

Algorytm Turbo_BM

Adrian Siwiec

June 19, 2020

Opis Algorytmu.

Algorytm Turbo_BM jest modyfikacją algorytmu Boyera-Moore'a, która zużywa tylko stałą dodatkową pamięć oraz przyspiesza złożoność pesymistyczną do $2n$. Nie będziemy wykonywać żadnego dodatkowego preprocessingu – tak jak w oryginalnym algorytmie korzystać będziemy z tablicy BM obliczonej dla wzorca.

Sam pomysł algorytmu Turbo_BM jest bardzo prosty – podczas skanowania będziemy zapisywać jedno podśłowo wzorca, o którym wiemy że pasuje do tekstu na obecnej pozycji. Dzięki temu możemy pominąć ten fragmeny przy sprawdzaniu dopasowania, oraz w przypadku jego braku możemy wykonać tzw. *Turbo-shift*.

Niech x będzie najdłuższym sufixem, który pasuje do tekstu na danej pozycji wzorca, a będzie pierwszą literą wzorca która nie pasuje, a niech y (*memory*) będzie poprzednio zapamiętanym podśłowem, które również pasuje do wzorca (patrz fig. 1). Założmy również, że x i y są rozłączne i y jest dłuższy od x . Po wykonaniu Turbo-shift zapominamy o y (patrz kod *boyer_moore_turbo*), więc w poprzednim kroku wykonaliśmy zwykłe przesunięcie (zgodnie z tablicą *BM*), znane z algorytmu Boyera-Moore'a. W tej sytuacji ax jest sufixem y , oraz litery a i b tekstu są o siebie oddalone o $|y|$. Ale suffix $y \dots x$ ma okres długości $|y|$ (z definicji *BM*). Możemy więc przesunąć wzorzec o $|y| - |x|$ do przodu i to właśnie przesunięcie nazywać będziemy *Turbo-shift*.

Analiza złożoności.

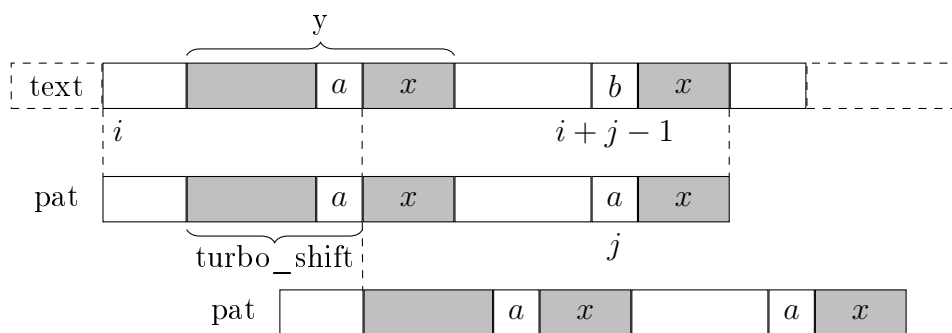


Figure 1: Turbo-shift