Zusammenfassung zum Vortrag von Elena Atanasova über das Thema „Multiple Stringsuche: Verfahren von Aho-Corasick“

Zu Beginn des Vortrages geht Elena auf die Motivation zu diesem Verfahren ein. Das Thema beschäftigt sich mit Pattern Matching, also der Mustererkennung bzw. der Suche von Schlüsselwörtern. Das Pattern Matching wird in sehr vielen Bereichen der Informatik immer wichtiger. Bei dem simplesten Verfahren von der Suche eines Musters der Länge n in einem Text der Länge m beträgt die Laufzeit O(m * n). Der Grund hierfür ist, dass bei einem fehlschlagendem Match, das ganze Muster von vorne durchgegangen werden muss, an der nächsten Stelle im Text.

Die Laufzeit dieses Verfahrens, also der Suche von nur einem Muster in einem Text, kann durch Anwendung des Knurt-Morris-Pratt-Algorithmus auf eine Laufzeit-Komplexität von O(m + n) verbessert werden. Die Idee dieses Algorithmus ist, dass Informationen über bereits gelesene Stellen in einer Sprungtabelle gespeichert werden. Dadurch ist es nicht notwendig, beim Fehlschlagen eines Matches nochmal komplett von vorne zu Beginnen. Es werden also erneute Vergleiche vermieden.

Falls man nun mehrere Schlüsselwörter in einem Text suchen möchte, z.B. k Schlüsselwörter, so muss man für jedes Schlüsselwort eine Suche machen, was einer Laufzeit von O(m+k*n) entspricht. Um diese Laufzeit zu verbessern, wurde der Aho-Corasick-Algorithmus entwickelt. Dieser Algorithmus kombiniert die Idee des Knurt-Morris-Pratt-Algorithmus mit einem endlichen Automaten. Es wird ein sogenannter Keyword-Baum (engl. Trie) für eine Patternmenge erstellt. Die Patternmenge enthält hierbei die Schlüsselwörter, die in einem Text gesucht werden sollen.

Der Algorithmus konstruiert einen deterministischen endlichen Automaten, der aus folgenden Elementen besteht: einer endlichen Menge von Zuständen, einem endlichen Eingabealphabet, einer Übergangs- bzw. goto-Funktion, einer Fehlerfunktion, einer Ausgabefunktion und einem Startzustand.

Die Übergangsfunktion beschreibt, von welchem Zustand man durch das Lesen welches Buchstabens in welchen neuen Zustand kommt. Diese Funktion wird dadurch erstellt, dass man vom Startzustand ausgehend zunächst für jeden Anfangsbuchstaben der Schlüsselwörter einen Übergang zu einem neuen Zustand definiert. Dabei gibt es jeweils nur einen Übergang pro Buchstaben (also keine Mehrfach-Übergänge für denselben Buchstaben). Nun verfährt man von den neuen Zuständen ebenso: man definiert für jeden zweiten Buchstaben, allerdings nur der Schlüsselwörter, die den Anfangsbuchstaben haben, der zu diesem Zustand führt, einen Übergang zu einem neuen Zustand. Dies wird für alle Zustände wiederholt, bis keine weiteren Zustände mehr hinzukommen. Für jeden Endzustand (also für jeden Zustand, bei dem eines der Schlüsselwörter komplett gelesen wurde) wird ein Identifier gespeichert, mit dem man das zugehörige Schlüsselwort aus der Patternmenge erhält.

Die Fehlerfunktion definiert, in welchen Zustand man kommt, falls in einem Zustand kein Übergang für den nächsten Buchstaben definiert ist. Diese Funktion ist rekursiv und wird wie folgt aufgebaut: alle Knoten des Baumes der Tiefe 1 werden auf die Wurzel abgebildet. Für alle anderen Knoten geht man zunächst zu dem letzten Knoten. Dann geht man zu dem Knoten, auf den dieser Knoten durch die Fehlerfunktion abgebildet wird. Nun schaut man, ob in dem aktuellen Knoten ein Übergang für den aktuellen Buchstaben definiert ist, ansonsten geht man erneut zu dem Knoten, auf den der aktuelle durch die Fehlerfunktion abgebildet wird, usw.

Die Ausgabefunktion bildet jeden Zustand auf die Menge der Wörter, die in diesem Zustand bereits gelesen wurden, ab.

Durch dieses Verfahren erhält man eine Laufzeit-Komplexität von O(m + n + k), wobei m die Länge des Textes, n die Länge der Schlüsselwörter und k die Anzahl der Schlüsselwörter ausdrückt.

Der Aho-Corasick Algorithmus wird z.B. in der Bildverarbeitung, in der Bioinformatik zur Durchsuchung der menschlichen DNA oder in der Medizin verwendet.