

# Gwarantowana synchronizacja kodów Huffmana

**Marek Tomasz Biskup**

**Uniwersytet Warszawski**

## Streszczenie

Błąd bitu w ciągu zakodowanym kodem Huffmana może prowadzić do utraty synchronizacji przez dekodery. Rozsynchronizowany dekodery widzi nieprawidłowe granice słów kodowych i, w konsekwencji, dekoduje przypadkowy ciąg znaków. W przypadku większości kodów taki dekodery w końcu się zsynchronizuje i będzie dalej dekodował poprawne symbole. Niestety, taka spontaniczna synchronizacja jest tylko statystyczna i nie ma ograniczenia górnego na liczbę bitów koniecznych do przeczytania zanim taka synchronizacja nastąpi (na opóźnienie synchronizacji).

Moja praca jest związana z ograniczeniem opóźnienia synchronizacji do pewnej zadanej z góry wartości. Metoda polega na wstawianiu w pewne miejsca dodatkowych bitów, które pomagają rozsynchronizowanemu dekodery odzyskać synchronizację, a są ignorowane przez zsynchronizowany dekodery. Metoda wykorzystuje naturalną tendencję kodów Huffmana do synchronizacji. W optymistycznym przypadku, kiedy spontaniczna synchronizacja zawsze wystarcza, żadna redundancja nie jest dodawana. Dla testowanych plików, przy opóźnieniu synchronizacji rzędu 400 bitów, względna redundancja wyniosła  $5 \cdot 10^{-6}$ .

## Dodatkowe informacje

Część kodów Huffmana posiada tzw. *ciąg synchronizujący*. Jest to ciąg słów kodowych, po przeczytaniu którego dekodery jest zawsze zsynchronizowany. Taki ciąg synchronizujący  $s$  można scharakteryzować warunkiem, że dla dowolnego ciągu bitów  $x$ , ciąg  $xs$  jest ciągiem słów kodowych. Niektóre kody Huffmana mają *synchronizujące słowo kodowe*, czyli słowo kodowe, które jest ciągiem synchronizującym. Jeśli takie słowo synchronizujące  $c$  będzie wstawiane co średnio  $L$  bitów, to opóźnienie synchronizacji będzie ograniczone do  $(L + \text{kilka})$  bitów. W takim wypadku konieczne jest aby  $c$  nie odpowiadało żadnej literze alfabetu źródłowego, aby dekodery mógł je pominąć ( $c$  może jednak być sufiksem innego słowa kodowego). Ta metoda jest więc nieoptymalna, ponieważ eliminacja słowa  $c$  z kodu zmniejszyłaby średnią długość słowa kodowego. Co więcej, sposób wstawiania słowa synchronizującego można poprawić.

Zaproponowana przeze mnie metoda polega na śledzeniu synchronizacji dekodery zaczynających dekodowanie od wybranego bitu zakodowanego ciągu. Znacznik synchronizujący będzie wstawiony tylko jeśli jakiś dekodery nie zsynchronizuje się spontanicznie w ciągu  $L$  pierwszych bitów, które czyta. Śledzenie synchronizacji wszystkich możliwych dekodery można wykonać podczas kodowania ciągu. Pesymistyczny narzut czasowy wynosi jedynie  $O(1)$  na każdy zakodowany bit. Jako znacznik synchronizujący może być wzięty dowolny ciąg synchronizujący. Znacznik taki może również występować naturalnie w kodowanym ciągu, a dekodery usunie tylko znaczniki wstawiane przez koder.

Gwarantowana synchronizacja może być użyta np. do ograniczenia propagacji błędów bitowych lub do dekodowania wybranego fragmentu zakodowanego ciągu.

Metody synchronizacji opracowałem wspólnie z dr hab. Wojciechem Plandowskim, prof. UW.