Searching with Extended Guard and Pivot Loop

Waltteri Pakalén
Jorma Tarhio
Bruce W. Watson

Outline

- Exact string matching:
  Find the occurrences of $P = p_0 \ldots p_{m-1}$ in $T = t_0 \ldots t_{n-1}$

- Tune-up technique: Guard test with q-gram
  - Experimental results
- Tune-up of SSM algorithm
  - Pivot loop
  - Variations
  - Experimental results

Guard Test with q-gram

Horspool/SMART

$s \leftarrow 0$;

while $s \leq n-m$ **do**

    $i \leftarrow 0$

    while $i < m$ **and** $p_i = t_{s+i}$ **do** $i \leftarrow i+1$

    if $i = m$ **then** report

    $s \leftarrow s + shift[t_{s+m-1}]$

Horspool with Guard Test

prefix $\leftarrow p_0$

$s \leftarrow 0$;

while $s \leq n-m$ **do**

    if prefix = t_s **then**

        $i \leftarrow 1$

        while $i < m$ **and** $p_i = t_{s+i}$ **do** $i \leftarrow i+1$

        if $i = m$ **then** report

    $s \leftarrow s + shift[t_{s+m-1}]$

Guard Test

- A. Hume and D. Sunday: *Fast string searching*.
  Software: Practice and Experience 1991

- T. Raita: *Tuning the Boyer-Moore-Horspool string searching algorithm*.
  Software: Practice and Experience 1992

Guard Test with q-gram, q = 2, 4, 8

prefix $\leftarrow p_0 \ldots p_{q-1}$

$s \leftarrow 0$

while $s \leq n-m$ **do**

  if prefix = $t_s \ldots t_{s+q-1}$ **then**

    $i \leftarrow q$

    while $i < m$ **and** $p_i = t_{s+i}$ **do** $i \leftarrow i+1$

    if $i = m$ **then** report

  $s \leftarrow s + \text{shift}[t_{s+m-1}]$

Checking q-grams as Entities

- J. Tarhio and B. W. Watson: *Tune-up for the Dead-Zone algorithm*. PSC 2020

- S. Faro and M. O. Külekci: *Fast packed string matching for short patterns.* ALENEX 2013
- A. Sharfuddin and X. Feng: *Improving Boyer-Moore-Horspool using machine-words for comparison*. ACM 2010
- M. A. Khan: *A transformation for optimizing string-matching algorithms for long patterns*. The Computer Journal 2016

Speed-ups, English, q = 8, m = 8

BF	3.11
SMITH	1.31
NSN	1.29
HOR	1.26
TSW	1.25
TS	1.15
RAITA	1.14
BR	1.12
GRASPM	1.03
ASKIP	1.02

Speed of BF with q = 4, 8

- BF4 is faster than other comparison-based algorithms for m = 4 on English and DNA
- BF8 is faster than other comparison-based algorithms for m = 8 and 16 on English and DNA
- On English, the rank of BF8 for m=8 is #16 among all algorithms of SMART
- BF8 is faster than any algorithm of SMART for m = 8 on rand2

Pivot Loop and SSM

SSM, Pivot loop

- The SSM algorithm[1] (Simple String Matching) utilizes a special skip loop where a pivot, a selected position of the pattern, is tested at each alignment of the pattern.
- In case of failure, the Horspool shift is applied.
- We call this kind of skip loop a pivot loop.

[1] A. M. Al-Ssulami: Hybrid string matching algorithm with a pivot. Journal of Information Science 2015

Model of Exact String Matching

```
1  s ← 0;
2  while s ≤ n−m do
3     while condition do s ← s + shift1
4     match loop
5     s ← s + shift2
```

SSM

- condition: $p_c \neq t_{s+c}$ where c is the pivot position
- shift1: Horspool
- match loop: backward
- shift2: proprietary
- a stopper is used

Selection of Pivot

- Compute the distance of each p_i to its next occurrence to the left in the pattern or to the left end.
- Select p_i with the largest distance.
- Example: aaacbaaaa

SSM shift2 vs. BM good suffix

- Shift of the pattern after character mismatch in the match loop

a	a	g	t	a	a	a	a	**c**	a	a	a	
9	9	9	9	9	9	9	9	**9**	**9**	**9**	**9**	SSM shift2
10	10	10	10	10	10	10	10	4	1	2	3	good suffix

a	a	a	a	**b**	a	a	a	a	
5	5	5	5	5	**5**	**5**	**5**	**5**	SSM shift2
5	5	5	5	5	1	2	3	4	good suffix

a	**b**	a	b	a	b	a	b	
2	2	2	2	2	2	2	**2**	SSM shift2
2	2	4	4	6	6	8	1	good suffix

Variations

2 **while** s ≤ n−m **do**
3 **while** condition **do** s ← s + shift1
4 backword match loop
5 s ← s + shift2

- Pivot (8 alternatives)
  - least frequent character
  - q-gram
- Shift1 (4 alternatives)
  - Sunday
  - Berry-Ravindran
- Shift2 (2 alternatives)
  - original + good suffix

Experimental Results (speed-up)

	English			DNA		
	5	10	20	5	10	20
SSM-ASC	1.13	1.06	1.03	1.04	0.99	0.98
SSM-AFSC	1.31	1.21	1.11	0.99	0.83	0.72
SSM-ASGC	1.33	1.28	1.19	1.00	0.84	0.73
SSM-USC	1.51	1.28	1.07	1.96	1.46	1.17
SSM-UBC	1.82	1.70	1.72	2.61	2.56	2.71
SSM-UXC	1.68	1.70	1.82	2.90	2.96	3.22
SSM-VBC	-	1.76	1.82	-	2.61	2.71
SSM-VXC	-	1.70	1.82	-	3.09	3.22
SSM-MB	1.41	1.38	1.41	1.24	1.27	1.45
SSM-WB	1.82	1.76	1.72	2.61	2.61	2.71
SSM-WX	1.75	1.76	1.94	3.03	3.09	3.32
SSM-WBZ	2.28	1.89	1.35	-	-	-

- U: 4-gram pivot $p_{m-4}...p_{m-1}$
- X: shift1 modified Berry-Ravindran
- C: shift2: SSM shift2 + good suffix

X X

Conclusions

- Guard test with 8-gram made BF competitive.
- The SSM algorithm offers a platfom for experimentation.
- Tune-up of an algorithm may outcome a significant speed-up.