XKCD zur Theoretische Informatik

Ergänzung zu den Kapiteln 4, 5 und 13

Dies ist ein Ergänzungsartikel zum Buch *Automaten und Sprachen: Theoretische Informatik für die Praxis*, von Andreas Müller. Erschienen im Verlag Springer-Vieweg, ISBN 978-3-662-70145-4 (Softcover) und ISBN 978-3-662-70146-1 (eBook).

Die Rechte an den Bilder gehören den in der Bildunterschrift angegebenen Bildquellen, wenn keine Quelle angegeben ist dem Autor. Die Teile des Autors werden unter der Lizenz CC BY-SA 4.0 zur Verfügung gestellt, Details: https://creativecommons.org/licenses/by-sa/4.0/.

Website zum Buch: https://autospr.ch

Springer-Link: https://link.springer.com/book/10.1007/978-3-662-70146-1

XKCD zur Theoretische Informatik

Andreas Müller

Zusammenfassung

Der XKCD Webcomic von Randall Munroe thematisiert oft auch Themen der theoretischen Informatik. Dieser Artikel beschreibt drei Beispiele von XKCD-Comics, die für das Buch relevant sind[1].

Seit 2005 publiziert Randall Munroe seinen Web-Comic XKCD [4]. Besonders oft werden Witze thematisiert, die nur mit mathematischem, programmiertechnischem oder wissenschaftlichem Inside-Wissen verstanden werden können.. Natürlich kommen dabei auch Themen der theoretischen Informatik zur Sprache. Im Folgenden werden drei für das Buch besonders hervorstechende Beispiel vorgestellt und erklärt.

1 Kapitel 4: Reguläre Ausdrücke

Reguläre Ausdrücke sind seit den siebziger Jahren fester Bestandteil vieler Unix-Programme. Besonders populär gemacht wurden sie aber durch die Skriptsprache Perl [5] von Larry Wall. Darauf spielt auch der XKCD#208 (Abbildung 1) an [3]. Im Comic wird ein kompliziertes Szenario beschrieben, in welchem regulären Ausdrücke eine Schlüsselrolle zur Lösung des Problems zukommt. Im Sinne der Einführung zum Comic, die erklärt, dass ein solches Szenario die Motivation erhöhen, sich eine neue Fertigkeit anzueignen, wird auch dem Leser des Buches empfohlen, sich zu den vermittelten theoretischen Kenntnissen und den dargelegten Anwendungsmöglichkeiten Situationen seiner eigenen Praxis auszudenken, in denen die Fähigkeiten von Nutzen sein würden.

2 Kapitel 5: Kontextfreie Grammatiken

Kontextfreie Grammatiken sind das Werkzeug der Wahl, um Programmiersprachen zu spezifizieren. In Kapitel 6 wird dies gezeigt an einer großen Anzahl von Beispielen gezeigt. Die automatische Konstruktion von Parsern direkt aus der Grammatik wird in Kapitel 7 am Beispiel des Parsergenerators Bison vorgeführt. Der XKCD#1090 (Abbildung 2) erlaubt sich ein Wortspiel dazu. [1] Im Comic stürmt der Protagonist auf die Bühne an einem Symposium über formale Sprachen, schreit das Wort “Grammatik” und verschwindet. Der Mouse-over-Text erklärt die Szene:

[1] Die Bewilligung zum Abdruck der Comics im Buch erreichte den Autor im letzten Moment nur zwei Tage nach der ursprünglichen Manuskriptabgabe. Wir freuen uns sehr, die Comics auch im Buch zeigen zu dürfen.

Abbildung 1: XKCD#208: Regular Expression (10. Januar 2007)

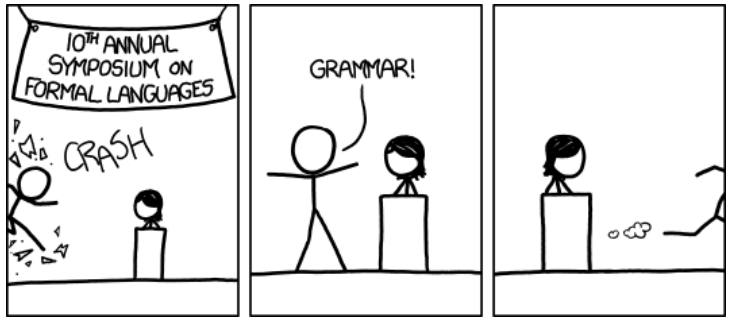

Abbildung 2: XKCD#1090: Formal Languages (3. August 2012)

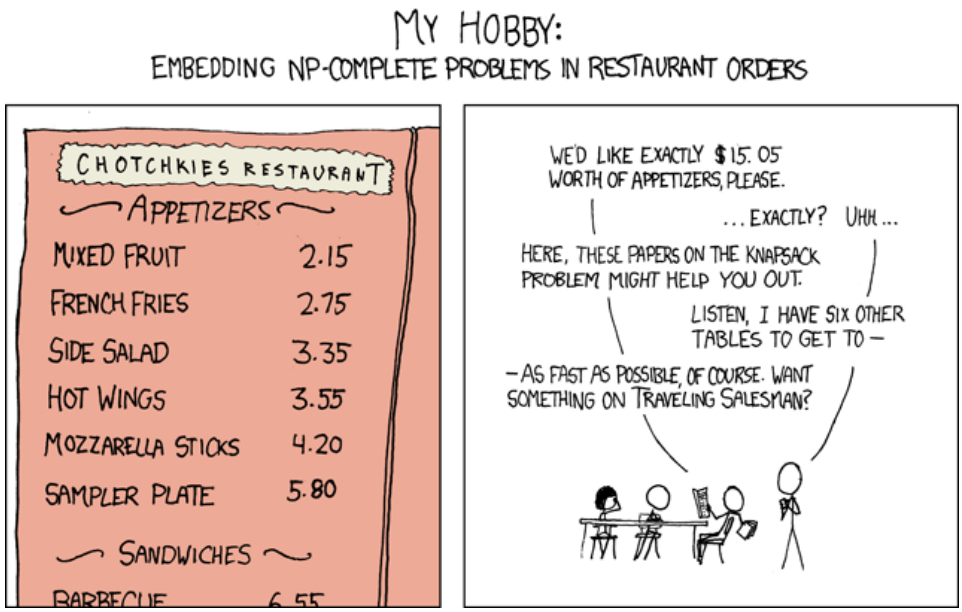

Abbildung 3: XKCD#287: NP-Complete (9. Juli 2007)

> [audience looks around] 'What just happened?' 'There must be some context we're missing.'

Das ratlose Publikum fragt sich, ob es vielleicht Kontext gibt, der das seltsame Verhalten und insbesondere den Schrei des Protagonisten erklären könnte. So wie ist im Comic dargestellt ist, ist ds Wort Grammatik ohne Kontext, "kontextfreie Grammatik".

Der Comic ist damit ein Wortspiel, allerdings eines, welches man nur mit soliden Kenntnissen der theoretischen Informatik verstehen kann.

3 Kapitel 13: NP-vollständig

Die Schwierigkeit, NP-vollständige Probleme effizient zu lösen, ist das Thema von XKCD#287 (Abbildung 3) [2], der am 9. Juli 2007 erschien. Die Speisekarte gibt eine Liste von Zahlen vor. Die Gäste, statt wie jede andere Kundschaft genau anzugeben, was sie konsumieren möchten, geben nur die Summe vor. Die Servierperson soll Vorspeisen zusammenstellen, die zusammen genau 15.05 USD kosten. Dies ist das Problem *SUBSET-SUM*, welches im Satz 13.16 mit Hilfe einer Reduktion $3SAT \leq_P SUBSET\text{-}SUM$.

Für das sehr kleine Problem lässt sich natürlich immer eine Lösung durch ausprobieren finden. Im vorliegenden Problem kann man sogar zwei Lösungen finden:

$$15.05 = 7 \cdot 2.15 = 1 \cdot 2.15 + 2 \cdot 3.55 + 1 \cdot 5.80.$$

Das im Comic gestellte Rucksackproblem ist also noch nicht ein schwer lösbares Problem. Wie bei allen NP-Problemen liegt die Schwierigkeit in der Skalierung für große Probleme.

Die protestierende Servierperson weist darauf hin, dass Sie in möglichst kurzer Zeit noch 6 weitere Tische besuchen soll, was natürlich eine Instanz des Travelling Salesman Problems ist. Darauf spielt die letzte Bemerkung des Bestellers an. Auch hier sind nur 6 Tische natürlich noch nicht ein wirklich schwierig lösbares Problem.

Literatur

[1] Randall Munroe. *Formal languages*. 2012. URL: https://xkcd.com/1090/ (besucht am 16. 10. 2024).

[2] Randall Munroe. *NP-Complete*. 2007. URL: https://xkcd.com/287/ (besucht am 16. 10. 2024).

[3] Randall Munroe. *Regular expressions*. 2007. URL: https://xkcd.com/208/ (besucht am 16. 10. 2024).

[4] Randall Munroe. *xkcd. A webcomic of romance, sarcasm, math, and language*. 2005. (Besucht am 16. 10. 2024).

[5] Perl. URL: https://www.perl.org (besucht am 16. 10. 2024).