김일성종합대학학보
수 학

주체109(2020)년
제66권 제4호

흐름길이부호에 의한 확장된 바러우－윌러변환의 무리짓기효과평가

김철은, 안금성

경애하는 최고령도자 **김정은**동지께서는 다음과 같이 말씀하시였다.

《과학기술을 확고히 앞세우고 과학기술과 생산을 밀착시키며 경제건설에서 제기되는 모든 문제들을 과학기술적으로 풀어나가는 기풍을 세워 나라의 경제발전을 과학기술적으로 확고히 담보하여야 합니다.》

론문에서는 확장된 바러우－윌러변환의 무리짓기효과가 나타나지 않는 경우에 대하여 연구하였다.

선행연구[1]에서는 단어의 문자우에서 가역변환인 바러우－윌러변환(BWT)을 내놓았으며 오늘날 많은 본문압축기들이 이 변환에 기초하고있고 생물정보학과 콤퓨터생물학을 비롯한 여러 분야에서 리용되는 도구들도 이 변환에 기초하고있다. BWT는 실제적인 압축기는 아니지만 입력본문의 문자들을 문맥종속치환하는 변환이다. 이 치환은 자주 초기본문의 같은 문자들의 흐름(무리)보다 더 긴 무리들을 산생한다. 보통 이 성질을 BWT의 무리짓기효과라고 부른다.

선행연구[2]에서는 바러우－윌러변환의 변형으로서 확장된 바러우－윌러변환(eBWT)을 내놓았으며 이 변환은 같은 블로크길이에 대하여 초기변환보다 수행이 더 좋다는것을 실험적으로 보여주었다.

선행연구[3]에서는 BWT의 무리짓기효과가 나타나지 않는 경우를 연구하였다. 단어의 같은문자흐름수인 복잡도를 정의하고 BWT가 무리짓기효과를 나타내지 못하는 많은 단어들이 존재한다는것을 보여주었다.

우리는 선행연구에서 정의한 복잡도를 리용하여 확장된 바러우－윌러변환이 무리짓기효과를 나타내지 못하는 단어들의 개수를 평가하였다.

바러우－윌러변환을 일반화하는 한가지 방법은 순서화단계에서 리용되는 순서를 개선하는것이다. 초기 BWT에서 리용한 사전식순서는 순서의 한가지 실례이다. 선행연구[2]에서는 사전식순서와는 다른 순서를 리용하는 BWT의 확장을 내놓았다.

새로운 순서는 BWT변환이 단일한 단어가 아니라 단어들의 다중모임을 처리할수 있게 확장한다.

$<$를 표준사전식순서라고 하고 $\Sigma=\{a_1, a_2, \cdots, a_\sigma\}$를 유한순서자모라고 하자. Σ^*을 Σ 우의 단어들의 모임이라고 하자. 유한단어 $w=w_1w_2\cdots w_n\in\Sigma^*$, $w_i\in\Sigma$가 주어졌을 때 w의 길이를 $|w|$로, w에서 나타나는 모든 문자들로 이루어진 Σ의 부분모임을 $alph(w)$로 표시한다. 유한단어 $w=w_1w_2\cdots w_n\in\Sigma^*$, $w_i\in\Sigma$가 주어지면 단어 w의 인자를 $w[i, j]=w_i\cdots w_j$, $1\le i\le j\le n$ 으로 표시한다. $w[1, j]$ 형의 인자를 앞붙이라고 부르며 $w[i, n]$ 형의 인자를 뒤붙이라고 부른다. w의 부분렬은 w에서 몇개의 문자(반드시 련이어 있지 않는)들을 제거하여 얻은 단어이다. 임의의 $1\le i\le n$에 대하여 w의 i째 문자를 $w[i]$로 표시한다.

두 단어 w와 v의 련접을 wv로 표시하는데 그것은 단순히 w의 문자들을 먼저 놓고 그뒤에 v의 문자들을 련이어 놓아 구성한 단어이다.

자모 Σ가 주어졌을 때 두 문자렬 $S_1,\ S_2 \in \Sigma^*$은 어떤 $u,\ v \in \Sigma^*$이 있어서 $S_1 = uv,\ S_2 = vu$이면 공액이라고 부른다. 두 공액인 문자렬들은 서로 순환회전관계에 있다. 단어 $w \in \Sigma^*$의 공액클라스(w)는 모든 $1 \le i \le n$과 $w_i \in \Sigma$에 대하여 모든 단어 $w_i w_{i+1} \cdots w_n w_1 \cdots w_{i-1}$들의 모임이다.

비지 않은 단어 $w \in \Sigma^*$는 $w = v^k$으로부터 $w = v$이고 $k = 1$이 얻어지면 원시적이라고 한다. 임의의 비지 않은 단어 $u \in \Sigma^*$은 유일한 방법으로 원시단어의 제곱으로 표시할수 있다. 즉 $u = w^k$인 u의 뿌리라고 부르는 유일한 원시단어 w와 유일한 옹근수 k가 존재한다. 이때 u의 뿌리 w를 $w = root(u)$로 표시한다.

공액클라스에서 사전식순서로 최소인 원시단어를 런든단어라고 부른다.

단어 w에서 $w_i w_{i+1} \cdots w_{i+l_i-1}$을 문자 w_i의 최대흐름(즉 $w_i = w_{i+1} = \cdots = w_{i+l_i-1}$, $w_{i-1} \ne w_i,\ \ w_{i+l_i} \ne w_i$)이라고 할 때 모든 최대흐름들을 나타나는 순서로 쌍 $(w_i,\ l_i)$들의 렬로 렬거하는것을 흐름길이부호화라고 부르고 $rle(w)$로 표시한다. w에서 쌍의 개수 혹은 동등하게 w에서 같은 문자흐름들의 개수를 $\rho(w) = |rle(w)|$로 표시한다. 또한 $w_j = a_i$일 때 $rle(w)$에서 쌍 $(w_j,\ l_j)$들의 개수를 $\rho(w)_{a_i}$로 표시한다.

$w_1 w_2 \cdots w_p = w$가 w의 임의의 분할일 때 $\rho(w) \le \rho(w_1) + \rho(w_2) + \rho(w_p)$이다.

$u^\omega = uuu\cdots$은 u가 무한번 반복되는 문자렬이다. 분명히 두 렬 u^ω와 v^ω는 $root(u) = root(v)$일 때 그리고 다만 그때에만 같다. 즉 두 렬 u와 v는 같은 원시단어의 제곱일 때 그리고 다만 그때에만 같다.

문자렬 $u,\ v \in \Sigma^*$사이에 순서관계를 다음과 같이 정의한다:

$$u \le_\omega v \Leftrightarrow \begin{cases} \exp(u) \le \exp(v),\ root(u) = root(v) \\ u^\omega < v^\omega, \qquad \text{기타} \end{cases}$$

$u \le_\omega v$는 u와 v사이에 완전순서관계를 준다. 분명히 $u \le_\omega v$는 일반적으로 사전식순서 $u < v$와 다르다. 실례로 $u = ab$이고 $v = aba$일 때 $u < v$이지만 $v \le_\omega u$이다.

$T = \{S_1,\ S_2,\ \cdots,\ S_s\}$를 원시단어렬들의 다중모임이라고 하자. 원시단어렬들이 아닌 경우 매 문자렬은 끝에 단순히 기호를 첨부하여 원시단어로 만들수 있다. 확장된 BWT변환은 T우에서 다음의 걸음들을 리용하여 수행된다.

걸음 1 T의 매 단어의 공액들을 구성한다. 공액들의 행렬 A를 구성하는데 A의 매 행은 구성한 공액들중에서 정확히 1개의 단어에 대응한다. 여기서 행들은 모두 같은 수의 렬을 가지지 않아도 된다. 결과에 영향을 주지 않으면서 행들에 원소들을 덧붙여 모든 행들이 같은 개수의 렬을 가지도록 할수 있다.

걸음 2 공액들의 행렬 A를 새로운 순서관계에 따라서 순서화한다. $A_s = w_1,\ w_2,\ \cdots,\ w_m$을 순서화된 공액들의 목록이라고 하자. 즉 $1 \le i < j \le m$에 대하여 $w_i \le_\omega w_j$이다.

걸음 3 $a = \{a_1,\ a_2,\ \cdots,\ a_s\}$를 A_s에서 T의 원래단어들의 위치를 나타내는 첨수모임이라고 하자. 여기서 a_i는 A_s에서 S_i의 위치이다.

걸음 4 L을 A_s에서 마지막문자들의 렬이라고 하자. 즉 $1 \le i \le m$에 대하여 $L[i] = w_i[|w_i|]$

는 w_i의 마지막기호이다. 류사하게 $F[i]=w_i[1]$이 렬 w_i의 첫번째 기호인 렬 F를 정의한다.

걸음 5 확장된 BWT변환의 출력은 쌍 (L, a)이다.

확장된 BWT는 가역이며 가역변환의 절차는 BWT의 가역변환절차와 매우 류사하다.

BWT변환을 연구하는 많은 론문들이 BWT의 무리짓기효과에 초점을 집중하였다. 그런데 BWT의 무리짓기효과가 나타나지 않는 많은 실례들을 찾을수 있다. 확장된 BWT에 대하여서도 마찬가지이다.

실례 $T=\{aacb, bccc\}$를 단어들의 다중모임이라고 하면 $\rho(T)=4$, $\rho(e\mathrm{BWT}(T))=6$이다.

론문에서는 확장된 BWT변환의 출구에서의 같은 문자흐름수가 입구에서의 같은 문자흐름수보다 더 많아지는 경우를 고찰한다.

우선 $(e\mathrm{BWT}(T))$가 $\rho(T)$에 비해볼 때 얼마나 커질수 있는가 하는 문제를 고찰하자.

우리는 단어에서 다만 같은 문자흐름수에만 관심을 가지므로 입력하는 다중모임을 원시런든단어들의 다중모임이라고 가정하겠다. 왜냐하면 런든단어가 공액인 단어들중에서 최소의 같은 문자흐름을 가지는 단어부류에 속하기때문이다.

정리 1 $T=\{S_1, S_2, \cdots, S_s\}$가 유한자모 Σ우에서 원시런든단어들의 다중모임이라고 하자. 그러면

$$\rho(e\mathrm{BWT}(T))\le 2\rho(T)$$

가 성립한다.

확장된 바러우－윌러변환의 적용이 입력본문의 같은 문자흐름수를 얼마나 증가시키는가를 평가하기 위하여 비 $\rho(e\mathrm{BWT}(T))/\rho(T)$로 정의된 척도를 고찰하자. 정리 1로부터 제기되는 문제는 $\rho(e\mathrm{BWT}(T))/\rho(T)>1$인 런든단어들의 다중모임이 얼마나 많은가 하는 것이다. 이 문제를 해결하기 위하여 몇가지 사실을 보조정리로 준다.

보조정리 1 T가 런든단어(단일한 문자가 아닌)들의 다중모임이고 k를 정의 옹근수라고 하자. 그러면

$$\frac{\rho(e\mathrm{BWT}(T^k))}{\rho(T^k)}=\frac{1}{k}\frac{\rho(e\mathrm{BWT}(T))}{\rho(T)}$$

가 성립한다.

따름 런든단어(단일한 문자가 아닌)들의 다중모임 T와 정의 옹근수 k에 대하여 $\rho(e\mathrm{BWT}(T^k))/\rho(T^k)=2$이면 $\rho(e\mathrm{BWT}(T))/\rho(T)=2$이다.

$c\in\Sigma$를 제외한 모든 문자는 고정시키고 c는 c^k로 넘기는 준동형넘기기를 문자 c의 r차확장이라고 부르고 $\theta_{r,c}$로 표시한다. 즉 $b\ne c$이면 $\theta_{r,c}(b)=b$이고 $\theta_{r,c}(c)=c^r$이다.

보조정리 2 T를 Σ우에서의 런든단어들의 다중모임이라고 하자. 그러면 매 $c\in\Sigma$와 정의 옹근수 r에 대하여

$$\frac{\rho(e\mathrm{BWT}(\theta_{r,c}(T)))}{\rho(\theta_{r,c}(T))}\ge\frac{\rho(e\mathrm{BWT}(T))}{\rho(T)}$$

가 성립한다.

따름 T를

$$\frac{\rho(e\mathrm{BWT}(T))}{\rho(T)}=2$$

인 Σ 우에서의 런든단어들의 다중모임이라고 하자. 그러면 임의의 $c \in \Sigma$ 와 정의 웅근수 r 에 대하여

$$\frac{\rho(e\mathrm{BWT}(\theta_{r,c}(T)))}{\rho(\theta_{r,c}(T))} = 2$$

이다.

다음의 정리는 $\rho(e\mathrm{BWT}(T))/\rho(T) = 2$ 인 런든단어들의 다중모임이 무한히 많다는것을 보여준다.

정리 2 임의의 자모 Σ 와 임의의 $k \geq \max\{|\Sigma|, 3\}$ 에 대하여 $\rho(T) = 2k-2$ 이고 $\rho(e\mathrm{BWT}(T)) = 4k-4$ 인 Σ 우에서 런든단어들의 다중모임이 존재한다.

참 고 문 헌

[1] M. Burrows et al.; A Block Sorting Data Compression Algorithm, DIGITAL System Research Center, 68～174, 1994.

[2] S. Mantaci et al.; Theoretical Computer Science, **317**, 3, 298, 2007.

[3] S. Mantaci et al.; Theoretical Computer Science, **698**, 79, 2017.

주체109(2020)년 6월 5일 원고접수

Measuring Clustering Effect of the Extended Burrows-wheeler Transform via Run-length Encoding

Kim Chol Un, *An Kum Song*

In this paper we make study of the cases where the clustering effect of the extended BWT is not achieved. We show that the application of extended BWT to any word at the worst doubles the number of equal-letter runs.

Keywords: extended Burrows-Wheeler transform, Lyndon word