



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	1
1.2 Forschungsstand	2
1.3 Forschungsfrage	2
2 Rahmenbedingungen	4
2.1 Programmiersprache	4
2.2 Webserver	4
2.2.1 Apache HTTP Server	5
2.2.2 Nginx Webserver	6
2.2.3 Fazit	7
2.3 Framework	7
2.3.1 Django	8
2.3.2 Flask	8
2.3.3 Fazit	8
2.4 Entwurfsmuster	9
2.4.1 Model View Controller	10
3 Die Mathematik hinter Sudoku	11
3.1 Mathematische Fragen	11
3.1.1 Abzählfragen	11
3.1.2 Komplexität	12
3.1.3 Strategie	12

3.2	Mathematisches Argumentieren beim Lösen eines Sudoku	12
3.2.1	Existenzbeweis	12
3.2.2	Eindeutigkeitsbeweis	12
3.3	Lösungshilfen: Kandidaten-Notation	12
4	Frontend	13
5	Backend	14
6	Implementierung Lösungsstrategien	15
Anhang		A

Abkürzungsverzeichnis

BSD	Berkley Software Distribution
BSP	Board Support Package
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
HTTP	Hypertext Transport Protocol
HTTPS	Hypertext Transport Protocol Secure
MVC	Model View Controller

Abbildungsverzeichnis

2.1	Quelle: https://w3techs.com/technologies/overview/web_server	5
2.2	Model View Controller (MVC) Konzept	10

Tabellenverzeichnis

2.1	Tabelle	7
2.2	Tabelle	9

Formelgrößenverzeichnis

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

Formelverzeichnis

3.1 Beispielformel	11
------------------------------	----

Listings

1 Einleitung

In vielen Zeitungen ist eine Vielzahl von Logikrätseln gedruckt. Darunter eines der häufigsten Logikrätsel: das Sudoku. In der üblichen Version ist es das Ziel, ein 9×9 -Gitter mit den Ziffern 1 bis 9 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.

1.1 Motivation

Sudokus gibt es mittlerweile in vielen verschiedenen Varianten mit modifizierten Regeln. In dieser Arbeit und für den Sudoku Solver wird jedoch nur das klassische Sudoku, wie es in der Einleitung beschrieben wurde, betrachtet. Es gibt verschiedene Schwierigkeitsstufen von Sudokus. Meistens werden Sudokus mit vielen vorgegebenen Zahlen als einfach eingestuft und Sudokus mit wenigen vorgegebenen Zahlen als schwierig. Ein weiterer Punkt, der Einfluss auf die Schwierigkeitsstufe eines Sudokus hat, ist die Anordnung der vorgegebenen Ziffern.

Wer schon einmal versucht hat ein schweres Sudoku zu lösen, ist vielleicht auch bereits an seine Grenzen gekommen. Gerade bei schwierige Sudokus mit wenigen vorgebenden Zahlen kann es schwierig sein selbst eine passende Lösungsstrategie für das Sudoku zu finden. Oft sind die Lösungsstrategie schwierig zu verstehen und anzuwenden und es lassen sich damit nur Kandidaten eliminieren. Bei einfachen Sudokus werden diese Lösungsstrategien meist unbewusst und intuitiv eingesetzt. In dieser Studienarbeit soll ein Web Frontend entwickelt werden, auf dem ein Bearbeiter des Sudokus unterstützt wird, eine Lösung zu finden.

Das Ziel dieser Arbeit ist es diese schwierige Lösungsstrategie sichtbar zu machen und über mehrere Stufen von Hilfestellungen den Bearbeiter des Sudokus zu zeigen, wel-

che Lösungsstrategie im nächsten Schritt anzuwenden sind und an welcher Stelle des Sudokugitters.

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 Anwender jedoch keinen Nutzen ziehen. Des Weiteren gibt es Websites mit Lösungsstrategien, die bei schwierigeren Sudokus angewendet werden müssen.

Das Beherrschen der Lösungsstrategien besitzt einen großen Stellenwert. Einfache Sudokus lassen sich meist noch intuitiv und durch konzentriertes Absuchen lösen. Die beiden dafür ausreichenden Anfängertechniken heißen „Nackter Single“ und unter „Versteckter Single“ und werden im Verlauf dieser Arbeit noch erläutert. Sie werden mehrheitlich 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. Dafür wird die Methode der „Kandidaten“ verwendet. Damit lassen sich die Zusammenhänge am besten beschreiben. Manche Situationen in einem anspruchsvollen Sudoku erfordern sehr komplexe Lösungsansätze, die nicht immer so einfach zu verstehen sind.

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 eine korrekte 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, im folgenden auch Hilfestellungen genannt, 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 gezeigt werden, 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 etwa 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

Für das Projekt werden verschiedene Rahmenbedingungen festgelegt. Dazu gehört die Programmiersprache, indem der Sudoku Solver programmiert werden soll, ein Webserver auf dem das Programm später laufen wird und ein Framework, dass als Gerüst für das Programm auf dem Webserver fungiert. Wichtig ist, dass das Framework auf die Programmiersprache angepasst ist, um Probleme zu vermeiden. Im Zuge der Recherchen wurden verschiedenen Webserver in Betracht gezogen und besonders die Webserver Apache HTTP Server und Nginx Server aufgrund ihrer Beliebtheit miteinander verglichen. Auch bei den Frameworks wurden zwei besonders in Betracht gezogen und evaluiert, welches besser geeignet ist für die Studienarbeit.

2.1 Programmiersprache

TODO: Programmiersprache... Warum Python? Vorteile Python?

2.2 Webserver

Ein Webserver ist in der Regel ein Server, der zur Verbreitung von Webinhalten im Inter- oder Intranet dient. Die jeweiligen Informationen und Dokumentationen können demnach weltweit oder firmenintern erreicht werden. Damit eine Website jederzeit erreichbar sein kann, muss der Webserver permanent online sein. Der Rechner, auf dem der Webserver läuft, wird als Host bezeichnet. Der Webserver ist für die zuverlässige Übertragung von statischen, wie beispielsweise von unveränderlichen Hypertext Markup Language (HTML)-Dateien, aber auch von dynamischen Dateien verantwortlich. Für dynamische Dateien muss der Webserver vor der Antwort Programmcode ausführen. Dieser Programmcode wird in diesem Fall in der Programmiersprache Python geschrieben. Mittels einer Cascading Style Sheets (CSS) Datei können die Inhalte der dynamischen Dateien weitestgehend von den Darstellungsvorgaben getrennt werden. In der HTML Datei wird folglich nur

die inhaltliche Gliederung definiert und in der CSS Datei die Darstellung, wie etwa Farben oder Layout. Für die Übermittlung wird das Übertragungsprotokoll Hypertext Transport Protocol (HTTP) oder die verschlüsselte Variante Hypertext Transport Protocol Secure (HTTPS) verwendet. Ein Webserver ist in der Lage, die Inhalte auf viele verschiedene Rechner gleichzeitig zu übermitteln. Wie viele Nutzeranfragen (Requests) ein Server bearbeiten kann, hängt von der Hardware und der Auslastung des Hosts ab.

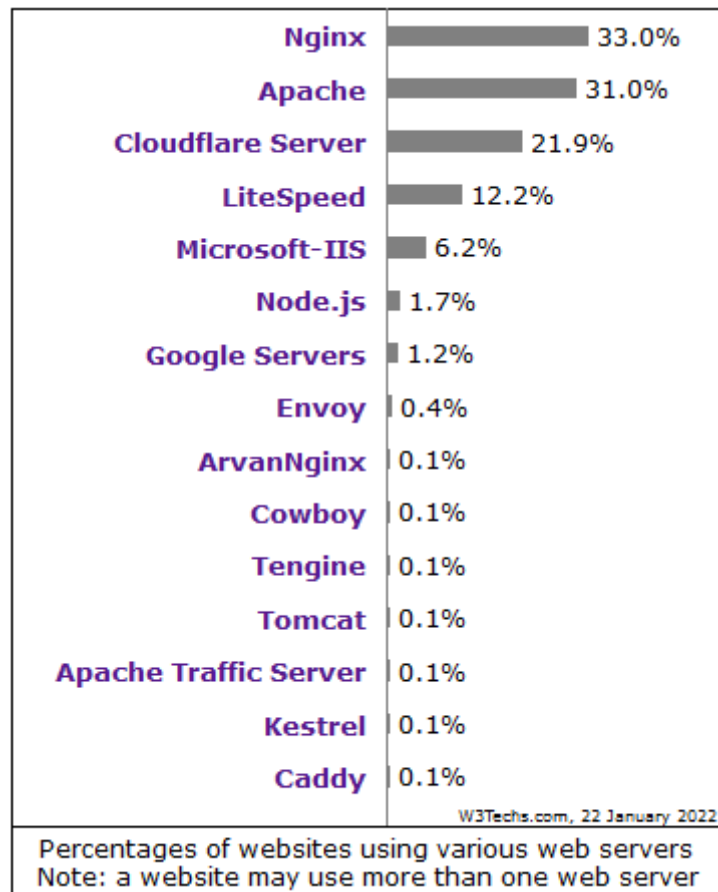


Abbildung 2.1: Quelle: https://w3techs.com/technologies/overview/web_server

Am verbreiteten sind unter anderem die Webserver Apache HTTP Server und nginx, wie in die Statistik 2.1 dargestellt. Beide Programme sind freie Software und wurden daher für das Projekt in Betracht gezogen. Im Folgenden werden diese beide Programme verglichen.

2.2.1 Apache HTTP Server

Der Apache Webserver wurde erstmals 1995 veröffentlicht und hat sich schnell zu einem der beliebtesten Webserver entwickelt. Er unterstützt neben Unix und Linux noch eine Vielzahl an weiteren Betriebssystemen.

Der Apache Server ist modular aufgebaut, wodurch benötigte Funktionen, die der Server nicht nativ bereitstellt, durch Module importiert werden können. Das Erstellen dynamischer Webseiten wird mittels serverseitiger Skriptsprachen bewerkstelligt. Über ein Modul kann der entsprechende Interpreter in den Server integriert werden.

Beim Apache-Webserver wird ein Ansatz verfolgt, bei dem jede Clientanfrage von einem separaten Prozess oder Thread bearbeitet wird. Dadurch werden Prozesse, die Schreib- oder Leseoperationen erfordern, nacheinander abgearbeitet und es kann passieren, dass ein Request in der Warteschlange verweilen muss, bis der vorherige Request durchgeführt werden konnte. Damit man dieses Problem umgehen kann, gibt es die Möglichkeit, mehrere Single-Threading-Prozesse gleichzeitig zu starten. Diese Strategie ist jedoch mit einem hohen Ressourcenaufwand verbunden. Um dies zu vermeiden, kommen Multi-Threading-Mechanismen zum Einsatz. Für die parallele Abfrage von Clientanfragen gibt es verschiedene Multi-Processing-Module, die integriert werden können.

2.2.2 Nginx Webserver

Der Marktanteil von Nginx ist in den letzten Jahren kontinuierlich gestiegen, weshalb auch dieser Webserver für die Studienarbeit in Betracht gezogen wurde. Nginx wurde erstmals 2004 veröffentlicht und ist wie der Apache Server auch mit diversen Betriebssystemen kompatibel.

Wie Apache ist auch Nginx modular aufgebaut und verschiedene Funktionen können über Module bereitgestellt werden. Zu diesen Modulen gehört jedoch nicht die Option, Interpreter für eine Programmiersprache entsprechend in den Webserver zu integrieren. Es wird also ein weiterer Anwendungsserver benötigt. Für kleine Webprojekte ist das ein Mehraufwand, der nicht immer eingegangen werden muss.

Dieser Server zeichnet sich durch eine hohe Performance aus. Dabei können eine möglichst große Anzahl an Clients gleichzeitig bedient werden und der Ressourcenverbrauch trotzdem gering gehalten. Durch die ereignisorientierte Architektur können Client-Anfragen asynchron bearbeitet werden, wodurch Arbeitsspeicher und Zeit gespart werden kann. Die Nebenläufigkeit ist realisiert, ohne dass für jede neue Verbindung ein zusätzlicher Prozess oder Thread benötigt wird. Die Stärke dieser Architektur zeigt sich bei großen Webprojekten.

2.2.3 Fazit

Die folgende Tabelle 2.1 zeigt die Unterschiede und Gemeinsamkeiten der beiden Webserver auf. Der ausschlaggebende Grund warum sich für einen Apache Server entschieden wurde, ist das Merkmal der dynamischen Webinhalte.

Merkmal	Apache	NGINX
Funktion	Webserver, Proxy-Server	Webserver, Proxy-Server, E-Mail-Proxy, Load-Balancer
Betriebssystem	Alle unixoiden Plattformen, Windows	FreeBSD, Linux, Solaris, IBM AIX, HP-UX, macOS, Windows
Lizenz	Apache License v2.0	BSD-Lizenz
Entwickler	Apache Software Foundation	Nginx, Inc.
Statische Webinhalte	Ja	Ja
Dynamische Webinhalte	Ja	Nein
Software-Architektur	Prozess-/threadbasiert	Eventgesteuert

Tabelle 2.1: Vergleich der Webserver Apache HTTP Server und Nginx Webserver

Der Apache Webserver bietet eine breite Möglichkeit an Modulen, um die Software zu erweitern. Der ausschlaggebende Grund, warum sich in diesem Projekt für den Apache HTTP Server entschieden wurde, ist die Möglichkeit Interpreter für Programmiersprachen über ein Modul direkt in den Webserver zu integrieren. Zudem muss für das Projekt nur ein Client bedient werden. Dadurch wird für das Anzeigen von dynamischen Inhalten kein separaten Anwendungsserver benötigt. Damit bietet der Apache HTTP Server eine bequemere Lösung für kleine Websites, deren Inhalt dynamisch erzeugt wird.

2.3 Framework

Ein Framework ist ein Programmiergerüst. Es wird insbesondere bei komponentenbasierten und bei der objektorientierten Softwareentwicklung verwendet. Ein Framework ist kein fertiges Programm, sondern stellt den Rahmen, in der eine Anwendung erstellt werden soll, zur Verfügung. Frameworks lassen sich in Typen gliedern, wobei es nicht immer eine strikte Trennung gibt. Zu diesen Typen gehören beispielsweise Application Frameworks oder Webframeworks.

Die beiden Frameworks, Django und Flask, die für das Projekt in Betracht gezogen wurden, gehören zu den beliebtesten Webframeworks für Python und sind für die Entwicklung von dynamischen Webanwendungen ausgelegt.

2.3.1 Django

Django ist ein kostenloses Open Source Webframework und kann plattformübergreifend genutzt werden. Es folgt dem MVC Entwurfsmuster, das auch in der Studienarbeit genutzt wird. Der Fokus von Django liegt in einer schnellen Entwicklung mit weniger Code. Mit dem Framework kommen viele bereits eingebaute Pakete, die genutzt werden können. Dazu gehört unter anderem die integrierte Anbindungen an verschiedenen Datenbanksysteme. Django hat verschiedene Mechanismen integriert, um die Sicherheit der Website zu garantieren. So können SQL Injektionen oder Cross-site Scripting ausgeschlossen werden.

Ein weiterer Vorteil von Django ist die große und aktive Community, wodurch viel Fragen schnell beantwortet werden können und es gibt es eine ausführliche Dokumentation zu dem Framework.

2.3.2 Flask

Webseiten die Flask als Framework nutzten sind meistens einseitige Anwendungen. Flask ist im Gegensatz zu Django modular und eher minimalistisch aufgebaut. Daher muss in der Entwicklung eher mit externen Bibliotheken gearbeitet werden. Man nennt Flask deshalb auch ein Mikroframework. Für die Sicherheit der Website gibt es die Flask-Security Bibliothek, die dieselben Mechanismen wie Django beinhaltet.

Aus diesen Gründen ist Flask eher für kleinere Projekte geeignet, die benutzerdefinierte Komponenten erfordern oder für das Prototyping. Die Anzahl der Webseiten, die Flask als Framework nutzen, steigt stetig weiter. Der Vorteil, warum Flask immer beliebter wird ist die Kontrolle, die man über das Projekt bekommt, indem die Komponenten selbst bestimmt.

2.3.3 Fazit

Beide Frameworks wären für das Projekt geeignet. Die Django Community ist größer und aktiver als die von Flask. Daher gibt es mehr Informationen über Django. Jedoch wird

Flask mittlerweile in mehreren Projekten genutzt. In der Tabelle 2.2 sind die wichtigsten Eigenschaften nochmals gegenübergestellt.

Merkmal	Django	Flask
Use cases	mittelgroße bis große Projekte	kleine bis mittelgroße Projekte
Community	große, aktive Community	kleinere Community
Packages	built-in Packages	modular
Performance	gut	gut

Tabelle 2.2: Vergleich der Frameworks Django und Flask

Aufgrund der Leichtigkeit wurde sich bei den Frameworks für Flask entschieden. Da es bei der Performance keine großen Unterschiede zwischen Django und Flask gibt, ist es als Merkmal für die Entscheidung eines Frameworks nicht relevant. Für den Sudoku Solver wird nur eine einseitige Anwendung ohne Datenbankbindung benötigt, wodurch Flask das geeignetere Framework ist. Auch der modulare Aufbau spricht für Flask, da nur wenige zusätzliche Pakete genutzt werden.

Das Entwurfsmuster MVC wird im Framework Flask übernommen.

2.4 Entwurfsmuster

Wie im Abschnitt der Frameworks schon erwähnt, wird in diesem Projekt eine bestimmte Softwarearchitektur verwendet. Damit wird einem Projekt die grundlegenden Komponenten und das Zusammenspiel dieser visualisiert. Damit ist es einfacher nicht-funktionale Eigenschaften wie die Modifizierbarkeit, Wartbarkeit oder Sicherheit darzustellen und zu verstehen.

Das Framework Django arbeitet immer mit dem MVC Architekturmuster. Da aber Flask als Framework genutzt wird, muss diese Architektur selbst implementiert werden. Im Folgenden wird das Architekturmuster mit seinen drei Komponenten kurz beschrieben (*und wie es in der Studienarbeit umgesetzt wurde*).

2.4.1 Model View Controller

Das Ziel dieser Entwurfsarchitektur ist es, spätere Änderungen oder Erweiterungen zu vereinfachen und die Wiederverwendbarkeit der einzelnen Komponenten zu steigern. In der Abbildung 2.2 ist die Architektur visualisiert. Durchgezogene Linien stehen hierbei für eine direkte Assoziation und gestrichelte für eine indirekte Assoziation.

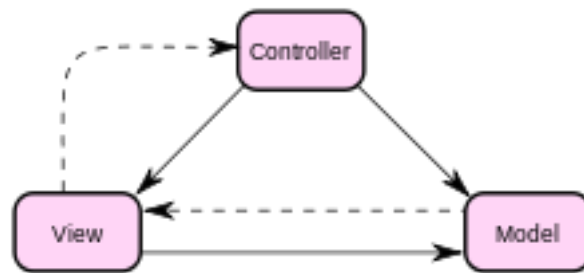


Abbildung 2.2: MVC Konzept

Model

Im Model sind die Daten enthalten, die dargestellt werden sollen. In Falle des Sudoku Solvers ist hier das Sudoku Board gespeichert.

View

In der View oder der Präsentation werden die Daten aus dem Model dargestellt. Außerdem wird in der View die Benutzerinteraktion geregelt. Die View kennt die Daten aus dem Model, soll diese aber nicht bearbeiten oder verändern und ist völlig unabhängig von dem Controller.

Controller

Im Controller werden das Model und die View verwaltet. Die View unterrichtet den Controller von Benutzerinteraktionen. Im Controller wird diese dann ausgewertet und gegebenenfalls das Model und die View angepasst. In dieser Komponente sind die Lösungsstrategien für den Sudoku Solver implementiert.

3 Die Mathematik hinter Sudoku

3.1 Mathematische Fragen

3.1.1 Abzählfragen

Anzahl der Lösungen und nicht Anzahl der Rätsel. Möglichkeit auf eine Lösung zu kommen mit unterschiedlichen Rätseln.

Um alle denkbaren, vollständig ausgefüllten 9×9 Standard-Sudokus zu erzeugen, könnte man wie folgt vorgehen: man beginnt mit einem leeren 9×9 -Gitter und setzt nun zeilenweise von links nach rechts die Ziffern ein. Für das erste Feld in der ersten Zeile hat man offenbar 9 Möglichkeiten, für das zweite 8, das dritte 7 usw. Insgesamt ergeben sich für die erste Zeile $9!$ (d. h. 9 Fakultät) Möglichkeiten. Wenn man in den verbleibenden 8 Zeilen ebenso vorgeht, erzeugt man mithin

$$a = b + \lambda - \frac{\phi - \lambda}{2 \cdot \pi} \quad (3.1)$$

verschiedene 9×9 -Gitter. Da allerdings unberücksichtigt blieb, dass jede Ziffer auch in jeder Spalte und in jedem Block nur genau einmal auftreten darf, hat man bei einem solchen Vorgehen (sehr) viele 9×9 -Gitter erzeugt, die keine vollständig ausgefüllten 9×9 Standard-Sudokus darstellen.

Bertram Felgenhauer und Frazer Jarvis konnten 2005 zeigen, dass es (nur) 6.670.903.752.021.072.936.960 (ca. 6,7 Trilliarden oder verschiedene (vollständig ausgefüllte) 9×9 Standard-Sudokus gibt.

Es gibt 5472730538 grundlegend verschiedene Sudoku-Raster und die Sudoku-Symmetriegruppe- Januar 2006 Ed Russel und Frazer Jarvis, ohne Vertauschung der Ziffern, ohne Drehungen, Zeilen- bzw. Spaltenvertauschungen und Blockzeilen- bzw. Blockspaltenvertauschungen) vollständig ausgefüllte, also gelöste Sudokus gibt.

$$9! = 362.880$$

3.1.2 Komplexität

3.1.3 Strategie

3.2 Mathematisches Argumentieren beim Lösen eines Sudoku

3.2.1 Existenzbeweis

Backtracking

3.2.2 Eindeutigkeitsbeweis

Wenn ein Sudoku-Rätsel nur ein einziges Feld vorgibt, so gibt es offenbar so viele verschiedene Lösungsmöglichkeiten (Vervollständigungen), wie es vollständig ausgefüllte Sudokus gibt, geteilt durch 9. Die in Medien veröffentlichten Sudoku-Rätsel werden mit der Maßgabe erstellt, eindeutig lösbar zu sein:

3.3 Lösungshilfen: Kandidaten-Notation

4 Frontend

5 Backend

6 Implementierung Lösungsstrategien

Anhang