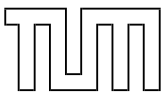
Algorithmische Bioinformatik 1

Dr. Hanjo Täubig

Lehrstuhl für Effiziente Algorithmen
(Prof. Dr. Ernst W. Mayr)
Institut für Informatik
Technische Universität München

Sommersemester 2009



Übersicht

Ausgangssituation direkt nach Mismatch

- Der erste Mismatch soll im zu durchsuchenden Text t an der Stelle $i + j$ (also im Suchwort s an Stelle j) auftreten.
- Da der Boyer-Moore-Algorithmus das Suchwort von hinten nach vorne vergleicht, ergibt sich folgende Voraussetzung:

$$s_{j+1} \cdots s_{m-1} = t_{i+j+1} \cdots t_{i+m-1} \qquad \wedge \qquad s_j \neq t_{i+j}$$

- in Worten: Das Suffix des Suchworts ab Index $j + 1$ stimmt mit den entsprechenden Zeichen im Text überein, das Zeichen an Position j jedoch nicht.

Werte der Shift-Tabelle: kleine Shifts

- Um nun einen sinnvollen Shift um σ Positionen zu erhalten, muss gelten:

$$s_{j+1-\sigma} \cdots s_{m-1-\sigma} = t_{i+j+1} \cdots t_{i+m-1} = s_{j+1} \cdots s_{m-1} \qquad \wedge$$

$$s_j \neq s_{j-\sigma}$$

- Diese Bedingung ist nur für „kleine“ Shifts mit $\sigma \leq j$ sinnvoll, also für solche, bei denen der Anfang des Suchworts s noch (mindestens) den gesamten bisher gematchten Bereich (inklusive das Mismatch-Zeichen) vom Text t abdeckt.

- Beispiel: erster Shift in der vorigen Abbildung

Werte der Shift-Tabelle: große Shifts

- Für „große“ Shifts $\sigma > j$ muss gelten, dass das Suffix des übereinstimmenden Bereichs mit dem Präfix des Suchwortes übereinstimmt, d.h.:

$$s_0 \cdots s_{m-1-\sigma} = t_{i+\sigma} \cdots t_{i+m-1} = s_\sigma \cdots s_{m-1}$$

- Zusammengefasst ergibt sich für beide Bedingungen:

$$\sigma \leq j \quad \wedge \quad s_{j+1} \cdots s_{m-1} \in \mathcal{R}(s_{j+1-\sigma} \cdots s_{m-1}) \quad \wedge \quad s_j \neq s_{j-\sigma}$$
$$\sigma > j \quad \wedge \quad s_0 \cdots s_{m-1-\sigma} \in \mathcal{R}(s_0 \cdots s_{m-1})$$

$\mathcal{R}(s)$: Menge aller Ränder von s

Zulässige und sichere Shifts

- Erfüllt ein Shift nun eine dieser Bedingungen, so nennt man diesen Shift zulässig.
- Um einen sicheren Shift zu erhalten, wählt man das minimale σ, das eine der Bedingungen erfüllt.
- Somit gilt:

$$S[j] = \min \left\{ \sigma : \begin{array}{c} (s_{j+1} \cdots s_{m-1} \in \mathcal{R}(s_{j+1-\sigma} \cdots s_{m-1}) \wedge \\ s_j \neq s_{j-\sigma} \wedge \sigma \leq j) \vee \\ (s_0 \cdots s_{m-1-\sigma} \in \mathcal{R}(s_0 \cdots s_{m-1}) \wedge \sigma > j) \vee \\ (\sigma = m) \end{array} \right\}$$

Bestimmung der Shift-Tabelle

Algorithmus 8 : compute_shift_table(int $S[]$, char $s[]$, int m)

```
for (int j := 0; j ≤ m; j++) do
    S[j] := m ;                                  // Initialisierung von S[]
// Teil 1: σ ≤ j
int border2[m + 1];
border2[0] := −1;
int i := border2[1] := 0;
for (int j' := 2; j' ≤ m; j'++) do
    // Hier gilt: i = border2[j' − 1]
    while ((i ≥ 0) && (s[m − i − 1] ≠ s[m − j'])) do
        int σ := j' − i − 1;
        S[m − i − 1] := min(S[m − i − 1], σ);
        i := border2[i];
    i++;
    border2[j'] := i;
```

Bestimmung der Shift-Tabelle

Algorithmus 9 : compute_shift_table(int $S[]$, char $s[]$, int m)

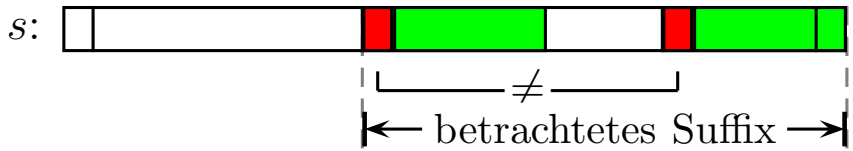
```
   ⋮

// Teil 2: σ > j
int j := 0;
for (int i := border2[m]; i ≥ 0; i := border2[i]) do
    int σ := m − i;
    while (j < σ) do
        S[j] := min(S[j], σ);
        j++;
```

Bestimmung der Shift-Tabelle

- Zu Beginn wird die Shift-Tabelle an allen Stellen mit der Länge des Suchstrings initialisiert.
- Im Wesentlichen entsprechen beide Fälle von möglichen Shifts (siehe obige Vorüberlegungen) der Bestimmung von Rändern von Teilwörtern des gesuchten Wortes.
- Dennoch unterscheiden sich die hier betrachteten Fälle vom KMP-Algorithmus, da hier, zusätzlich zu den Rändern von Präfixen des Suchwortes, auch die Ränder von Suffixen des Suchwortes gesucht sind.

Bestimmung der Shift-Tabelle: $\sigma \leq j$

- Dies gilt besonders für den ersten Fall ($\sigma \leq j$). Hier soll das Zeichen unmittelbar vor dem Rand des Suffixes ungleich dem Zeichen unmittelbar vor dem gesamten Suffix sein:



- Diese Situation entspricht dem Fall in der Berechnung eines eigentlichen Randes, bei dem der vorhandene Rand nicht zu einem längeren Rand fortgesetzt werden kann (nur werden hier Suffixe statt Präfixe betrachtet).
- Daher sieht die erste for-Schleife genau so aus, wie in der Prozedur compute_borders des KMP-Algorithmus.
- Lässt sich ein betrachteter Rand nicht verlängern, so wird die while-Schleife ausgeführt.

Bestimmung der Shift-Tabelle: $\sigma > j$

- Zweiter Fall ($\sigma > j$): man muss alle Ränder von s durchlaufen.
- Der längste Rand ungleich s ist der eigentliche Rand von s. Diesen erhalten wir über $border2[m]$, da ja das Suffix der Länge m von s gerade wieder s ist.
- Der nächstkürzere Rand von s muss ein Rand des eigentlichen Randes von s sein. Damit es der nächstkürzere Rand ist, muss s der eigentliche Rand des eigentlichen Randes von s sein, also das Suffix der Länge $border2[border2[s]]$.
- Somit können wir alle Ränder von s durchlaufen, indem wir immer vom aktuell betrachteten Rand der Länge ℓ das Suffix der Länge $border2[\ell]$ wählen. Dies geschieht in der for-Schleife im zweiten Teil. Solange der Shift σ größer als die Position j eines Mismatches ist, wird die Shift-Tabelle aktualisiert. Dies geschieht in der inneren while-Schleife.

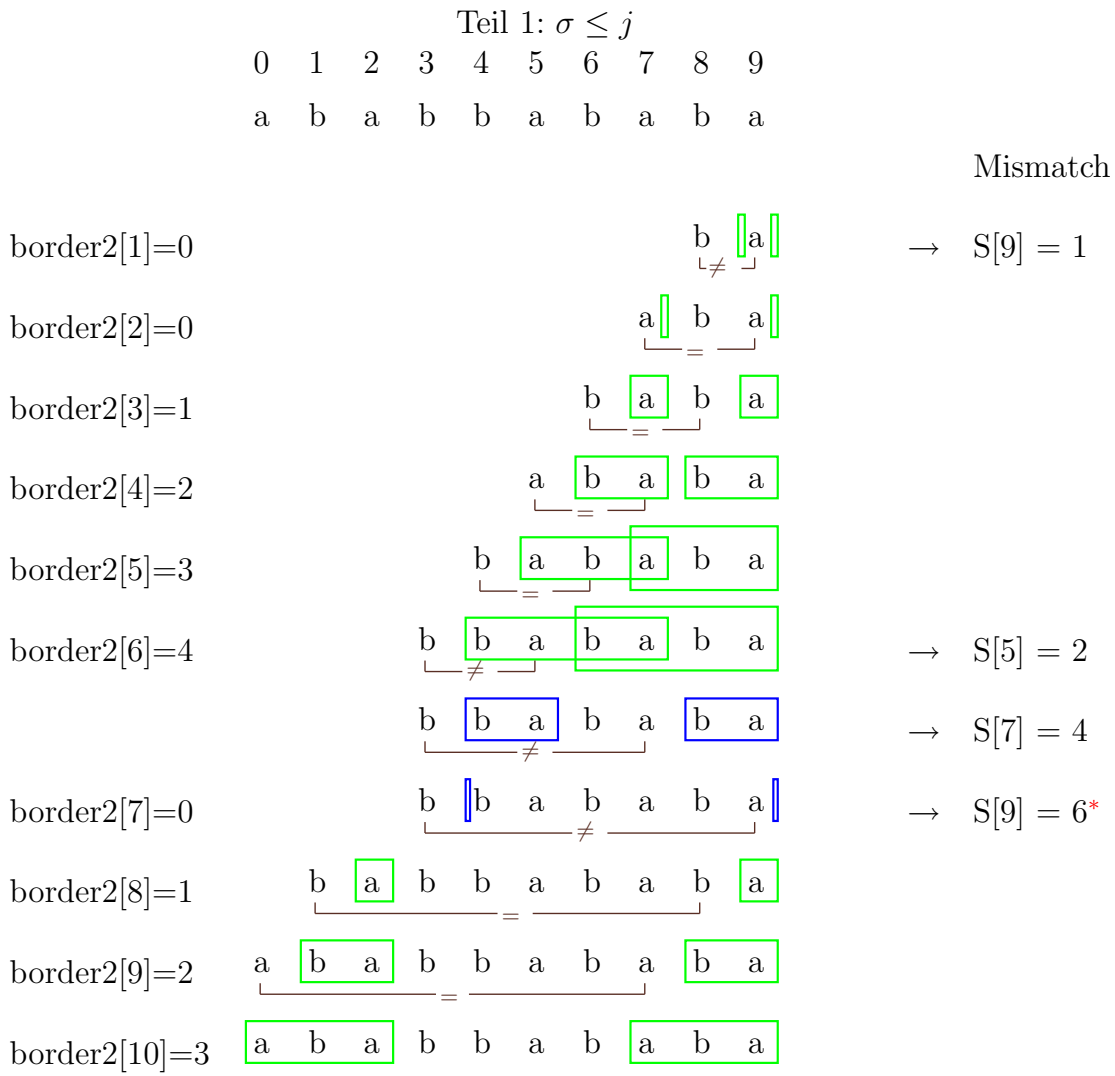
Bestimmung der Shift-Tabelle

- Bei der Aktualisierung der Shift-Tabelle werden nur zulässige Werte berücksichtigt. Damit sind die Einträge der Shift-Tabelle (d.h. die Länge der Shifts) nie zu klein.

- Eigentlich müsste jetzt noch gezeigt werden, dass die Werte auch nicht zu groß sind, d.h. dass es keine Situation geben kann, in der ein kleinerer Shift möglich wäre.

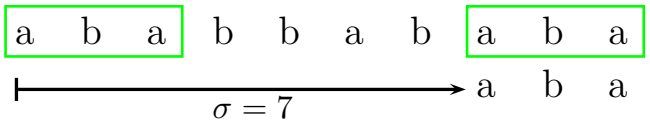
- Dass dies nicht der Fall ist, kann mit einem Widerspruchsbeweis gezeigt werden (man nehme dazu an, bei einem Mismatch an einer Position j in s gäbe es einen kürzeren Shift gemäß der Strong-Good-Suffix-Rule und leite daraus einen Widerspruch her).

Bestimmung der Shift-Tabelle: Beispiel

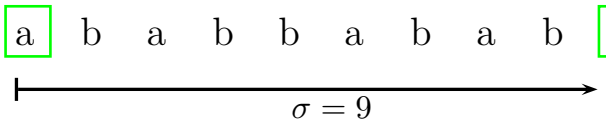


Bestimmung der Shift-Tabelle: Beispiel

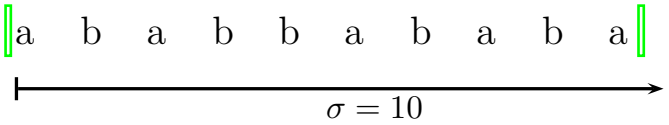
Teil 2: $\sigma > j$



bei Mismatch an Position $j \in [0:6]$ S[j] = 7



bei Mismatch an Position $j \in [7:8]$ S[j] = 9



bei Mismatch an Position $j \in [9]$ S[j] = 10

Zusammenfassung:

	1.Teil	2.Teil	Erg		1.Teil	2.Teil	Erg
S[0] =		7	7	S[5] =	2		2
S[1] =		7	7	S[6] =		7	7
S[2] =		7	7	S[7] =	4		4
S[3] =		7	7	S[8] =		9	9
S[4] =		7	7	S[9] =	1		1

Laufzeit von *compute_shift_table*

- Die Prozedur *compute_shift_table* entspricht im ersten Teil der Prozedur *compute_borders* des KMP-Algorithmus
- Anzahl (Zeichen-)Vergleiche ist also wieder $\leq 2m - 1$

- Der zweite Teil kann höchstens m-mal durchlaufen werden.
- Dabei werden gar keine (Zeichen-)Vergleiche durchgeführt.

Lemma

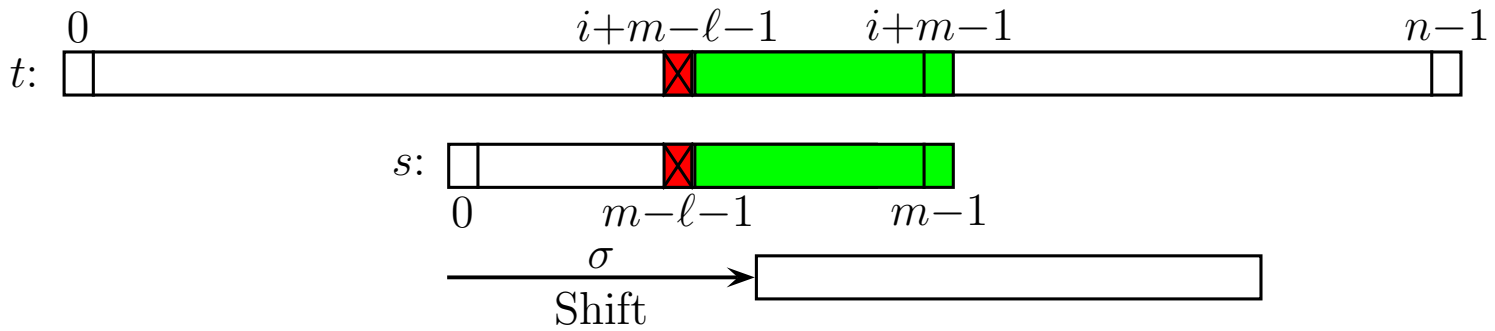
Die Shift-Tabelle des Boyer-Moore-Algorithmus lässt sich für eine Zeichenkette der Länge m mit maximal 2m Vergleichen berechnen.

Laufzeit der Hauptprozedur

- Betrachte nur die Suche nach ersten Vorkommen des Suchworts s in t (bzw. eine erfolglose Suche)
- Unterscheide initiale und wiederholte Vergleiche

- Initiale Vergleiche:
  Vergleiche von s_j mit t_{i+j}, so dass t_{i+j} zum ersten Mal beteiligt ist
- Wiederholte Vergleiche:
  Vergleiche von s_j mit t_{i+j}, so dass t_{i+j} schon früher bei einem Vergleich beteiligt war

Laufzeit der Hauptprozedur



Skizze: Shift nach ℓ erfolgreichen Vergleichen

Laufzeit der Hauptprozedur

Lemma

Für eine Textposition $i \in [0 : n - m]$ und eine Anzahl $\ell \in [0 : m - 1]$ gelte

- $s_{m-\ell} \cdots s_{m-1} = t_{i+m-\ell} \cdots t_{i+m-1}$, *sowie*
- $s_{m-\ell-1} \neq t_{i+m-\ell-1}$.

(Es gab also ℓ erfolgreiche und einen erfolglosen Vergleich.)

- *Dabei seien l initiale Vergleiche durchgeführt worden.*
- *Sei σ die Länge des folgenden Shifts gemäß der Strong-Good-Suffix-Rule.*

Dann gilt:

$$\ell + 1 \leq l + 4 \cdot \sigma$$

Lemma: kurze und lange Shifts

Unterscheidung des darauf folgenden Shifts in Abhängigkeit seiner Länge im Verhältnis zu ℓ:

- $\sigma \geq \lceil \ell/4 \rceil$: langer Shift,
- $\sigma < \lceil \ell/4 \rceil$: kurzer Shift.

Lemma: Fall 1 (langer Shift)

Shift σ ist lang, d.h. $\sigma \geq \lceil \ell/4 \rceil$:

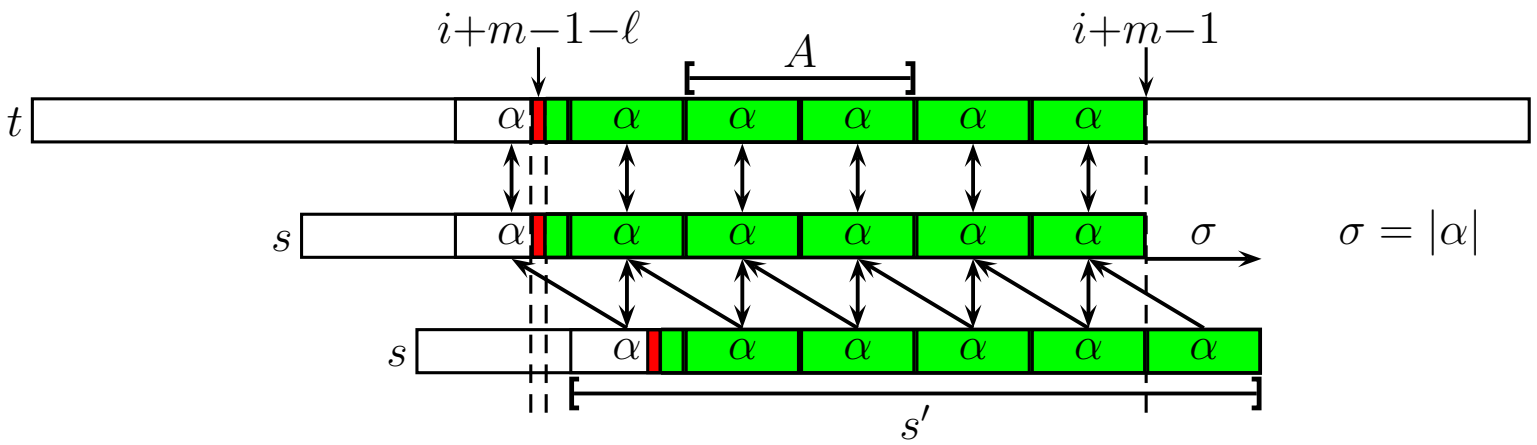
$$\begin{aligned} \text{Anzahl der ausgeführten Vergleiche} &= \ell + 1 \\ &\leq 1 + 4 \cdot \lceil \ell/4 \rceil \\ &\leq 1 + 4 \cdot \sigma \\ &\leq l + 4 \cdot \sigma \end{aligned}$$

Letzte Ungleichung: bei jedem Versuch muss es immer mindestens einen initialen Vergleich geben (zu Beginn und nach einem Shift wird das letzte Zeichen von s mit einem Zeichen aus t verglichen, das vorher noch an keinem Vergleich beteiligt war).

Lemma: Fall 2 (kurzer Shift)

Shift σ ist kurz, d.h. $\sigma \leq \lceil \ell/4 \rceil - 1 \leq \ell/4$:

Wir zeigen zuerst, dass dann das Ende von s periodisch sein muss, d.h. es gibt ein $\alpha \in \Sigma^+$, so dass s mit α^5 enden muss.



Lemma: Fall 2 (kurzer Shift)

- Im Intervall $[i+m-\ell : i+m-1]$ in t stimmen die Zeichen mit der verschobenen Zeichenreihe s überein (Good-Suffix-Rule)
- Das Suffix α von s der Länge σ muss mindestens $\lfloor \ell/\sigma \rfloor$ mal am Ende der Übereinstimmung von s und t vorkommen.
- α muss also mindestens $k := 1 + \lfloor \ell/\sigma \rfloor \geq 5$ mal am Ende von s auftreten (siehe Abbildung).
- Nur falls das Wort s kürzer als $k \cdot \sigma$ ist, ist s ein Suffix von α^k.
- Sei s' das Suffix der Länge $k \cdot \sigma$ von s, falls $|s| \geq k \cdot \sigma$ ist, und $s' = s$ sonst.
- Im Suffix s' wiederholen sich die Zeichen im Abstand von σ Positionen.