Plan na dziś

1. O problemie
2. Algorytm aproskymacyjny
3. Zastosowania
4. Dodatki

O problemie

- Definicja
- Ograniczenia algorytmów aproksymacyjnych
- Poprzednie algorytmy

Algorytm - rozkady

- Algorytm LZ77

Algorytm - rozkady

- Algorytm LZ77
- Rozkład LZ

Algorytm - rozkady

- Algorytm LZ77
- Rozkład LZ
- Rozkład względem gramatyki

Algorytm - rozkady

- Algorytm LZ77
- Rozkład LZ
- Rozkład względem gramatyki
- Ograniczenie dolne na wielkość gramatyki

Algorytm - rozkady

- Algorytm LZ77
- Rozkład LZ
- Rozkład względem gramatyki
- Ograniczenie dolne na wielkość gramatyki

  Komentarz o kompresji

Algorytm - balansowanie

- Gramatyki AVL

Algorytm - balansowanie

- Gramatyki AVL
- Wysokość drzewa wyprowadzenia

Algorytm - balansowanie

- Gramatyki AVL
- Wysokość drzewa wyprowadzenia
- Operacja konkatenaji gramatyk
  Koszt O(height(A) - height(B))

Algorytm - balansowanie

- Gramatyki AVL
- Wysokość drzewa wyprowadzenia
- Operacja konkatenaji gramatyk
  Koszt O(height(A) - height(B))
- Rozkad czynnika LZ f_i w gramatyce
  f_i=$val(S_1)val(S_2)...val(S_{t(i)})$ gdzie $t(i) = O(log(|f_i|)$

Algorytm - pseudokod

Construct-Grammar(w) |w| = n;

1. wywoaj algorytm LZ, otrzymucjąc rozkład $f_1, f_2, ...f_k$
2. jeśli $k > \frac{n}{\log(n)}$ to zwróć trywialny rozkad
3. w przeciwnym przypadku dla i od 1 do k wykonaj
   a) Niech $S_1, S_2, ..., S_{t(i)}$ będzie rozkładem czynnika f_i w G
   b) H := Concat(S_1,S_2,..., $S_{t(i)}$)
   c) G := Concat(G,H)
4. zwróć G

Algorytm - analiza

- Poszczególne kroki algorytmu

Algorytm - analiza

- Poszczególne kroki algorytmu
- Czas działania Concat z ciągniem bitionicznym

Algorytm - analiza

- Poszczególne kroki algorytmu
- Czas działania Concat z ciągniem bitionicznym
- Czas działania i stopień aproksymacji

Algorytm - analiza

- Poszczególne kroki algorytmu
- Czas działania Concat z ciągniem bitionicznym
- Czas działania i stopień aproksymacji
- Czas działania LZ77 - przy użyciu drzew sufiksowych $O(n \log(|\Sigma|))$

Zastosowania

- Wyszukiwanie wzorca

Zastosowania

- Wyszukiwanie wzorca
- Problem łańcucha

Zastosowania

- Wyszukiwanie wzorca
- Problem łańcucha
- Modyfikacja złożoności Komogorowa

Dodatki

- Algorytmy globalne (re-pair)

Dodatki

- Algorytmy globalne (re-pair)
- Blisko optymalnoci

Dodatki

- Algorytmy globalne (re-pair)
- Blisko optymalnoci

Niech $\sigma = x^{k_1}|...|x^{k_q}$, przy czym k_1 jest najwiksze z k_i

Dodatki

- Algorytmy globalne (re-pair)
- Blisko optymalnoci

Niech $\sigma = x^{k_1}|...|x^{k_q}$, przy czym k_1 jest największe z k_i
Rozkład LZ
$\lambda = x(1,1)(1,2)(1,4)...(1,k_1 - 2^j)|(1,k_2)|...|(1,k_q)$
gdzie $j = |(log(k_1))|$

Dodatki

- Algorytmy globalne (re-pair)
- Blisko optymalnoci

Niech $\sigma = x^{k_1}|...|x^{k_q}$, przy czym k_1 jest najwiksze z k_i
Rozkład LZ
$\lambda = x(1,1)(1,2)(1,4)...(1,k_1 - 2^j)|(1,k_2)|...|(1,k_q)$
gdzie $j = |(log(k_1))|$
Dla $q = \Theta(log(k_1))|\lambda| = O(log(k_1))$

Dodatki- fina

- Dla L-dugość najkrótszego łańcucha i R-rozmiar najmniejszej gramatyki mamy $L \leq R \leq 4 * L$

Dodatki- fina

- Dla L-dugość najkrótszego łańcucha i R-rozmiar najmniejszej gramatyki mamy $L \leq R \leq 4 * L$
- Twierdzenie o problemie łańcucha Istniej liczby $k_1, k_2..., k_q$ takie, że dugość najkrótszego łańcucha wynosi $\Omega(log(k_1) + q * \frac{log(k_1)}{loglog(k_1)+log(q)})$

Dodatki- fina

- Dla L-dugość najkrótszego łańcucha i R-rozmiar najmniejszej gramatyki mamy $L \leq R \leq 4 * L$
- Twierdzenie o problemie łańcucha Istniej liczby $k_1, k_2..., k_q$ takie, że dugość najkrótszego łańcucha wynosi $\Omega(log(k_1) + q * \frac{log(k_1)}{loglog(k_1)+log(q)})$
- Zatem, dla naszego q, rozmiar najmniejszej gramatyki wynosi $\Omega(\frac{log^2(k_1)}{loglog(k_1)})$

Koniec

Dziękuję za uwagę