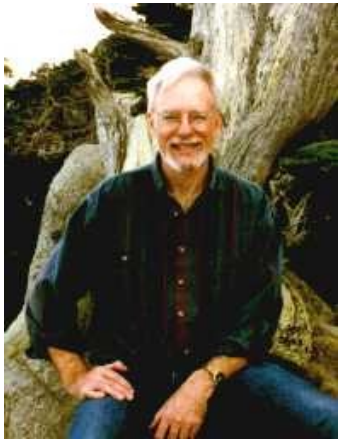
LMU, CIS,
SoSe 2006
Veranstaltung: Programmierung von Stringmatchingalgorithmen in C
Kursleiter: Max Hadersbeck

Boyer–Moore–Algorithmus

Referenten:
Galina Hinova
Stefan Partusch
Andreas Neumann

Boyer–Moore–Algorithmus
Die Erfinder



J Strother Moore

J. Strother Moore hat den Admiral B. R. Inman Centennial Chair der Computing Theory an der University of Texas in Austin inne. Darüber hinaus steht er auch dem Department vor.

Entwickelte den Suchalgorithmus 1977 in Zusammenarbeit mit Bob Boyer.

Gewann 2005 mit Boyer und Kaufmann den ACM Software System Award für den Boyer-Moore-Theorem-Beweiser.

Robert Stephen Boyer

Professor für Computer Wissenschaften, Philosophie und Mathematik an der University of Texas in Austin.

Entwickelte den Suchalgorithmus 1977 in Zusammenarbeit mit J. Strother Moore.

Ein weiteres gemeinsames Projekt ist der automatisierte Theorem-Beweiser Nqthm den sie in Zusammenarbeit 1992 veröffentlicht haben.

Boyer–Moore–Algorithmus
Idee

- Der Text wird von links nach rechts durchlaufen.
- Der Vergleich zwischen Text und Suchmuster jedoch wird von rechts nach links durchgeführt.
- Gibt es keinen Treffer wird das Suchmuster nach rechts verschoben.

Problemstellung:

Um wieviele Stellen kann man das Muster verschieben?

Das hängt ab von:

- ... dem aktuellen Zeichen
- ... dem Vorkommen des Zeichens im Muster

Boyer–Moore–Algorithmus

Die Idee an einem Beispiel

```
0 1 2 3 4 5 6 7 8 9 10 ...
a b b a d a b a c b a
b a b a c
          b a b a c
```

1. Das Muster ist fünf Zeichen lang.
2. Man vergleicht von rechts nach links, das letzte Musterzeichen mit dem Textzeichen an fünfter Stelle.

Hier also „c“ mit „d“ =>  kein Treffer

2. Der Buchstabe „d“ kommt im Suchmuster nicht vor.
3. Also kann das Muster um fünf Positionen nach rechts verschoben werden.

Boyer–Moore–Algorithmus
Fallunterscheidung

I. „Schlechtes Zeichen“-Strategie (Bad Character Heuristics)

Das letzte Zeichen des Musters stimmt nicht überein.

1. Das Textzeichen kommt im Muster nicht vor

=> das Muster wird um die Musterlänge nach rechts verschoben

2. das Textzeichen kommt im Muster vor

=> das Muster kann soweit verschoben werden, bis das letzte Vorkommen des Zeichens im Muster auf das Textzeichen ausgerichtet ist:

```
0 1 2 3 4 5 6 7 8 9 10 ...
a b b a b a b a c b a
b a b a c
    b a b a c
```

Boyer–Moore–Algorithmus
Fallunterscheidung

II.“Gutes Ende“-Strategie (Good Suffix Strategie)

Das letzte Zeichen stimmt überein.

Die „Schlechtes Zeichen“-Strategie erweist sich in diesem Fall als nicht richtig, da sie zu einer negativen Verschiebung führen kann:

```
    0 1 2 3 4 5 6 7 8 9 10 ...
    a b a a b a b a c b a
    c a b a b
  c a b a b
        c a b a b
```

Naive Lösung: Verschiebung um eine Stelle nach vorne

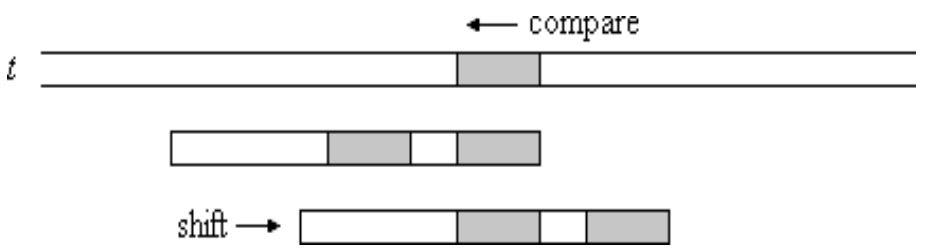
Kluge Lösung: Die größtmögliche Schiebedistanz aus der Struktur des Musters ablesen

Boyer–Moore–Algorithmus

Fallunterscheidung - „Gutes Ende“-Strategie

1. Fall:

Das übereinstimmende Suffix kommt an anderer Stelle im Muster vor



- Wie bei Knuth-Morris-Pratt wird ein Rand berechnet.
- Wichtig ist dabei, dass das Muster sich nicht nach links fortsetzen lässt.

Boyer–Moore–Algorithmus

Fallunterscheidung - „Gutes Ende“-Strategie

```
Vorverarbeitung
i: 0 1 2 3 4 5 6
p: a b b a b a b
f: 5 6 4 5 6 7 7 8
s: 5 5 5 5 2 5 4 1
occ: a b
     5 6
```

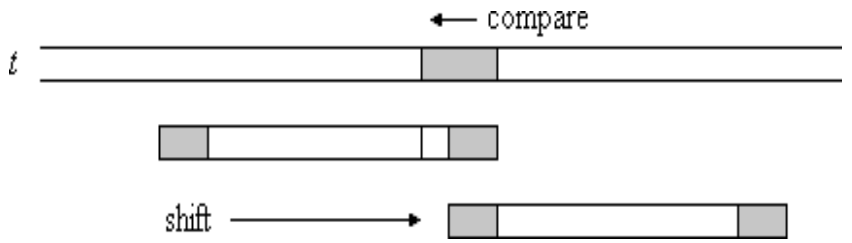
Das Suffix „bab“ hat als breitesten Rand „bab“ . Somit ist die Maximale Schiebedistanz wenn bei p[4] ein Mismatch auftritt 2, bei p[6] aber 4.

Boyer–Moore–Algorithmus

Fallunterscheidung - „Gutes Ende“-Strategie

2. Fall:

Das übereinstimmende Suffix kommt teilweise am Anfang des Musters vor.



- Am Anfang der Sprungtabelle ist der breiteste Rand vermerkt.
- Dieser wird überprüft. Falls er nicht zutrifft kann das Suchfenster um die Suffixlänge verschoben werden

Boyer–Moore–Algorithmus

Fallunterscheidung - „Gutes Ende“-Strategie

```
Vorverarbeitung
i: 0 1 2 3 4 5 6
p: a b b a b a b
f: 5 6 4 5 6 7 7 8
s: 5 5 5 5 2 5 4 1
occ: a b
     5 6
```

Tritt bei p[0-3] ein Mismatch auf kann das Suchfenster um 5 Positionen verschoben werden.
p[5] kann der Anfang eines neuen Treffens des Suffixes „abba“ sein und somit bei einem Nicht-Treffer um 5 Positionen verschoben werden.

Boyer–Moore–Algorithmus

Schritte

I. Vorverarbeitung

1. eine Tabelle für die „Schlechtes Zeichen“-Strategie, die die mögliche Schiebedistanz für alle Zeichen des Alphabets berechnet

2. Zwei Tabellen für die „Gutes Ende“-Strategie

a. eine Tabelle für die Berechnung der Positionen der Ränder für die Suffixe

b. eine Tabelle für die Berechnung der Schiebedistanz

II. Suchphase

Boyer–Moore–Algorithmus
Suchphase

- Der Text wird von links nach rechts durchsucht
- Das Muster wird von rechts nach links durchsucht
- Sobald ein Mismatch beim Vergleich auftritt, wird die Sprungdistanz für jede der beiden Strategien bestimmt
- Die größere Distanz wird verwendet und das Muster entsprechend verschoben
- Nach einem Treffer wird um die Musterlänge verschoben