2021학년도 봄학기 연구계획

문자열 매칭 색인 개발 및 그래프 임베딩

김 성 환
sunghwan@pusan.ac.kr

연구 주제

- 문자열 매칭을 위한 자료구조 개발
- 그래프 임베딩 및 유사도 모형 개발

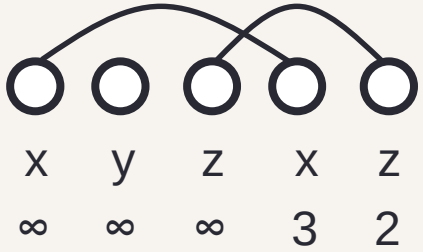
문자열 매칭을 위한 자료구조 개발 (1)

- Parameterized String Matching

  xyzxz = yzxyx, xyz ≠ xyy

  T = xyzzxyzxyxxzyyxyzxxxyyzxzyxxy, P = xyzz

- Encoding:
  - E(xyzxz) = ∞ ∞ ∞ 3 2



- Indexing Encoded Suffixes
  - E(T[16:])=E(yzxxxyyzxzyxxy)=∞∞∞∞11516424313
  - E(P) = E(xyzz) = ∞∞∞∞1

문자열 매칭을 위한 자료구조 개발 (1)'

- Parameterized String Matching

  xyzxz = yzxyx, xyz ≠ xyy

  T = xyzzxyzxyxxzyyxyzxxxyyzxzyxxy, P = xyzz

  - 기존연구(SODA'17): nlgσ + O(n) bits
  - 발표(IPL'21): (1+1) n lgσ + 2n + o(n) bits
  - 추후연구방향
    - 2n bits (RMQ)는 제거 가능
    - F/L array 대응 관계를 이용하면 +1항을 줄일 수 있을 것으로 보임
    - (1+1)을 (1+O(1/lg σ))로 줄여야 n lg σ +O(n) bits 달성 가능
    - 우선 Cartesian Tree Matching 문제에서 줄여본 후 적용시키는 방향으로...

문자열 매칭을 위한 자료구조 개발 (2)

- Order-preserving Matching

  2 3 1 5 = 5 8 3 9

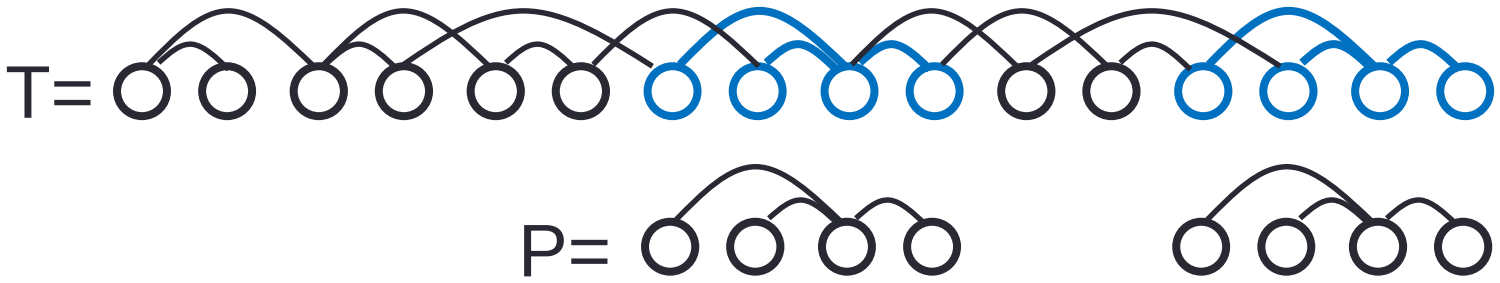
  rank 2 3 1 4 2 3 1 4

  T = 1 5 4 7 6 2 3 8 9, P = 2 1 3

  - 기존
    - Suffix Tree 기반: O(n lgn) bits, O(m) query time
    - 패턴 길이 제한($m < \lg^c n$): O(n lglgn) bits, O(m) query time

    - Open question: n lgn + O(n) bits ?

문자열 매칭을 위한 자료구조 개발 (3)



- Linear-Structured Pattern Matching
  - Extension from ISAAC'20
  - (1+1) e lg L + O(n+e) bits
    - n: #nodes, e: #arcs, L: max length of arc
  - Special Cases:
    - Parameterized String Matching
    - Cartesian Tree Matching
    - Isodirectional Pointer Sequences
    - Length-restricted order-preserving matching (△)

문자열 매칭을 위한 자료구조 개발 (4)

- 다른 문제들 찾기
  - Standard Matching:
    연속된 문자열이 동일
  - Cartesian Tree Matching:
    문자열로부터 만들어지는 Cartesian Tree가 동일

  - 주어진 문자열로부터 유일하게 유도되는 선형 이산 구조에 따른 매칭 문제

그래프 임베딩 및 유사도 모형 개발 (1)

- $f(G)=v \in R^k$

  $sim(G_1,G_2) \approx sim(f(G_1),f(G_2))$

1. 임베딩 방법
   - graph2vec: Weisfeiler-Lehman algorithm
   - Anonymous Walk

2. 그래프 유사도 $sim(G_1,G_2)=?$

그래프 임베딩 및 유사도 모형 개발 (2)

BWT(banana$)

i	BWT	suffixes
1	a	$
2	n	a$
3	n	ana$
4	b	anana$
5	$	banana$
6	a	na$
7	a	nana$

BWT(anaba$)

i	BWT	suffixes
1	a	$
2	b	a$
3	n	aba$
4	$	anaba$
5	a	ba$
6	a	naba$

BWT(banana$$_1$anaba$$_2$)

i	DA	BWT	suffixes
1	1	a	$$_1$
2	2	a	$$_2$
3	1	n	a$$_1$
4	2	b	a$$_2$
5	2	n	aba$$_2$
6	1	n	ana$$_1$
7	2	$$_1$	anaba$$_2$
8	1	b	anana$$_1$
9	2	a	ba$$_2$
10	1	$$_2$	banana$$_1$
11	1	a	na$$_1$
12	2	a	naba$$_2$
13	1	a	nana$$_1$

출처: “Computing Burrows-Wheeler Similarity Distributions for String Collections”

- Graph로의 응용:
  - 만약 Random walk가 그래프의 특성을 나타낸다면
    → 두 그래프의 Random walk에 대한 유사도를 위와 유사한 방법으로 계산
  - Geometric Graph: 노드의 좌표나 방향을 고려해서 Walk를 표현할 수도?