Klassifikation der Schwierigkeitsgrade von Sudokus mit Methoden des maschinellen Lernens

Investigation of Sudoku difficulty levels

Bachelor-Thesis von Michael Bräunlein

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Informatik
Knowledge Engineering Group
Betreuer Prof. Dr. Johannes Fürnkranz

Zusammenfassung

Das Zahlenrätsel Sudoku ist weltweit bei Rätselliebhabern bekannt und beliebt. Seit seiner Veröffentlicheung 1986 begeistern sich immer mehr Menschen für mehr oder weniger schwierige Exemplare. Sudokus finden sich im Internet, in der Rätselecke der Tageszeitungen und sogar als ganze Bücher, um nur einige Erscheinungsorte zu nennen.
Die Regeln sind einfach zu lernen und doch kann man sich sehr lange mit Sudokus beschäftigen, da die schwersten Sudokus meißt nur von Profis gelöst werden können.
Der Spielspass ist sehr stark davon abhängig, dass die Schwierigkeit zum persönlichen Können passt. Ist das Sudoku zu leicht, stellt es keine Herausforderung dar. Ist es zu schwer, kommt schnell ein Gefühl der Überforderung auf.
Die ausgewiesenen Schwierigkeitsstufen von Sudokus aus verschiedenen Quellen haben zwar of die gleichen Namen wie zum Beispiel “Mittel”, unterscheiden sich aber dennoch häufig nach Meinung des Spielers.
Das Ziel dieser Bachelorarbeit ist, Merkmale aus Sudokus zu extrahieren, anhand derer die Sudokus von einem Klassifizierer möglichst zuverlässig in Schwierigkeitsstufen eingeteilt werden können.

Inhaltsverzeichnis

1 Aufgabenstellung und Zielsetzung

Diese Bachelorarbeit beschäftigt sich mit der Einteilung von Sudokus in verschiedene Schwierigkeitsstufen. Hierzu sollen Methoden des maschinellen Lernens verwendet werden.
Es soll eine Methode gefunden werden, mit der Merkmale aus Sudokus extrahiert werden können, die dann als Feature Vectoren in einer .arff Datei[1] gesammelt werden. Die Feature Vectoren werden anschließend mit Hilfe von Weka[2] klassifiziert.
Es werden verschiedene Klassifikatoren und unterschiedliche Parameter betrachtet. Ausserdem werden Optimierungen der Featurevektoren diskutiert.

[1] http://www.cs.waikato.ac.nz/ml/weka/arff.html
[2] http://www.cs.waikato.ac.nz/ml/weka/

2 Einführung

Die Vorfahren des heutigen Sudokus waren vermutlich die lateinischen Quadrate, mit denen sich vor allem der Mathematiker Leonhard Euler befasste. Hier ging es darum, in ein Quadrat mit n Zeilen und n Spalten Symbole so einzutragen, dass jedes Symbol in jeder Spalte und Zeile jeweils genau einmal vorkommt.

1	2	3	4	5
2	5	4	1	3
3	4	5	2	1
4	3	1	5	2
5	1	2	3	4

Abbildung 2.1: Lateinisches Quadtrat

Daraus hat sich das heutige Sudoku entwickelt, das sich nicht nur bei Mathematikern großer Beliebtheit erfreut.

2.1 Die Regeln

Diese Arbeit beschäftigt sich nur mit der meist verbreiteten Art von Sudokus. Dabei spielt man auf einem 9x9 Felder großen Spielfeld, das wiederum in neun 3x3 Felder große Blöcke eingeteilt ist. Weiter handelt es sich nur dann um ein Sudoku, wenn genau eine Lösung vorhanden ist. Ein Sudoku gilt dann als gelöst, wenn jede Zeile, jede Spalte und jeder Block die Ziffern 1 bis 9 genau einmal enthält.

4	8	1	5	6		3		
7	6	9	3	2	4		5	
3	5			7		6		
	9	7		8	5	2	1	
1					6	5	4	
5	4		1				8	6
	7	6	9	5	3		2	
2		5		4			3	
9	3	4					6	5

Abbildung 2.2: Sudoku

2.2 Begriffserklärung

Ein Sudoku besteht aus 81 *Feldern* oder *Zellen*. Diese bilden ein Quadrat der Größe 9x9, das *Grid*. Aufgrund dieser Aufteilung hat ein Sudoku 9 *Zeilen* und 9 *Spalten*. Das Grid wird in 9 Unterquadrate geteilt, die jeweils 3x3 Felder groß sind. Diese werden *Blöcke* genannt. Zeilen, Spalten und Blöcke werden unter dem Begriff *Figur* zusammengefasst. Die Nummerierung der Blöcke erfolgt zeilenweise von links oben nach rechts unten.
Vorgaben sind Zahlen, die schon von Anfang an gegeben sind.
In **Abbildung 2.2** sieht man im mittleren Block sogenannte *Kandidaten*. Ein Kandidat ist eine Zahl, die in der Zelle noch möglich ist. Jede Zelle hat ihre eigene Liste mit Kandidaten.
In der Beschreibung der Lösungstechniken ist es notwendig bestimmte Felder zu betrachten. Hierzu wird eine Abkürzung verwendet, die Zeile und Spalte enthält und somit eine Zelle eindeutig indentifiziert. *z2s3* meint zum Beispiel die Zelle in Zeile 2 und Spalte 3.
In der folgenden Abbildung sind die erläuterten Begriffe zum besseren Verständniss eingetragen.

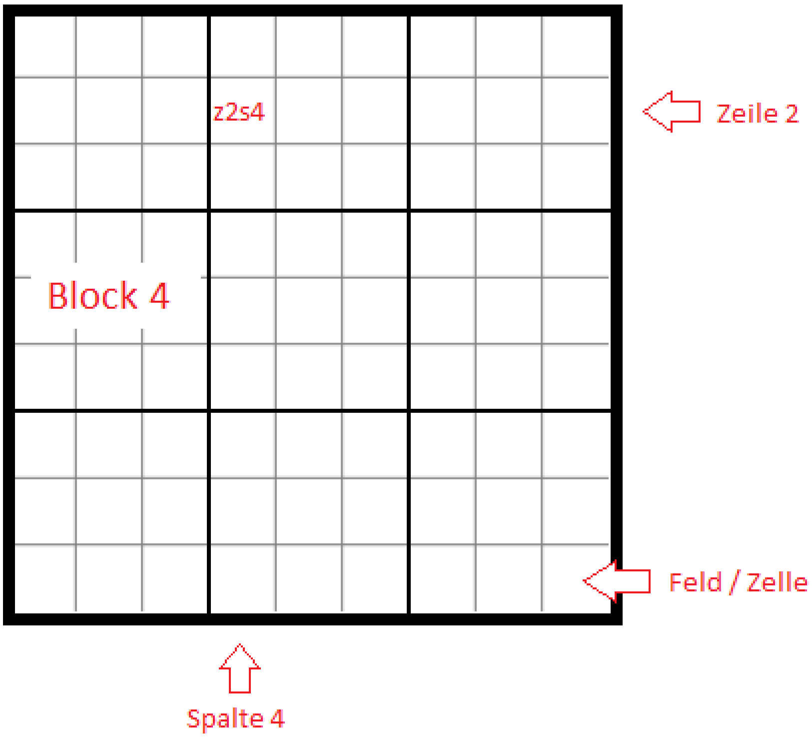

Abbildung 2.3: Begriffe

3 Lösungsmethoden

Alle in dieser Bachelorarbeit beschriebenen Techniken sind nicht im Rahmen dieser Arbeit entwickelt worden, sondern wurden aus verschiedenen Quellen zusammengetragen. Die Beschreibung der Lösungstechniken lehnt sich an die Beschreibung der Quellen an. Teile der Beispiele wurden aus den Quellen entnommen, dies ist entsprechend gekennzeichnet.
Grob kann man die Techniken zum Lösen von Sudokus in zwei Kategorien einteilen. Die erste Kategorie findet Zahlen heraus, die direkt in das Sudoku eingetragen werden können. Die Techniken der zweiten Kategorie entfernen Bedingungen in einzelnen Zellen des Sudokus.

3.1 Kandidatenlisten

Beim Lösen von Sudokus ist es üblich, in jedes Feld die Kandidaten einzutragen, die dort stehen können. Dabei wird vorerst nur die Sudoku Regel berücksichtigt, die besagt, dass in jeder Figure die Zahlen 1 bis 9 vorkommen müssen. Wenn in einer Zeile nun die Zahl 3 vorkommt, dann kann sie in der selben Zeile nicht nochmal vorkommen, daher kann sie aus allen Kandidatenlisten der Zellen in der selben Zeile gelöscht werden. Dasselbe gilt für Spalten und Blöcke. Immer wenn eine Ziffer in ein Feld eingetragen wird, dann muss der Spieler die Liste der Kandidaten aktualisieren.
Kandidatenlisten sind keine eigene Lösungstechnik, sind aber wesentlicher Bestandteil vieler Techniken.

3.2 Full House

Wenn in einer Figur 8 Zahlen eingetragen sind, dann kann die Technik *Full House* angewendet werden. Da in jeder Figur die Zahlen 1 bis 9 stehen müssen, kann die fehlende Zahl einfach per Ausschluss ermittelt werden.

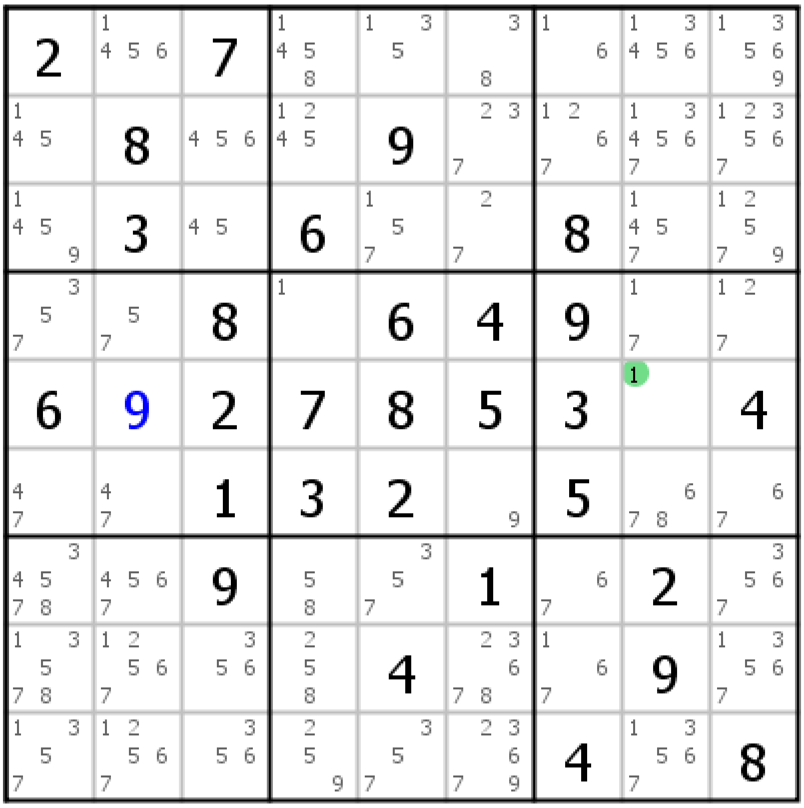

Abbildung 3.1: Full House

In **Abbildung 3.1** fehlt in Zeile 5 nur noch eine Ziffer. Da die Zahlen 2 bis 9 bereits vorhanden sind, kann in das Feld z5s8 die Zahl 1 eingetragen werden.

3.3 Naked Single

Bei der Technik *Naked Single* werden Kandidatenlisten verwendet. Diese Technik kann angewendet werden, wenn in der Kandidatenliste einer Zelle nur noch ein Kandidat steht. Dieser Kandidat kann dann in die Zelle eingetragen werden. Das funktioniert aufgrund des Aufbaus der Kandidatenlisten. Diese enthalten zuerst alle Kandidaten und es werden immer dann Kandidaten entfernt, wenn dieser Kandidat nicht mehr als Ziffer in der Zelle stehen könnte wel er dort eine Regel verletzen würde. Wenn also nur noch ein Kandidat in der Kandidatenliste steht, dann bedeutet das, dass dieser Kandidat die einzige Ziffer zwischen 1 und 9 ist, die in der Zelle stehen kann ohne eine Regel zu verletzen.

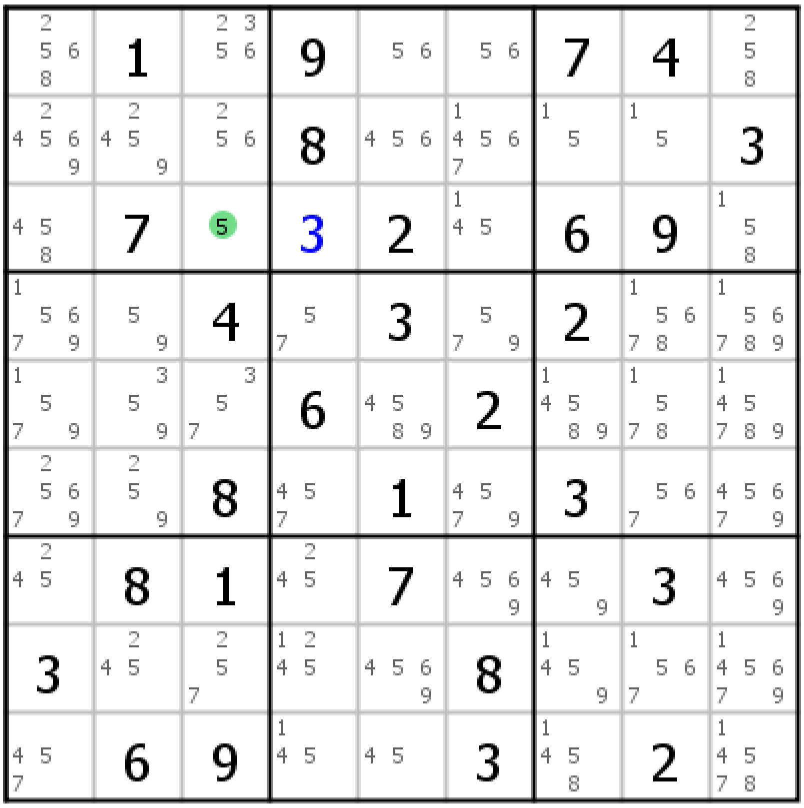

Abbildung 3.2: Naked Single

Im oben stehenden Beispiel **Abbildung 3.2** sieht man sofort, dass die Kandidatenliste in z3s3 nur noch einen Eintrag enthält. Dieser kann nun einfach eingetragen werden.

3.4 Hidden Single

Auch die Technck *Hidden Single* arbeitet wieder mit Kandidatenlisten. Wenn in einer Figur eine Kandidatenliste die einzige ist, in der eine bestimmte Zahl vorkommt, dann kann diese Zahl direkt in die Zelle eingetragen werden. Wenn in dieser Zelle die Zahl nicht stünde, dann gäbe es in der Figur keine Möglichkeit mehr, dass die Zahl auftaucht und damit wäre die Sudoku Regel verletzt, nach der jede Zahl genau einmal enthalten sein muss.

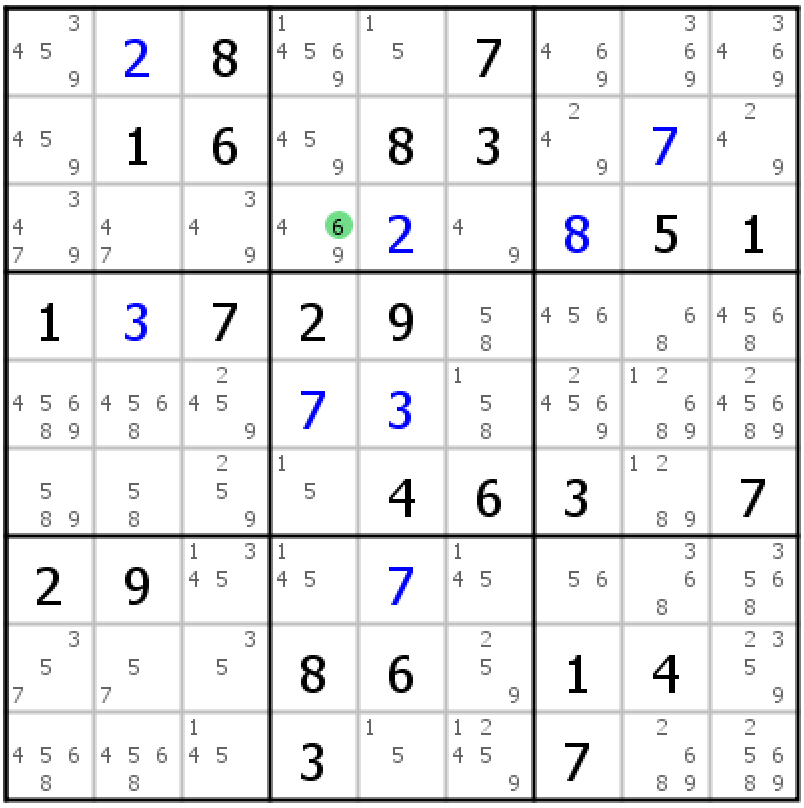

Abbildung 3.3: Hidden Single

In **Abbildung 3.3** sieht man, dass die Zahl 6 in der Zeile 3 nur in z3s4 vorkommen kann. Daher kann man sie dort eintragen.

3.5 Pointing Pair / Triple

Bei der Technik *Pointing Pair / Triple* müssen zum ersten mal die Kandidatenlisten mehrerer Felder gleichzeitig betrachtet werden, was diese Technik etwas schwerer macht. Ausserdem ist diese Technik die erste, die Kandidaten aus Kandidatenlisten entfernt und nur bedingt zum Einsetzen von Zahlen in das Sudoku führt.
Es werden die Kandidatenlisten in Blöcken jeweils zeilen- und spaltenweise betrachtet. Die Technik *Pointing Pair / Triple* kann angewendet werden, wenn in einem Block eine Kandidat nur in Kandidatenlisten der selben Zeile oder Spalte vorkommt. Dann kann jedes weitere vorkommen der Zahl in einer Kandidatenliste der selben Zeile oder Spalte entfernt werden. Das gilt, da die Zahl genau einmal in dem Block vorkommen muss. Da alle möglichen Vorkommen der Zahl in der selben Zeile oder Spalte liegen ist klar, dass die Zahl in dieser Zeile oder Spalte vorkommt. Da sie aber kein zweites mal in der Zeile oder Spalte vorkommen darf muss sie aus den Kandidatenlisten entfernt werden, die nicht im selben Block liegen.

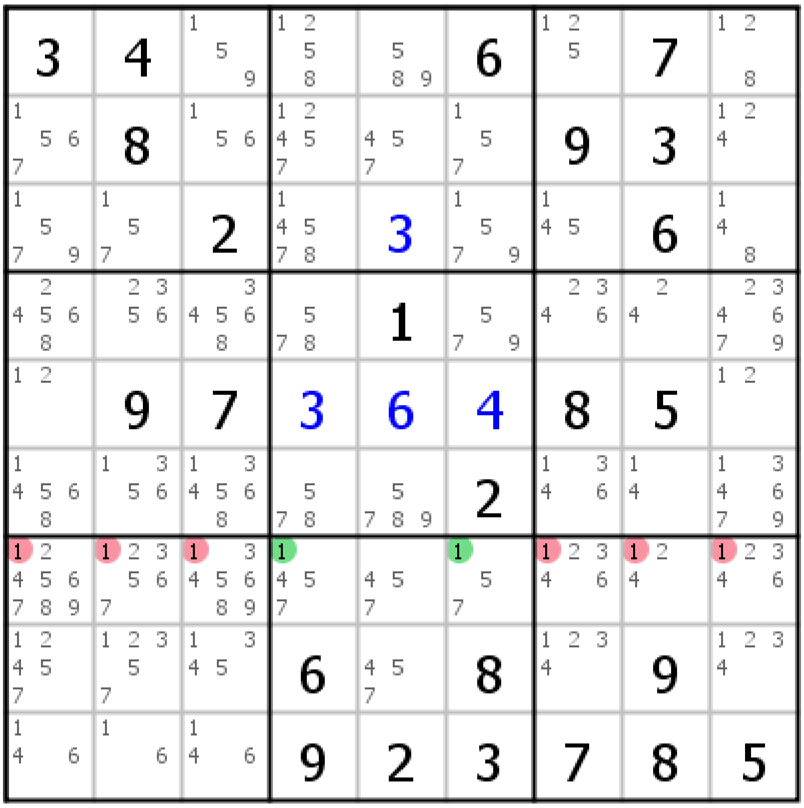

Abbildung 3.4: Pointing Pair / Triple

In **Abbildung 3.4** betrachten wir Block 8. Hier ist das Vorkommen der Zahl 1 in den Kandidatenlisten auf Zeile 7 beschränkt. Wie oben beschrieben können nun alle weiteren vorkommen in der selben Zeile, die nicht in Block 8 liegen aus den Kandidatenlisten entfernt werden. Im vorliegenden Beispiel führt das allerdings nicht dazu, dass eine neue Zahl in das Sudoku eingetragen wird. Dennoch ist das Sudoku nun genauer bestimmt, da weniger Möglichkeiten übrig sind.

3.6 Box-Line Reduction

Die Technik *Box-Line-Reduction* ist verwandt mit der Technik *Pointing Pair / Triple*. Hier wird das Sudoku zeilen- und spaltenweise betrachtet. Ist das Vorkommen einer Zahl in den Kandidatenlisten auf einen Block beschränkt, dann kann jedes weitere Vorkommen der Zahl aus den Kandidatenlisten der Zellen des selben Blocks gestrichen werden, die nicht in der Zeile oder Spalte liegen. Die Begründung dafür ist ähnlich der Begründung bei *Pointing Pair / Triple*. Da die Zahlen 1 bis 9 jeweils genau einmal in der Zeile oder Spalte vorkommen müssen und dieses Vorkommen auf einen Block beschränkt ist, ist klar, dass die Zahl letztendlich in diesem Block vorkommt und zwar in der gefundenen Zeile oder Spalte. Die Zahl kann aber nicht zweimal in dem Block vorkommen, daher kann sie aus den Kandidatenlisten des Blocks gelöscht werden, deren Zellen sich nicht in der Reihe oder Spalte befinden.

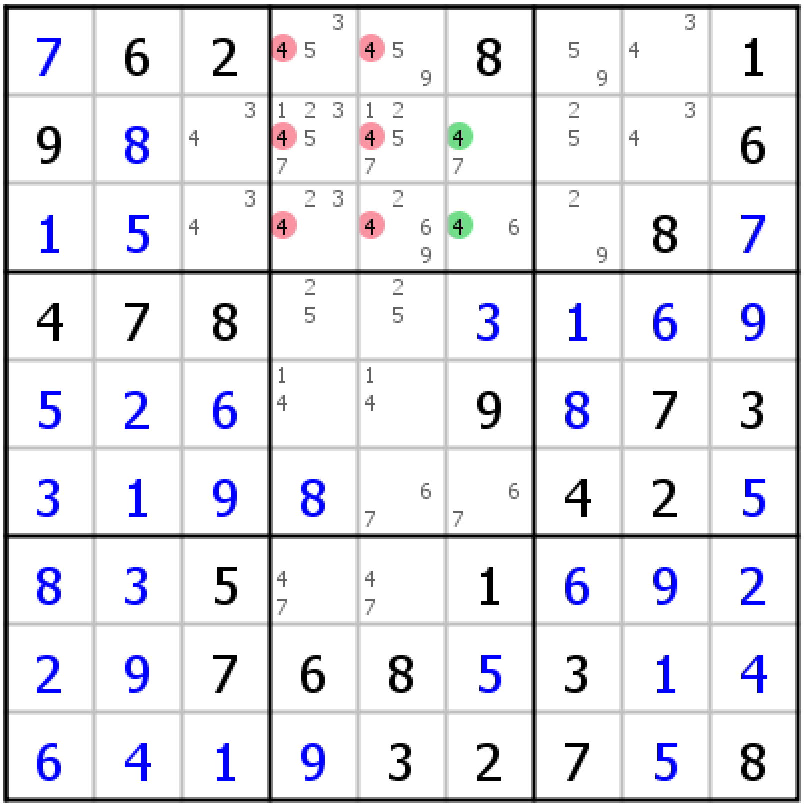

Abbildung 3.5: Box-Line Reduction

Wir betrachten Spalte 6 in **Abbildung 3.5**. Hier sieht man, dass das Vorkommen der Zahl 4 in dieser Spalte auf Block 2, den oberen Block, beschränkt ist. Anhand dieser Spalte sieht man also, dass die Zahl 4 entweder in z2s6 oder in z3s6 steht, also in jedem Fall in Block 2. Daher kann die Zahl 4 aus den Kandidatenlisten der anderen Zellen in Block 2 gestrichen werden.

3.7 Naked Subset

Die Technik *Naked Subset* ist ein Überbegriff für die Techniken *Naked Pair*, *Naked Triple* und *Naked Quadruple*. Alle Techniken Arbeiten nach dem selben Prinzip, der Unterschied liegt in der Anzahl der verwendeten Kandidatenlisten. Bei *NakedSubsets* sucht man nach Paaren, Tripeln oder Quadrupeln von Zellen in Figuren, nach Kandidatenlisten einer bestimmten Eigenschaft. Die Vereinigung der Listen muss eine bestimmte Anzahl Elemente enthalten. Bei Paaren sind das zwei, bei Tripeln drei und bei Quadrupeln vier Einträge in den Kandidatenlisten.
Findet man zum Beispiel ein Paar, das nur noch die selben beiden Zahlen enthalten kann dann ist klar, dass keine der Zahlen anderswo in der Figur stehen kann, da sonst für eine der Zellen keine Zahl mehr übrig bleibt. Daher können die beiden Zahlen dann aus den Kandidatenlisten aller anderen Zahlen aus der Figur entfernt werden. Die Begründung für Tripel und Quadrupel ist analog.
Es ist nicht nötig nach mehr als Quadrupeln zu suchen, da für jedes Naked Quintupel ein Hidden Quadrupel existiert.

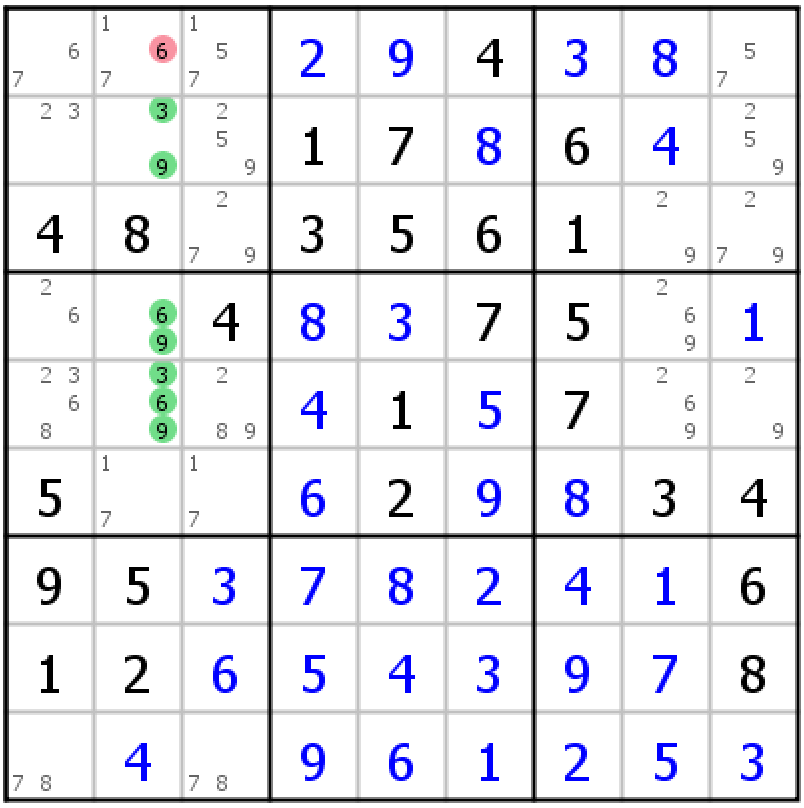

Abbildung 3.6: Naked Subset - Naked Triple

In **Abbildung 3.6** findet man das *Naked Subset*, bei dem es sich um ein *Naked Triple* handelt, in Spalte 2. Hier hat die Vereinigung der Kandidatenlisten der Zellen z2s2, z4s2 und z5s2 genau drei Einträge: 3, 6 und 9. Es gibt offensichtlich keine andere Möglichkeit, als die drei Zahlen auf diese Felder zu verteilen, Demnach können sie in der Reihe sonst nicht vorkommen und können aus den Kandidatenlisten der anderen Zellen entfernt werden.

3.8 Hidden Subset

Analog zu den *Naked Subset*-Techniken ist auch *Hidden Subset* ein Sammelbegriff. Er beinhaltet die Techniken *Hidden Pair*, *Hidden Triple* und *Hidden Quadruple*. Auch hier ändert sich nur die Anzahl der betrachteten Kandidatenlisten. Hier soll exemplarisch die Technik *Hidden Tuple* erklärt werden, im folgenden Beispiel wird dann die Technik *Hidden Quadruple* angewendet.
Wenn man in einer Figur zwei Zahlen findet, die ausschließlich in den zwei gleichen Zellen vorkommen können, dann müssen diese beiden Zahlen in die beiden Zellen gesetzt werden. Daher kann man alle anderen Kandidaten in den Zellen von der Kandidatenliste streichen.

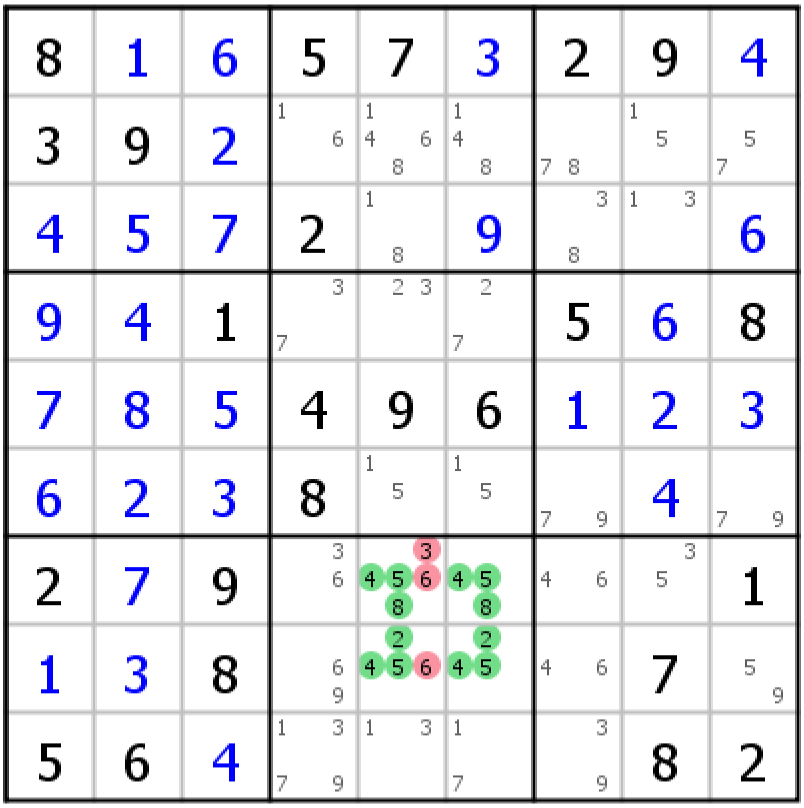

Abbildung 3.7: Hidden Subset - Hidden Quadruple

In **Abbildung 3.7** betrachten wir den Block 8 und in diesem Block die Zellen z6s5, z6s6, z7s5 und z7s6. Nur in diesen Zellen können die Zahlen 4, 5, 6 und 8 vorkommen. Da wir diese vier Zahlen nun auf die vier Zellen verteilen müssen gibt es dort keinen Platz mehr für andere Zahlen. Diese können also aus den Kandidatenlisten entfernt werden.

3.9 Fish

Die *Fish* Methoden sind ein Sammelbegriff für eine ganze Grupppe von Methoden, die alle nach dem gleichen Prinzip arbeiten. Wie echte Fische hat dieses Prinzip eine sehr große Anzahl Unterarten hervorgebracht. Kleine Fische wie zum Beispiel X-Wing sind von geübten Sudoku Spielern noch zu finden, wenn die Fische allerdings größer werden, dann sind sie nur noch mit sehr hohem Aufwand manuell zu finden und daher eher zur Verarbeitung mit dem Computer gedacht.

3.9.1 X-Wing

3.9.2 Swordfish

3.9.3 Jellyfish

3.10 Single Digit Patterns

3.10.1 Skyscarper

3.10.2 2-String Kite

3.10.3 Turbot Fish

3.10.4 Empty Rectangle

3.11 Wings

3.11.1 XY-Wing

3.11.2 XYZ-Wing

3.11.3 W-Wing

3.12 Sue de Coq

3.13 Coloring

3.14 Almost Locked Set

3.14.1 ALS XZ

3.14.2 ALS XY Wing

3.14.3 ALS Chain

4 Merkmalsextrahierung

4.1 Allgemeines Vorgehen

4.2 Entkopplung von konkreten Zahlen

5 Zusammenfassung und Ausblick