Protokoll 1 von 10.07.17

Ivana Daskalovska

Thema: *Multiple Stringsuche:Verfahren von Aho-Corasick*
Studentin: *Elena Atanasova*

Die Suche nach Mustern in Texten ist in vielen Bereichen der Informatik ein sehr wichtiges Thema. Das einfachste Verfahren zur Suche in einem Text wäre die zu suchende Zeichenkette immer um eine Position von links nach rechts zu verschieben und dann einen Vergleich zu machen. Die Laufzeit, welche sich aus diesem naiven Verfahren ergibt wäre O(n*m) bei (n-m+1) zu durchsuchenden Stellen im Text. (m- Musterlänge, n-Textlänge)

Um die Laufzeit zu verbessern wurde der Knuth-Morris-Pratt-Algorithmus entwickelt. Im Gegensatz zu diesem naiven Verfahren „merkt" sich der KMP-Algorithmus die bereits gelesenen Buchstaben. Dank dieser Speicherung sucht man nach einer fehlenden Übereinstimmung nicht wieder von vorne, sondern ab einer bestimmten gespeicherten Stelle. Damit wird die Laufzeit auf O(n+m) reduziert.

Sowohl der naive Algorithmus, als auch der KMP-Algorithmus suchen nur nach einem Schlüsselwort. Dies kann bei einer großen Anzahl an zu suchenden Schlüsselwörter zu Laufzeitproblemen führen. Soll nach k Schlüsselwörtern gesucht werden, so ergibt sich eine Laufzeit von O(m+k*n), da der Text für jedes Schlüsselwort wieder von vorne durchlaufen werden muss.

Um das Suchen mehrerer Schlüsselwörter gleichzeitig bei einem Suchdurchgang zu ermöglichen, entwickelten Alfred V. Aho und Margeret J. Corasick den Aho-Corasick-Algorithmus. Dieser Algorithmus ist eine Mischung aus dem Knuth-Morris-Pratt Algorithmus mit endlichen Automaten. Was man dazu braucht ist ein Buchstaben-Baum für Patternmenge P. Der Algorithmus konstruiert einen deterministischen endlichen Automaten **(Q, Σ, g, f, out, q0).**
Q bezeichnet eine endliche Menge von Zuständen, **Σ** das Eingabealphabet, **g** die Übergangsfuntion, **f** die Fehlerfunktion und **out** die Ausgabefunktion. Der Startzustand wird hier mit **q0** bezeichnet. Die drei Funktionen sind wichtig für die Konstruktion des Automaten.

Die Übergangsfunktion g ist eine Darstellung der Buchstaben der Schlüsselwörter in einem (Buchstaben-) Baum. Jeder Baum besitzt eine Wurzel mit dem Startzustand q0. Die Kanten des Baumes sind Zeichen aus dem Alphabet. Von jedem Zustand geht höchstens eine ausgehende Kante pro Symbol. Jeder Weg von der Wurzel zu einem Blatt entspricht einem Muster P. (P = {he, his, she}) Bei der Konstruktion des Baums startet man bei der Wurzel und fügt die Pattern nacheinander hinzu. Um ein Pattern hinzufügen folgt man dem Weg entlang der Zeichen des Patterns. Falls eine Kante mit dem ersten Symbol des einzufugenden Schlüsselwortes existiert, wird diese verwendet um in den nächsten bestehenden Knoten zu wechseln. Falls keine solche Kante existiert erfolgt eine Verzweigung. Man speichert ein Identifier i am Endknoten des Weges des Patterns.
Wenn man ein Patern P(i) im Baum suchtl, startet man bei der Wurzel und folgt dem Weg der Zeichen von Pi so lange wie möglich. Sobald man am Ende angekommen ist, und der Zustand im Buchstabenbaum ein Finalzustand ist, so ist P(i) in K enthalten. Andersrum ist P(i) in K nicht enthalten.

Fehler-Links werden durch eine Fehler-Funktion gekennzeichnet. Ein Fehler-Link zeigt von einem Zustandsknoten v auf einen Knoten w im Buchstaben-Baum. Fehler-Links mit der Länge 1 gehen zur Wurzel. Bei der Berechnung der Fehlerfunktion eines Knotens wird die Fehlerfunktion des Vorgängers in die Übergangsfunktion g(q, s) des aktuellen Zustandes eingesetzt (s- Das Symbol, das

von Vorgänger zum aktuellen Zustand führt, q-Fehlerfunktion des Vorgängers). Daher muss die Reihenfolge der Knoten bei der Berechnung unbedingt beachtet werden.

Zu jedem Zustand wird eine Menge von Schlüsselwörtern durch die Ausgabefunktion angegeben, die bis zu diesem Zustand gefunden wurden.

Die gesamte Laufzeit der Algorithmus ist O(m+n+k) dabei ist **m** die Gesamtzahl der Zeichen aller Schlüsselwörter, **n** die Länge des Textes und **k** die Anzahl der vorkommenden Schlüsselwörter im Text. Der Aho-Corasick-Algorithmus hat im Vergleich zu KMP-Algorithmus eine um das 5- bis 10-fache bessere Suchlaufzeit. Der Algorithmus wird heutzutage in vielen Bereichen verwendet: Bei dem Unix Befehl fgrep, in der Bildverarbeitung beim Bildvergleich, bei der Erkennung von Virenmuster, die sich in den Netzwerken befinden, in der Bioinformatik, bei dem Suchen in DNA-Sequenzen bzw. generell genommen in vielen Bereichen der Medizin, weil dort die Datenmengen sehr groß sind.