



BOSCH
Technik fürs Leben



DHBW
Duale Hochschule
Baden-Württemberg

Sudoku Solver

Studienarbeit

Bachelor of Science

des Studiengangs Informatik

an der Dualen Hochschule Baden-Württemberg Stuttgart

von

Ruben Hartenstein, Annika Harter

10.06.2022

Bearbeitungszeitraum
Matrikelnummer, Kurs
Ausbildungsfirma
Betreuer
Gutachter

22.10.2022 - 10.06.2022
2746235, 4810277, TINF19ITA
Robert Bosch GmbH, Stuttgart
Sebastian Trost
Sebastian Trost

Duale Hochschule Baden-Württemberg, Stuttgart

Ausbildungsbereich Technik, Fachrichtung IT-Automotive

Bericht über die Ausbildung in der betrieblichen Ausbildungsstätte im 5. und 6. Studienhalbjahr.

Name des Studierenden: Ruben Hartenstein, Annika Harter

Studienjahrgang: 2019

Einsatz in Abteilung:

Standort: Stuttgart

Thema: Sudoku Solver

Betreuer: Sebastian Trost

vom: 22.10.2022 bis: 10.06.2022

Stellungnahme des Betreuers:

Dieser Bericht wurde geprüft und ist sachlich und fachlich richtig.

Ort

Datum

Abteilung, Unterschrift

Selbstständigkeitserklärung des Studenten

Gemäß §5(2) der „Studien- und Prüfungsordnung DHBW Technik“ vom 29. September 2015: Ich habe die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Ort

Datum

Unterschrift

Sperrvermerk

Die vorliegende Studienarbeit mit dem Titel

Sudoku Solver

enthält unternehmensinterne bzw. vertrauliche Informationen der Robert Bosch GmbH, ist deshalb mit einem Sperrvermerk versehen und wird ausschließlich zu Prüfungszwecken am Studiengang Informatik der Dualen Hochschule Baden-Württemberg Stuttgart vorgelegt.

Der Inhalt dieser Arbeit darf weder als Ganzes noch in Auszügen Personen außerhalb des Prüfungsprozesses und des Evaluationsverfahrens zugänglich gemacht werden, sofern keine anders lautende Genehmigung der Ausbildungsstätte (Robert Bosch GmbH) vorliegt.

Stuttgart, 10.06.2022

Ruben Hartenstein, Annika Harter

Selbstständigkeitserklärung

Ich versichere hiermit, dass ich meine Studienarbeit mit dem Thema: *Sudoku Solver* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Stuttgart, 10.06.2022

Ruben Hartenstein, Annika Harter

Abstract

TODO: deutscher Abstract....

Inhaltsverzeichnis

Abkürzungsverzeichnis	VII
Abbildungsverzeichnis	VIII
Tabellenverzeichnis	IX
Formelgrößenverzeichnis	X
Formelverzeichnis	XI
Listings	XII
1 Einleitung	1
1.1 Motivation - ToDo	1
1.2 Forschungsstand	1
1.3 Forschungsfrage	2
2 Rahmenbedingungen	3
2.1 Programmiersprache	3
2.2 Apache HTTP Server	3
2.3 Framework	3
2.3.1 Django	3
2.3.2 Flask	3
3 Die Mathematik hinter Sudoku	4
3.1 Mathematische Fragen	4
3.1.1 Abzählfragen	4
3.1.2 Komplexität	4
3.1.3 Strategie	4
3.2 Mathematisches Argumentieren beim Lösen eines Sudoku	4
3.2.1 Existenzbeweis	4
3.2.2 Eindeutigkeitsbeweis	4
3.3 Lösungshilfen: Kandidaten-Notation	4
4 Frontend	5

5 Backend	6
6 Implementierung Lösungsstrategien	7
Anhang	A

Abkürzungsverzeichnis

BSP Board Support Package

Abbildungsverzeichnis

Tabellenverzeichnis

Formelgrößenverzeichnis

a	rad	Bedeutung von a
b	rad	Bedeutung von b
λ	rad	Bedeutung von lambda
ϕ	rad	Bedeutung von phi

Formelverzeichnis

Listings

1 Einleitung

In vielen Zeitungen ist eine Vielzahl von Logikrätseln gedruckt. Darunter eines der häufigsten Logikrätseln: das Sudoku. In der üblichen Version ist es das Ziel, ein 9×9 -Gitter mit den Ziffern 1 bis 9 so zu füllen. Dabei kann das Gitter in drei unterschiedliche Einheiten aufgeteilt werden. Diese Einheiten sind die Spalten, Zeile und Blöcke des Gitters. Ein Block ist eine 3×3 -Unterquadrat des Gitters. In jeder Einheit darf jede Ziffer genau einmal vorkommen. Ausgangspunkt ist ein Gitter, in dem bereits mehrere Ziffern vorgegeben sind. Sudokus gibt es in unterschiedlichen Schwierigkeitsgraden. Dabei spielt nicht nur die Anzahl der vorgegeben Ziffern eine Rolle, sondern auch die Anordnung der Ziffern.

1.1 Motivation - ToDo

Wer schon einmal ein schweres Sudoku versucht hat zu lösen ist vielleicht auch schon an seine Grenzen zu kommen.

1.2 Forschungsstand

Im Internet findet man immer wieder Websites mit dem Angebot Sudokus zu lösen. Hierbei können die angegebenen Zahlen eingegeben werden und das Sudoku wird korrekt vervollständigt, falls es eine eindeutige Lösung gibt. Daraus kann der Nutzer jedoch keinen Nutzen ziehen. Des weiteren gibt es Websites mit Lösungsstrategien, die bei schwierigeren Sudokus angewendet werden müssen.

Das Beherrschen der Lösungsstrategien besitzt unter Sudokufreunden einen großen Stellenwert. Einfache Sudokus lassen sich meist noch intuitiv und durch konzentriertes Absuchen lösen. Die beiden dafür ausreichenden Anfängertechniken werden im Folgenden unter „Nackter Single“ und unter „Versteckter Single“ erläutert. Sie werden meist intuitiv und unbewusst eingesetzt, wenn für ein bestimmtes Feld nur eine Zahl möglich ist oder wenn

eine bestimmte Zahl nur in ein einziges Feld passt. Bei kniffligeren Sudokus müssen gewissenhaft Notizen gemacht werden. Wir verwenden die Methode der „Kandidaten“. Mit ihr lassen sich die Zusammenhänge am besten beschreiben. Manche Situationen in einem anspruchsvollen Sudoku erfordern sehr komplexe Lösungsansätze. Diese leicht verständlich zu erklären ist eine Kunst für sich.

1.3 Forschungsfrage

Für diese Studienarbeit ist die Zielsetzung ein Sudoku Solver zu entwickeln. Ziel ist es nicht ein Sudoku automatisch zu lösen, sondern dem Benutzer Schritt für Schritt den Weg zur Lösung aufzuzeigen. Auf dieser Basis wird zunächst geprüft ob das Sudoku eindeutig lösbar ist. Ist dies der Fall muss der Nutzer die Möglichkeit haben sich Anweisungen geben zu lassen wie man weiter Vorgehen könnte und welche Lösungsstrategie der Benutzer anwenden muss, damit man das Sudoku lösen kann. Das heißt, dass die Zahlen auf Basis der bereits vorhandenen Zahlen und Kandidaten mit ein wenig überlegen ermittelt und erklärt werden kann, anstatt einfach irgendeine Zahl zu zeigen, damit der Benutzer das Sudoku lösen kann. Die Lösungsstrategien sollen mittels des aktuellen Rätels erläutert und von der Software angewandt werden können. Diese Anweisungen bzw Hilfestellungen sind in vier Varianten unterteilt.

Für die erste Hilfestellung soll das Programm dem Benutzer die anzuwendende Strategie nennen und den Bereich hervorheben, auf dem die Strategie angewendet werden soll. Wenn diese Hilfestellung nicht ausreichend ist dann markiert das Programm die konkreten Zellen und Kandidaten, die von der Strategie betroffen sind. Mit der dritten Hilfestellung soll eine Erläuterung stattfinden warum bestimmte Kandidaten laut Strategie gestrichen werden können. Mit der letzten Hilfestellungen löscht das Programm den betroffenen oder mehrere betroffene Kandidaten.

Um ein lösbares Sudoku beenden zu können müssen etwas 25 Lösungsstrategien implementiert werden.

Das Programm soll auf einem Ubuntu Server mit Apache und Django laufen. Hier gibt es des weiteren noch die Anforderung, dass das Frontend responsiv sein soll und immer eine benutzeroptimierte Darstellungen liefert.

2 Rahmenbedingungen

2.1 Programmiersprache

2.2 Apache HTTP Server

2.3 Framework

2.3.1 Django

2.3.2 Flask

3 Die Mathematik hinter Sudoku

3.1 Mathematische Fragen

3.1.1 Abzählfragen

3.1.2 Komplexität

3.1.3 Strategie

3.2 Mathematisches Argumentieren beim Lösen eines Sudoku

3.2.1 Existenzbeweis

3.2.2 Eindeutigkeitsbeweis

3.3 Lösungshilfen: Kandidaten-Notation

4 Frontend

5 Backend

6 Implementierung Lösungsstrategien

Anhang