

Algorithmen: Informatik und Alltag

Eine Seminararbeit von Anton Blazicevic & Daniel Edthofer

Zusammenfassung

Die Basis für die vorliegende Seminararbeit ist *Algorithmic adventures: from knowledge to magic* von Juraj Hromkovič. Die Arbeit gibt einen Einblick in das System der Algorithmen. Zu Beginn gibt es eine kurze Einführung in die Thematik sowie Geschichte. Anhand der Analogie eines Kochrezeptes werden die einzelnen Bestandteile und Eigenschaften erklärt und leicht verständlich demonstriert. Dabei werden alle Kriterien, die ein Algorithmus erfüllen muss aufgelistet. Durch den Vergleich mit einem Rezept ist es besonders einfach nachzuvollziehen wie ein Algorithmus entsteht, wie man ihn vereinfacht, wie man die einzelnen Befehle am besten auswählt und schließlich wie man ihn auf ein Programm anwendet, sodass er von einem Computer ausgeführt werden kann. Zu den einzelnen beschriebenen Szenarien gibt es jeweils ein Beispiel und am Ende befindet sich ein Literaturverzeichnis.

Verwandte Literatur

Durch die Wichtigkeit der Algorithmen in der Informatik, gibt es eine Fülle an Literatur und Internetquellen, die das Thema behandeln, wodurch sich natürlich auch mehrere Quelleninhalte überschneiden. Wie bereits erwähnt basiert diese Arbeit auf dem Konzept von *Algorithmic adventures: from knowledge to magic* von Juraj Hromkovič [LQ1]. Um mit der Geschichte des Algorithmus zu beginnen war die Hauptquelle *Grundlegende Algorithmen mit Java: vom Algorithmus zum fertigen Programm* von Doina Logofătu [LQ2] notwendig.

Hilfreich war weiters *Algorithmen für Ingenieure—realisiert mit Visual Basic: Eine anwendungsorientierte Einführung — Problemanalyse und Lösungsweg anhand konkreter Beispiele* von Harald Nahrstedt [LQ4]. Dieses Werk enthielt übersichtliche geliederte Erklärungen zum Konzept des Algorithmus was Einfluss auf Generelles hatte.

Das Werk *Grundkurs Algorithmen und Datenstrukturen in JAVA: Eine Einführung in die praktische Informatik* von Andreas Solymosi und Ulrich Grude [LQ5] enthält eine nötige, praxisorientierte Nähe, welche unter Generelles erkennbar ist.

Zuletzt enthielt *Fundamente der Informatik: Ablaufmodellierung, Algorithmen und Datenstrukturen* von Peter Hubwieser und Gerd Aiglstorfer [LQ3] eine Reihe von einzigartigen Herangehensweisen an den Stoff Algorithmus.

Einleitung

“Algorithmics, or What Have Programming and Baking in Common?” [LQ1]

Dieses Kapitel beschäftigt sich mit Algorithmen, deren Aufbau und wie man Vorgänge aus dem echten Leben in Algorithmen ausdrücken kann. Dazu ist es anfangs gar nicht notwendig eine gewisse Programmiersprache zu beherrschen. Für den Anfang kann man sich ganz simpler Methoden bedienen. Man kann zum Beispiel einen Algorithmus zu einem gewissen Grad mit einem ganz simplen Kochrezept vergleichen.

Doch zuerst mal ein paar grundlegende Dinge...

Geschichte

Das Wort Algorithmus stammt von dem im 8. Jahrhundert lebenden persischen Mathematiker Al Khwarizmi. Ein von ihm verfasstes Buch namens „Hisab al-jabr w'al-muqabala“ über lineare und quadratische Gleichungen ist der Ursprung der heutigen Algebra, welche ebenso wie das Wort Algorithmus durch Latinisierung seines Namens entstand.

Einer der wahrscheinlich ersten niedergeschriebenen Algorithmen, um ein mathematisches Problem zu lösen, stammt vom griechischen Mathematiker Euklid. Über seine Person ist nur sehr wenig bekannt bzw. umstritten, jedoch soll von ihm ein 13 bändiges Werk stammen, das das gesammelte mathematische Wissen der Antike beinhaltet.

Generelles

In der Informatik gibt es den Zweig der Algorithmik der sich mit folgenden Punkten befasst:

- Entwurf:
 - Beschreibung des Algorithmus in Pseudocode
 - Beweis der Korrektheit
- Analyse der Ressourcen wie Aufwand, Laufzeit, Speicherbedarf,...

Die drei Kriterien die ein Algorithmus laut Beschreibung erfüllen sollte sind:

- Er muss eine Menge von Regeln/Befehlen/Prozessen sein
- Er muss in endlicher Zeit ein Ergebnis liefern
- Er muss eine Sequenz von Schritten sein, die Eingaben in Resultate umwandelt.

Algorithmus, Konzepte und Anwendung/Beispiele

In dem Buch wird ein Algorithmus erklärt als eine einfache Beschreibung eines Vorgangs der uns zu dem gewünschten Ziel führt. Das heißt ein Algorithmus ist wie eine Schritt-für-Schritt-Anleitung. Da sieht man auch schon die Ähnlichkeit zum Kochrezept. Auch hier ist alles in einzelnen Schritten angegeben, und man arbeitet sich von Schritt zu Schritt. Jetzt gibt es aber trotzdem eine wesentliche Unterscheidung zwischen den beiden. Kann man beispielsweise bei einem Kochrezept gewisse Dinge voraussetzen, muss bei einem Algorithmus alles bis ins kleinste Detail ausformuliert sein, sodass jeder x-beliebige diesen Anweisungen, ohne Erfahrung usw., Folge leisten kann. So kann in einem Rezept zum Beispiel stehen Eiweiß schaumig schlagen, in einem Algorithmus muss genau definiert sein, was schaumig bedeutet.

Da er von alle Menschen gleichermaßen ausführbar sein soll, muss man, bevor man ihn deklariert, eine Liste mit Befehlen erstellen, die im Algorithmus vorkommen. Am Beispiel des Kochrezepts wäre das etwa:

- Ofen auf X Grad aufheizen
- Y Minuten lang backen
- Z Sekunden lang umrühren
- Uvm.

Diese Befehle sind von jedem durchführbar, unabhängig von seinem Können. Wie ist es jetzt, aber bei dem oben genannten schaumig schlagen? Woher weiß man wie lange man rühren muss bis die gewünschte Konsistenz erreicht ist? Dazu gibt es jetzt zusätzlich zu den Befehlen ein weiteres Element, und zwar Abfragen. Man baut also quasi eine Abfrage ein, etwa so:

1. Eiweiß 10 Sekunden lang schlagen
2. Abfrage: ist Eiweiß steif geschlagen?
 - a. Ja: mach weiter mit nächstem Schritt im Rezept
 - b. Nein: zurück zu Schritt 1

Die Abfragen müssen, ebenso wie die Befehle so formuliert sein, dass sie eindeutig von jedem durchgeführt werden können. Man muss also den Zustand „steif“ formalisieren: z.B. wenn man die Schüssel umdreht und der Schaum bleibt drin, dann ist er steif

genug geschlagen. Mit diesen beiden Elementen eines Algorithmus lassen sich bereits simple Programme erstellen.

Man erkennt durch die Analogie eines Kochrezepts, dass es viele verschiedene Algorithmen geben kann die zu demselben Ergebnis führen. So kann man beispielweise Änderungen vornehmen die den Ablauf nicht direkt beeinflussen.

Wie man sieht ist, kann man dieses Prinzip des Kochrezepts so etwa auf das Programmieren am Computer übertragen. Hier gibt es allerdings einen wesentlichen Unterschied. In der Küche gibt es ja nur eine bestimmte Menge an Zutaten die man zur Verfügung hat. Beim richtigen Programmieren geht man allerdings davon aus dass es unendlich viele input Möglichkeiten für eine spezifische Problemstellung gibt. Am Beispiel aus der Mathematik: $ax^2 + bx + c = 0$.

Bei folgender quadratischer Gleichung $x^2 - 5x + 6 = 0$ gibt es mehrere mögliche Lösungen, nämlich $x_1 = 2$, $x_2 = 3$. Da man die Koeffizienten a , b und c beliebig wählen kann hat man theoretisch unendlich viele Möglichkeiten. Dadurch ergibt sich eine weitere Eigenschaft des Algorithmus: Er muss für jeden möglichen Input korrekt funktionieren, d.h. ein richtiges Ergebnis erzeugen und zwar in endlicher Zeit.

Hierbei unterscheidet man nun zwischen einem sogenannten Programm und einem Algorithmus. Ein Programm muss nämlich nicht unbedingt eine Repräsentation eines Algorithmus sein. Es kann genauso gut eine zusammenhanglose Anordnung von beliebigen Befehlen sein. Genauso muss ein Algorithmus wiederum nicht in Form eines Programms sein. So kann man ihn z.B. auch als mathematische Funktion ansehen. Was wir unter Programmieren verstehen ist eigentlich das Formalisieren von Algorithmen in eine Programmiersprache.

Ein wichtiger Aspekt bei der Definition eines Algorithmus in einem Programm ist, dass dieser für jeden Input in endlicher Zeit ein Ergebnis liefert, sprich das Programm terminiert. Bei sogenannten Endlosschleifen kann es schnell zum Problem kommen. Man spricht von einer Endlosschleife, wenn die Endbedingung nicht erfüllt ist. Nimmt man nun wieder die Analogie des Kochens, sagen wir z.B. man möchte Wasser kochen, also auf 100°C aufheizen.

1. Wasser 20 Sekunden lang kochen
2. Abfrage: Hat Wasser 100°C ?
 - a. Ja: Fertig
 - b. Nein: zurück zu schritt 1

Wenn man anhand dieses Beispiel jetzt das Wasser kocht, und man überschreitet die 100°C Grenze, so wird die Endbedingung niemals erreicht und das Programm wird niemals terminieren. Das liegt ganz einfach an einem Formulierungsfehler. Man muss die Abfrage ändern zu: Hat das Wasser mindestens 100°C? So wird dieses Dilemma ganz einfach umgangen.

Konklusion

Das Thema „Algorithmen“ ist sehr umfangreich und weitläufig, doch durch die einfache Erklärung anhand eines simplen Kochrezepts ist es für jedermann sehr leicht diese Grundlage der Informatik nachzuvollziehen. Auch ein Programmieranfänger ist dadurch nicht überfordert und kann dem Inhalt sehr gut folgen. Doch auch wenn man sich schon ein bisschen mit dem Thema befasst hat, ist es dennoch interessant und bietet auch einige neue Aspekte. Durch die Beispiele wird das Ganze noch ein bisschen deutlicher veranschaulicht, allerdings bietet der Beitrag nur einen groben Überblick, wenn auch umfangreich. Wenn man sich noch weiter über das Thema informieren möchte, gibt es sehr viele vertiefende Quellen.

Literatur- und Quellenverzeichnis

[LQ1]: Juraj Hromkovič (2009). *Algorithmic adventures: from knowledge to magic*. Berlin: Springer.

[LQ2]: Doina Logofătu (2008). *Grundlegende Algorithmen mit Java: vom Algorithmus zum fertigen Programm; Lern- und Arbeitsbuch für Informatiker und Mathematiker*. Wiesbaden: Vieweg.

[LQ3]: Peter Hubwieser, Gerd Aiglstorfer (2004). *Fundamente der Informatik: Ablaufmodellierung, Algorithmen und Datenstrukturen*. München: Oldenbourg Verlagsgruppe

[LQ4]: Harald Nahrstedt (2006). *Algorithmen für Ingenieure—realisiert mit Visual Basic: Eine anwendungsorientierte Einführung — Problemanalyse und Lösungsweg anhand konkreter Beispiele*. Wiesbaden: Friedr. Vieweg & Sohn Verlag/GWV Fachverlage GmbH

[LQ5]: Andreas Solymosi, Ulrich Grude (2008). *Grundkurs Algorithmen und Datenstrukturen in JAVA: Eine Einführung in die praktische Informatik*. Wiesbaden: Vieweg+Teubner / GWV Fachverlage GmbH, Wiesbaden