz dále) → move-to-front (MTF) kódování permutovaného bloku

while načti ze vstupu nejvýše K symbolů jako blok $b^k = b_1^k \in A^+$ **do**
zapiš na výstup číslo k ;
 $i \leftarrow 2$;
while $i \leq k$ **do**
 $b_i^k \leftarrow$ rotace b_{i-1}^k o 1 symbol doleva;
 $i \leftarrow i + 1$;
seříd' b_1^k, \dots, b_k^k lexikograficky;
 $i \leftarrow 1$;
while $i \leq k$ **do**
 zapiš na výstup poslední symbol b_i^k ;
 $i \leftarrow i + 1$;
zapiš na výstup číslo i , kde $b^k = b_i^k$;

Příklad

vstup *barbaraabarboraubaru*,

$A = \{a, b, r, u, o\}, K = 20$

výstup: 20, *rabbbrruaurbaooaara*, 9

1	<i>barbaraabarboraubaru</i>	<i>aabarboraubarubarbar</i>
2	<i>arbaraabarboraubarub</i>	<i>abarboraubarubarbara</i>
3	<i>rbaraabarboraubaruba</i>	<i>araabarboraubarubarb</i>
4	<i>baraabarboraubarubar</i>	<i>arbaraabarboraubarub</i>
5	<i>araabarboraubarubarb</i>	<i>arboraubarubarbaraab</i>
6	<i>raabarboraubarubarba</i>	<i>arubarbaraabarboraub</i>
7	<i>aabarboraubarubarbar</i>	<i>aubarubarbaraabarbor</i>
8	<i>abarboraubarubarbara</i>	<i>baraabarboraubarubar</i>
9	<i>barboraubarubarbaraa</i>	<i>barbaraabarboraubaru</i>
10	<i>arboraubarubarbaraab</i>	<i>barboraubarubarbaraa</i>
11	<i>rboraubarubarbaraaaba</i>	<i>barubarbaraabarborau</i>
12	<i>boraubarubarbaraaabar</i>	<i>boraubarubarbaraaabar</i>
13	<i>oraubarubarbaraaabarb</i>	<i>oraubarubarbaraaabarb</i>
14	<i>raubarubarbaraaabarbo</i>	<i>raabarboraubarubarba</i>
15	<i>aubarubarbaraaabarbor</i>	<i>raubarubarbaraaabarbo</i>
16	<i>ubarubarbaraaabarbor</i>	<i>rbaraabarboraubaruba</i>
17	<i>barubarbaraaabarborau</i>	<i>rboraubarubarbaraaaba</i>
18	<i>arubarbaraaabarboraub</i>	<i>rubarbaraaabarboraub</i>
19	<i>rubarbaraaabarborau</i>	<i>ubarbaraaabarboraubar</i>
20	<i>ubarbaraaabarboraubar</i>	<i>ubarubarbaraaabarbor</i>

V lexikograficky seřazených blocích b_1^k, \dots, b_k^k z předchozího algoritmu:

Kódování

- posloupnost prvních symbolů bloků $b_i^k, i = 1, \dots, k$ obsahuje podposloupnosti stejných symbolů (kvůli seřazení b_i^k) \Rightarrow posloupnost posledních symbolů bloků b_i^k obsahuje podposloupnosti stejných symbolů, jestliže je v původním bloku b^k (a v každém bloku $\neq b_i^k$) před prvním symbolem bloku b_i^k (lokálně) opakovaně pro různá i tentýž poslední symbol bloku b_i^k (kvůli rotaci o 1 symbol doleva) = častý kontext

Dekódování

- posloupnost $F_i \in A, i = 1, \dots, k$ prvních symbolů bloků b_i^k v následujícím algoritmu = lexikograficky seřazená posloupnost $L_i \in A$ posledních symbolů bloků b_i^k = permutace φ posledních symbolů, $F_{\varphi(i)} = L_i \Leftrightarrow F_j = L_{\varphi^{-1}(j)}$ = „stejné za sebe“ se zachováním pořadí (lexikografické třídění)
- v původním bloku b^k (a v každém bloku $\neq b_j^k$) je za posledním symbolem L_j bloku b_j^k první symbol F_j bloku $b_j^k \Rightarrow F_{\varphi^{-1}(j)}$ je za $L_{\varphi^{-1}(j)} = F_j$

while načti ze vstupu číslo k **do**

načti ze vstupu k symbolů $L_1, \dots, L_k \in A$;

$F_i \leftarrow L_i$ pro $i = 1, \dots, k$;

setříd' F_1, \dots, F_k lexikograficky;

$a \leftarrow F_1$;

$i_F(a) \leftarrow 1$;

$i \leftarrow 2$;

while $i \leq k$ **do**

if $F_i \neq a$ **then**

$a \leftarrow F_i$;

$i_F(a) \leftarrow i$;

$i \leftarrow i + 1$;

// $i_F(a) = 1 + \sum_{b \in A} n_b$, n_b = počet stejných b před a v F_1, \dots, F_k

$i \leftarrow 1$;

while $i \leq k$ **do**

$\varphi^{-1}(i_F(L_i)) \leftarrow i$;

$i_F(L_i) \leftarrow i_F(L_i) + 1$;

$i \leftarrow i + 1$;

načti ze vstupu číslo j ;

$i \leftarrow 1$;

while $i \leq k$ **do**

zapiš na výstup symbol F_j ;

$j \leftarrow \varphi^{-1}(j)$;

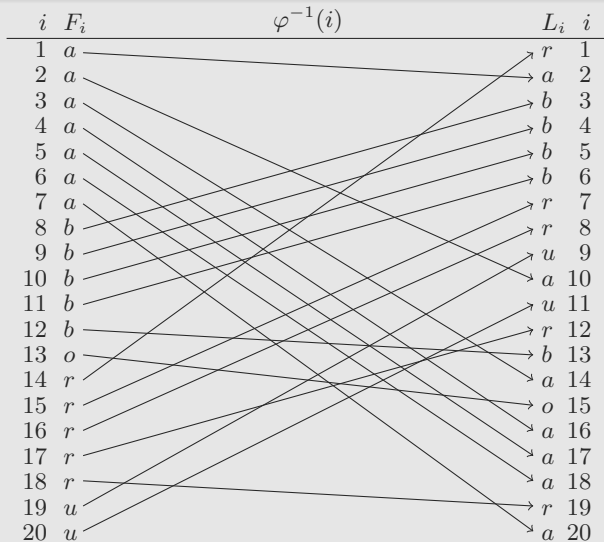
$i \leftarrow i + 1$;

Příklad

vstup: 20, rabbbrrruaurbaooaara, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup:

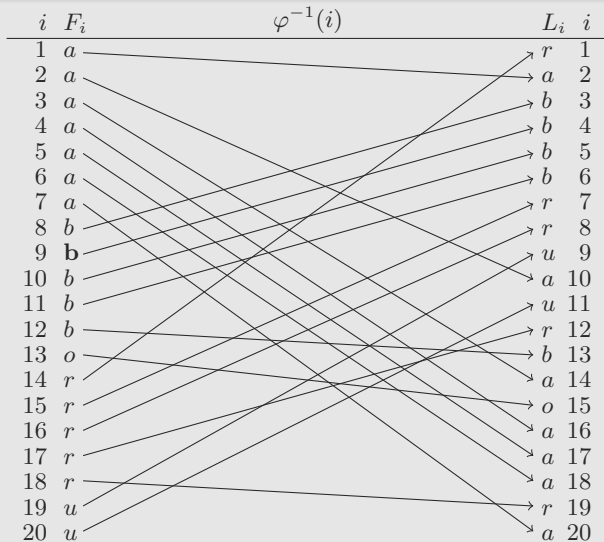


Příklad

vstup: 20, rabbbrrruaurbaooaara, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup: b

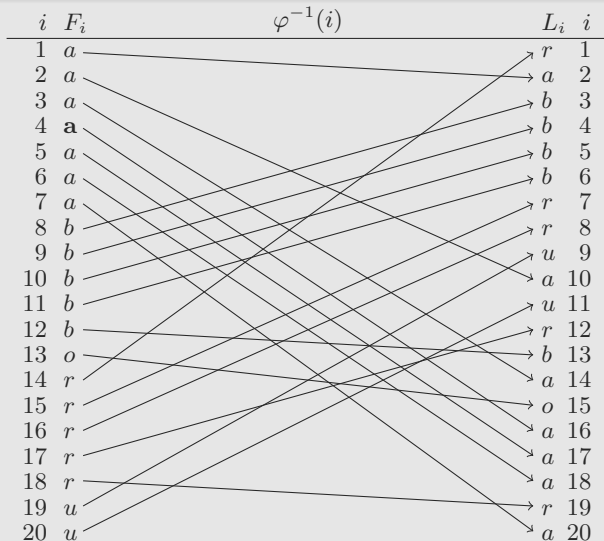


Příklad

vstup: 20, rabbbrrruaurbaooaara, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup: ba

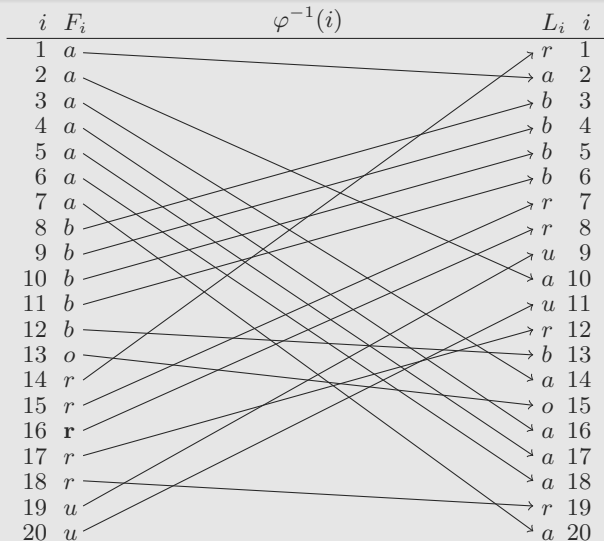


Příklad

vstup: 20, rabbbrrruaurbaooaara, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup: bar

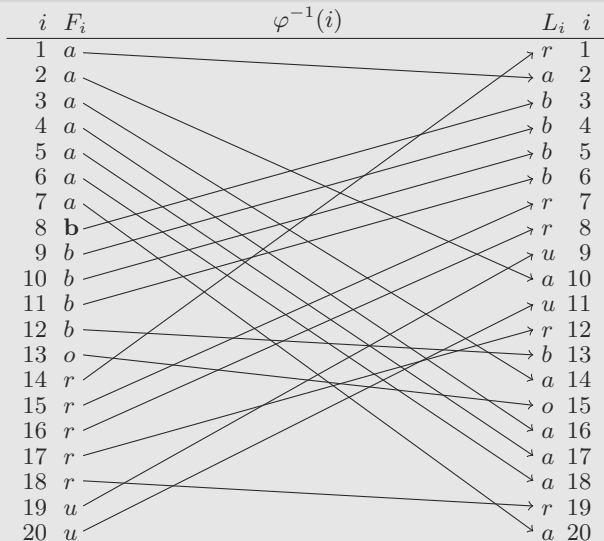


Příklad

vstup: 20, rabbbrrruaurbaooaara, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup: *barb*

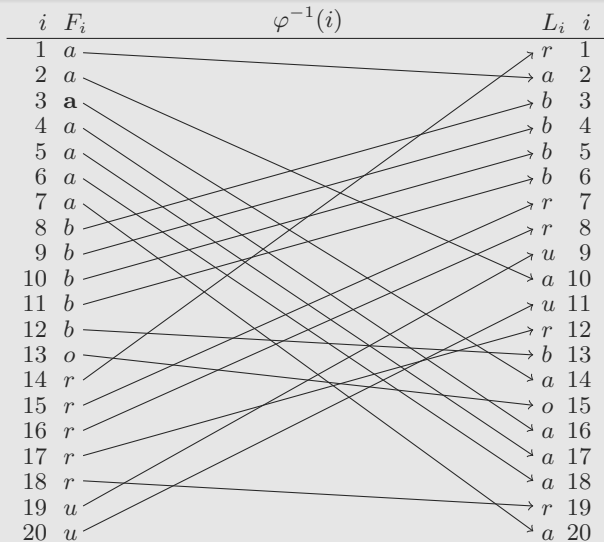


Příklad

vstup: 20, rabbbrrruaurbaooaara, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup: barba

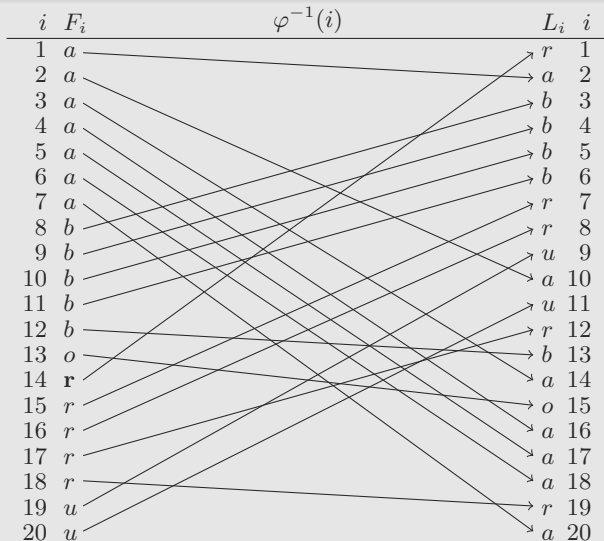


Příklad

vstup: 20, rabbbrrruaurbaooaara, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup: barbar

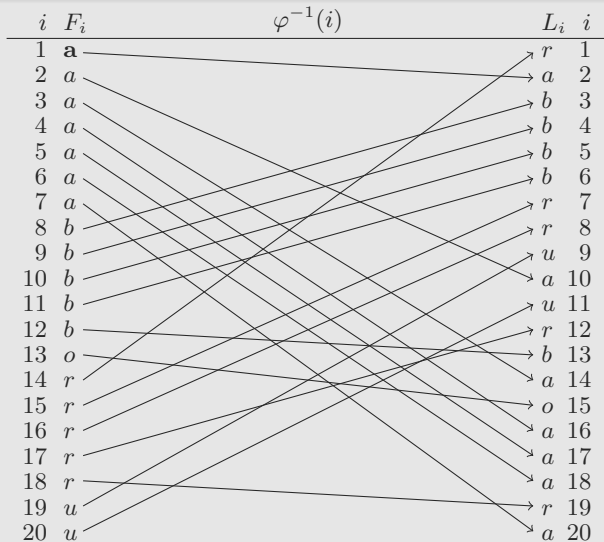


Příklad

vstup: 20, rabbbrrruaurbaooaara, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup: *barbara*

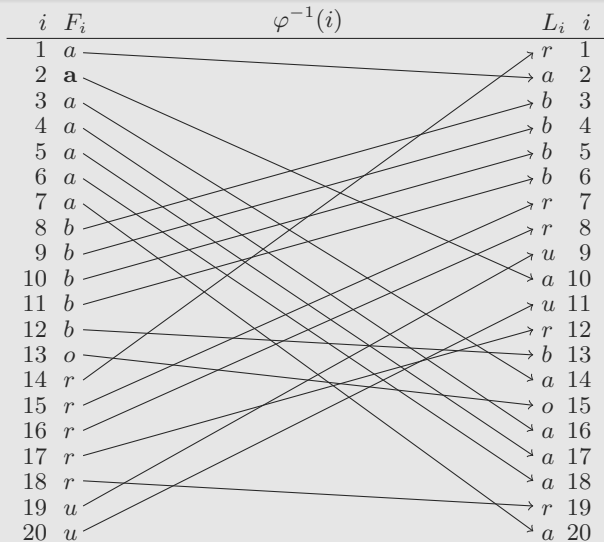


Příklad

vstup: 20, rabbbrrruaurbaooaara, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup: barbara

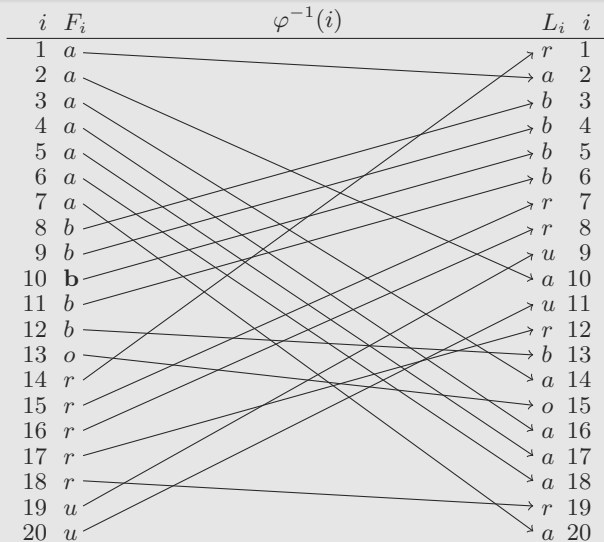


Příklad

vstup: 20, *rabbbrrruaurbaoaaara*, 9

$i_F(a) = 1$, $i_F(b) = 8$, $i_F(o) = 13$,
 $i_F(r) = 14$, $i_F(u) = 19$

výstup: *barbaraab*

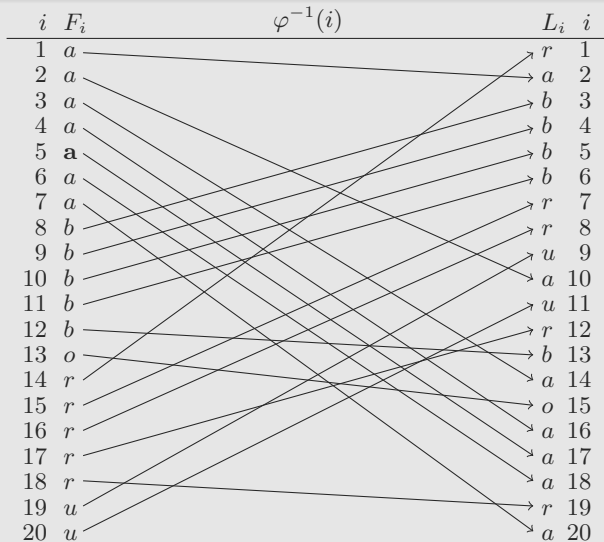


Příklad

vstup: 20, *rabbbrrruaurbaoaaara*, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup: *barbaraaba*

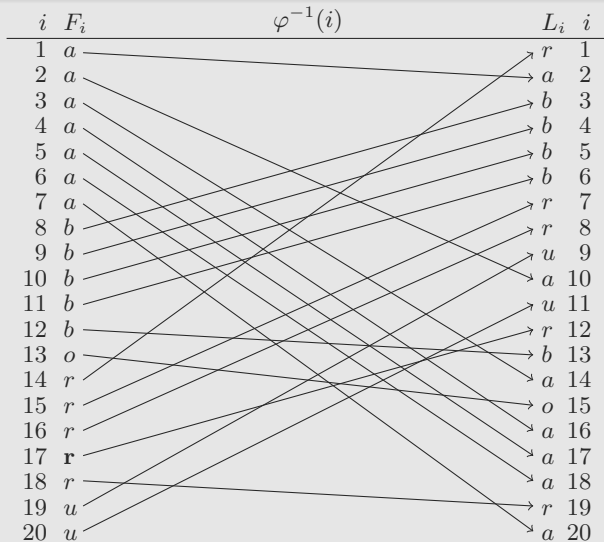


Příklad

vstup: 20, rabbbrrruaurbaooaara, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup: barbaraabar

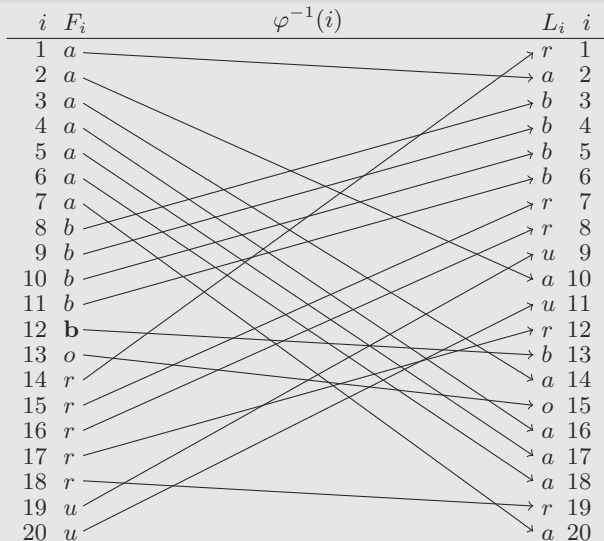


Příklad

vstup: 20, *rabbbrrruaurbaoaaara*, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup: *barbaraabarb*

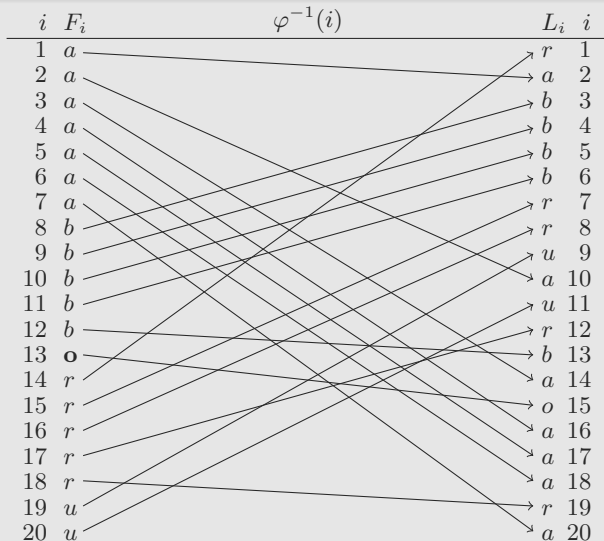


Příklad

vstup: 20, rabbbrrruaurbaooaara, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup: barbaraabarbo

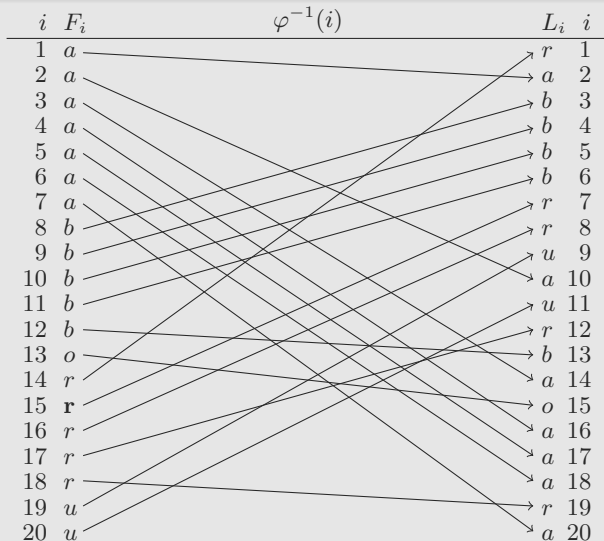


Příklad

vstup: 20, *rabbbrrruaurbaoaaara*, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup: *barbaraabarbor*

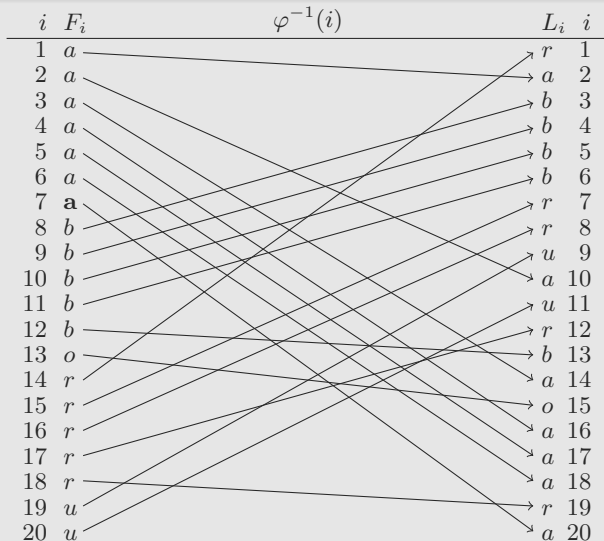


Příklad

vstup: 20, *rabbbrrruaurbaoaaara*, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup: *barbaraabarbor*a

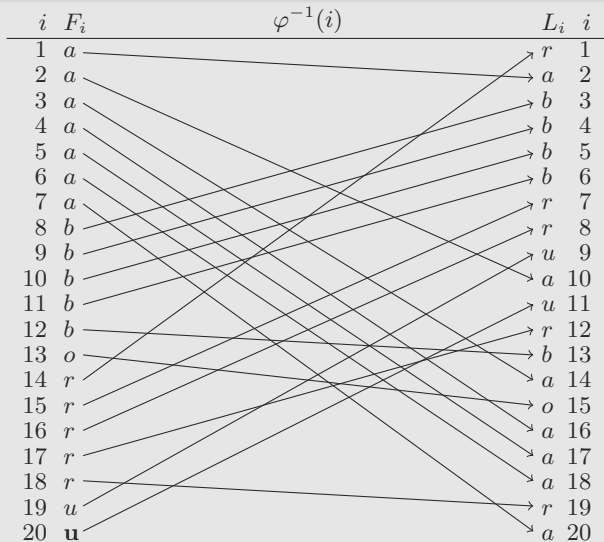


Příklad

vstup: 20, *rabbbrrruaurbaoaaara*, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup: *barbaraabarborau*

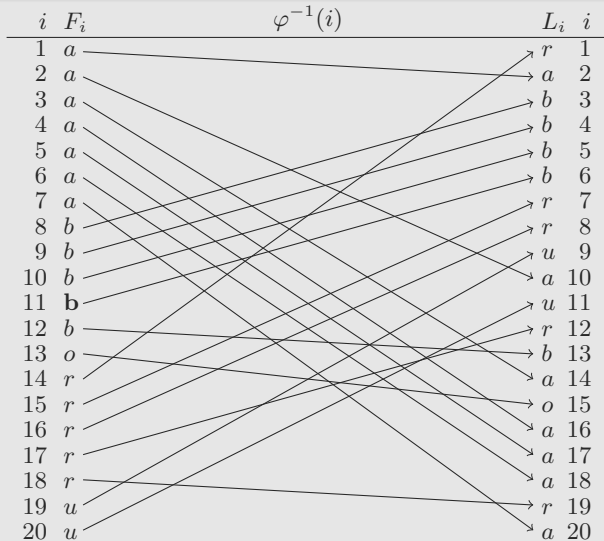


Příklad

vstup: 20, rabbbrrruaurbaooaara, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup: barbaraabarboraub

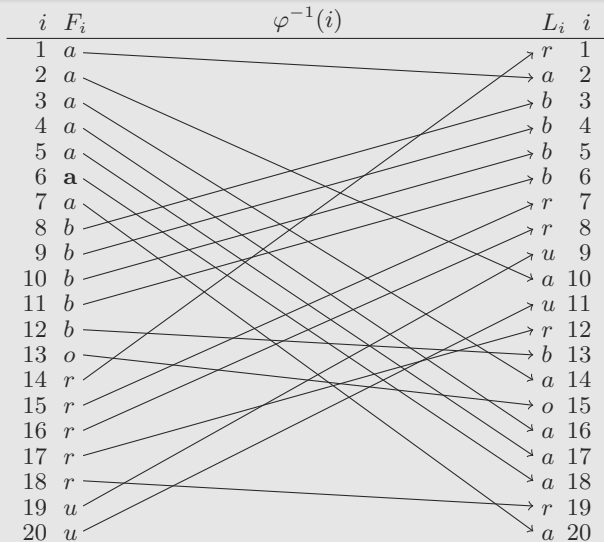


Příklad

vstup: 20, *rabbbrrruaurbaoaaara*, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup: *barbaraabarboraua*

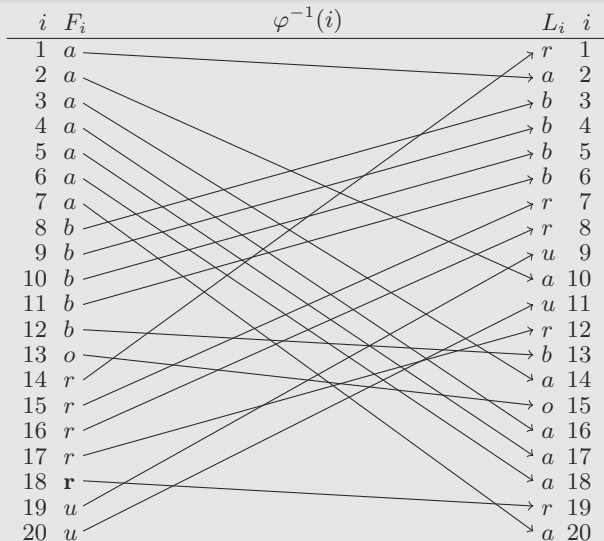


Příklad

vstup: 20, *rabbbrrruaurbaoaaara*, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup: *barbaraabarboraubar*

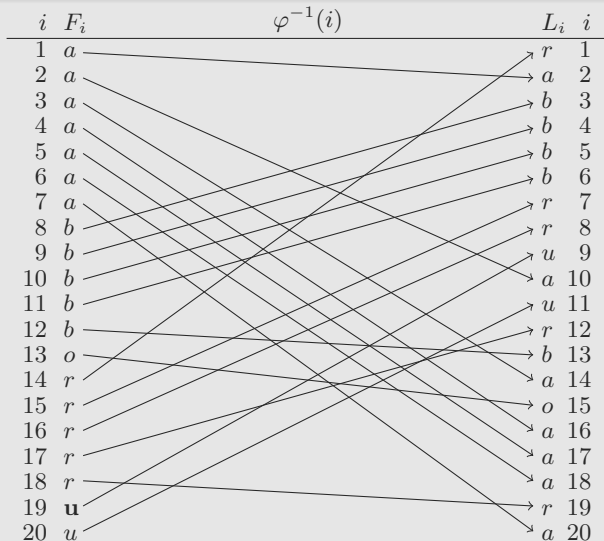


Příklad

vstup: 20, rabbbrrruaurbaooaara, 9

$i_F(a) = 1, i_F(b) = 8, i_F(o) = 13,$
 $i_F(r) = 14, i_F(u) = 19$

výstup: barbaraabarboraubaru



Implementace

- maximální délka K bloku až statisíce symbolů (bytů)
- kódování pouze s původním blokem $b^k - b_i^k$ ukazatele na první symbol bloku b_i^k v b^k
- dekódování pouze s posledními symboly $L_i \in A$ bloků $b_i^k, i = 1, \dots, k$
- move-to-front (MTF) kódování permutovaného bloku následované run-length kódováním (RLE) a statistickým kódováním (Huffmanovým nebo aritmetickým) – bzip