Einige Sudoku-Studien

Arnold Schönhage
Institut für Informatik der Universität Bonn

Dezember 2010

Vorbemerkung. Anfang Juni 2008 schickte mir *Wolfram Jehne* – einer vermittelnden Anregung von *Fritz Ostermann* folgend – ein umfangreiches Manuskript [J08] über seine mathematische Analyse von Sudokus, speziell im 9×9 Format, mit der Bitte um meine Hilfe bei dort anstehenden algorithmischen Problemen nebst deren Programmierung zur Klärung diverser Anzahl- und Strukturfragen. Mit beiden seit langem bekannt, seit meinen frühen Kölner Tagen, und weil ich Ende 2006 ohnehin schon ein Programm zum Lösen von Sudoku-Rätseln entwickelt hatte (s. [Sc2]), das hier gut als Basis taugte, bin ich dieser Bitte gern und mit regem Interesse gefolgt. Seither haben wir (zu zweit, zu dritt) häufiger miteinander konferiert, korrespondiert, neue Ideen wie auch Texte ausgetauscht. So hat sich inzwischen ein beachtliches Wissen zu diesem Thema angesammelt. Einige meiner Beiträge dazu möchte ich im folgenden Text zusammenfassend darstellen und damit allgemein – auch für Referenzzwecke – zugänglich machen.

Inhaltsverzeichnis

1 Grundlagen

Es erscheint zweckmäßig, diesen Text über 9×9 Sudokus weitgehend autark zu halten, zumal die von mir für meine Studien und vor allem auch im Hinblick auf bequeme Implementierungen gewählten Bezeichnungen, Indizierungen usw. teilweise sehr von den in [J08] benutzten abweichen. Deshalb gebe ich in diesem einleitenden Abschnitt zuerst eine eigene Darstellung der theoretischen Grundlagen, wie ich sie für meine Algorithmen benötige. Speziell gehören dazu auch neuartige Invarianten, meine sogenannten "Dreier-Mengen", die sich gerade bei einigen der theoretischen Untersuchungen als besonders nützlich erweisen werden.

1.1 Sudokus, Teilsudokus

Hier bezeichne A stets ein *volles* 9×9 Sudoku, anzusehen als 3×3 Blockmatrix $(A_{i,j})$ mit in $3{\times}3 = \{0,1,2\}\times\{0,1,2\}$ variierenden Indexpaaren (i,j) und zugleich auch als $A = (a_{\mu,\nu}) \in \{1,\dots,9\}^{9\times 9}$ unter Einhaltung der "Sudoku-Bedingungen", daß jede der Ziffern $1,\dots,9$ in jeder Zeile $a_{\mu,*}$, in jeder Spalte $a_{*,\nu}$ und in jedem der Blöcke $A_{i,j}$ vorkommt. Formal sind das also Abbildungen, die jedes Indexpaar $(\mu,\nu)\in 9\times 9$, wobei $9 = \{0,1,2,3,4,5,6,7,8\}$ (wie z. B. in [Hal, Kap. 11]) gemeint ist, mit einer Ziffer (= Zeichen aus jenem *Alphabet* $\Sigma = 9^{+}\setminus 1 = 10\setminus 1$) "belegen". In alternativer Form der Sudoku-Bedingungen, bei vollen Sudokus A äquivalent zu obigem, wird verlangt, daß die 27 *partiellen* Abbildungen, die man durch Einschränkung von A auf eine der Zeilen $\{\mu\}\times 9$, der Spalten $9\times\{\nu\}$ oder der *Felder*

$$F_{i,j} = \{(\mu,\nu) : 3i \le \mu \le 3i+2,\ 3j \le \nu \le 3j+2\} \qquad (i,j<3) \tag{1.1}$$

von 9×9 erhält, sämtlich *injektiv* sind. Daß es sehr viele A solcher Art gibt, ist jedem Freunde von Sudoku-Rätseln wohlvertraut. Dennoch sei gleich hier, zu erster Illustration und auch zum Zwecke späterer Referenzen, das nebenstehende **Urbeispiel** U als so ein Sudoku von besonders schöner, regelmäßiger Struktur präsentiert.

$U =$
```
|1 2 3 | 4 5 6 | 7 8 9|
|4 5 6 | 7 8 9 | 1 2 3|
|7 8 9 | 1 2 3 | 4 5 6|
|3 1 2 | 6 4 5 | 9 7 8|
|6 4 5 | 9 7 8 | 3 1 2|
|9 7 8 | 3 1 2 | 6 4 5|
|2 3 1 | 5 6 4 | 8 9 7|
|5 6 4 | 8 9 7 | 2 3 1|
|8 9 7 | 2 3 1 | 5 6 4|
```

Bei Rätseln wie auch bei Diskussion mancher Strukturfragen werden statt voller A allerdings nur partielle Belegungen P vorgegeben sein, die natürlich zumindest den Sudoku-Bedingungen in jener alternativen Form der Injektivität auf den Zeilen, Spalten und Feldern genügen sollen. Solche P sollen dann (wie in [J08]) *Präsudokus* genannt werden. Deren Fortsetzbarkeit zu einem vollen Sudoku Q auf 9×9 ist damit aber nicht immer gesichert, wie das nebenstehende Gegenbeispiel P_o zeigt, bei dem Vorgabe dieser 8 Ziffern für $q_{2,2}$ die einander widersprechenden Werte 3 <u>und</u> 5 bedingt.

$P_o =$
```
|- - - | 3 5 - | - - -|
|- - - | - - - | 3 5 -|
|- - ? | - - - | - - -|
|3 - - | - - - | - - -|
|5 - - | - - - | - - -|
|- - - | - - - | - - -|
|- 3 - | - - - | - - -|
|- 5 - | - - - | - - -|
|- - - | - - - | - - -|
```

Im Unterschied dazu möge *"Teilsudoku"* hier als Bezeichnung für Präsudokus dienen, die zu einem vollen Sudoku fortsetzbar sind, und wenn das auf genau eine Weise möglich ist, wie z. B. bei Sudoku-Rätseln üblicherweise verlangt, mag dafür die Bezeichnung "eindeutiges Teilsudoku" passend erscheinen.

1.2 Die Sudokugruppe

In der symmetrischen Gruppe zur Indexmenge 9×9 (also einer S_{81}) interessieren uns nur solche Operationen g, die bei Wirkung auf Sudokus A (durch Transport der belegenden Ziffern) wieder Sudokus liefern. Formal gesehen soll also solch g durch Vorschaltung der Permutation g^{-1} auf dem Definitionsbereich 9×9 der Sudokus vermöge der Zuordnung $A \mapsto A \cdot g^{-1}$ auf der Menge X aller 9×9 Sudokus eine *sudokutreue* Abbildung $\hat{g} : X \to X$ induzieren. Angesichts der geometrischen Natur der Sudoku-Bedingungen sind speziell folgende Typen offensichtlich sudokutreu:

- (R) Permutation der Feld- bzw. Blockzeilen $F_{i,*}$, $A_{i,*}$, mit den Bezeichnungen R für den Zyklus 0120, R_0 für 121, R_1 für 020, R_2 für 010;
- (S) Permutation der Feld- bzw. Blockspalten $F_{*,j}$, $A_{*,j}$, mit den Bezeichnungen S für den Zyklus 0120, S_0 für 121, S_1 für 020, S_2 für 010;
- (r) *lokale* Permutation der Zeilen $\{\mu\} \times 9$, in jeweils einer der Feldzeilen $F_{i,*}$, bzw. der Zeilen $a_{\mu,*}$ in jeweils einer der Blockzeilen $A_{i,*}$, also für $\mu \in \{0,1,2\}$ oder für $\mu \in \{3,4,5\}$ oder für $\mu \in \{6,7,8\}$;
- (s) *lokale* Permutation der Spalten $9 \times \{\nu\}$ in jeweils einer der Feldspalten $F_{*,j}$, bzw. der Spalten $a_{*,\nu}$ in jeweils einer der Blockspalten $A_{*,j}$, also für $\nu \in \{0,1,2\}$ oder für $\nu \in \{3,4,5\}$ oder für $\nu \in \{6,7,8\}$;
- (t) Transposition durch Tausch der Positionen (μ,ν) mit (ν,μ) für $\mu,\nu < 9$, hier mit t bezeichnet – und Schreibweise $A^t_{j,i}$ für die Blöcke von $\hat{t}(A)$.

Als *Sudokugruppe* sei hier (wie in [J08]) die von diesen Operationen in jener S_{81} erzeugte Untergruppe G definiert. Mit Schreibweisen wie R_0R_1 für "R_0 nach R_1" gelten darin speziell (wie halt in jeder S_3) die Regeln $R_0R_1 = R_1R_2 = R_2R_0 = R$, $R_2R = RR_1 = R_0$ usw., und analoge Regeln für die Permutationen vom Typ (S) oder (r), (s). Die R's kommutieren mit den S's, ebenso alle lokalen Zeilen- mit allen lokalen Spaltenoperationen. Konjugation mit t verwandelt Spaltenoperationen in analoge Zeilenoperationen, und umgekehrt. Ohne t wird eine halb so große Untergruppe G_0 erzeugt. Mit achtfacher S_3 ergeben sich so die Kardinalitäten

$$\#G_0 = 6^8 = 1\,679\,616, \qquad \#G = 3\,359\,232\,. \tag{1.2}$$

In Abschnitt 1.4 werden wir sehen, daß die oben erwähnte Zuordnung $g \mapsto \hat{g} \in X^X$ injektiv ist, also die Gruppe G mittels ihrer Wirkung auf X als zu G isomorphe Gruppe $\hat{G} \subset X^X$ einbettet, die wir später dann – schon zur Vereinfachung der Schreibweise – ebenfalls mit G bezeichnen werden.

Die in diesem Kontext naheliegende Frage, ob es außerhalb von G in $S_{9\times 9} \setminus G$ noch weitere sudokutreue Permutationen geben könnte, werde ich mit einem *"Nein"* beantworten[1], allerdings erst im Teil 3, denn in meinem Beweis dieser strukturellen Charakterisierung von G als Untergruppe *aller* sudokutreuen $h \in S_{9\times 9}$ werde ich einige Tatsachen über die Existenz gewisser Sudokus benötigen, die erst mit den Resultaten aus Teil 2 gesichert sein werden.

[1] ganz im Einklang mit unserer Intuition

1.3 Ziffern- und Mischgruppe

Eine weitere Möglichkeit, Sudokus A in Sudokus zu transformieren, ist durch ein-eindeutige Umbenennungen der belegenden Ziffern, also durch Permutationen π des Bildes $\Sigma = \{1, \dots, 9\}$ solcher A gegeben: Deren Gruppe Z (eine S_9) wirkt auf der Menge X aller 9×9 Sudokus, indem jedes $\pi \in Z$ vermöge der Zuordnung $A \mapsto \pi{\cdot}A$ eine sudokutreue Abbildung $\hat{\pi} : X \to X$ induziert. Weil solche $\hat{\pi}$ für $\pi \neq id_\Sigma$ fixpunktfrei sind, ist die Zuordnung $\pi \mapsto \hat{\pi} \in X^X$ injektiv und bewirkt so Einbettung von Z als eine dazu isomorphe Gruppe $\hat{Z} \subset X^X$, auch *Zifferngruppe* genannt, die später (vereinfachend) wieder mit Z bezeichnet wird. Das Zusammenspiel von Sudoku- und Zifferngruppe betreffend ist zunächst einmal leicht zu sehen, daß ihre Wirkungen vertauschbar sind: für alle $A \in X,\ \hat{g} \in \hat{G},\ \hat{\pi} \in \hat{Z}$ gilt $\hat{\pi}\hat{g}(A) = \hat{g}\hat{\pi}(A)$, denn anschaulich ist es egal, ob solche Umbenennung $\hat{\pi}$ vor oder nach einem Transport der belegenden Ziffern durch ein $\hat{g}$ erfolgt. Mehr formal mit den verketteten Abbildungen schließend ist es einfach deren Assoziativität $\pi{\cdot}(A{\cdot}g^{-1}) = (\pi{\cdot}A){\cdot}g^{-1}$.

Nach diesen Vorbereitungen kommen wir nun zu einer weiteren wesentlichen **Definition:** Das Kompositum $G^* = \hat{Z}\,\hat{G} \subset X^X$ wird *totale Mischgruppe* genannt. Dem Vorbild in [J08] folgend haben wir hierzu als grundlegendes Resultat den

Satz 1.1. *Es gilt* $\hat{Z} \cap \hat{G} = \{id_X\}$, *nebst Isomorphie* $G \simeq \hat{G}$, *die totale Mischgruppe ist also direktes Produkt* $G^* \simeq Z \times G$ *der Ordnung* $\#G^* = 9! \cdot \#G$.

Ein Beweis dafür folgt im nächsten Abschnitt mit den dort erst einzuführenden *Dreier-Mengen*. Schließlich sei hier am Rande auch noch die *(kleine) Mischgruppe* $M = \hat{Z}\hat{L_1}\hat{L_2} \simeq Z \times L_1 \times L_2$ erwähnt, wobei L_i (für $i < 3$) die allein von den *lokalen* Operationen der Typen (r), (s) in den Feldzeilen $F_{i,*}$ bzw. Feldspalten $F_{*,i}$ erzeugten Untergruppen von G bezeichnet, also jeweils der Ordnung $\#L_i = 6^2$. Diese Gruppe M operiert fixpunktfrei auf X und hat die Ordnung $\#M = 9! \cdot 6^4 = 470\,292\,480$.

1.4 Dreier-Mengen

Jetzt führe ich die schon erwähnten Dreier-Mengen ein, hier zuerst als Hilfsmittel beim Beweis von Satz 1.1 und dann besonders im Teil 4 ganz wesentlich benötigt. Zu jedem Sudoku $A = (A_{i,j}) = (a_{\mu,\nu})$ gehört ein durch die Funktionen

$$(1.3) \qquad v_l(A_{i,j}) := \{a_{\mu,3j+l} : 3i \le \mu \le 3i{+}2\}, \quad w_l(A_{i,j}) := \{a_{3i+l,\nu} : 3j \le \nu \le 3j{+}2\}$$

auf den Indextripeln $i, j, l < 3$ definiertes System $vw(A)$ von 27 “**v**ertikalen” sowie von 27 “**w**aagerechten” dreielementigen Ziffernmengen, also mit Werten in der 84-elementigen Menge D all solcher dreielementigen Teilmengen $d \subset \{1, \dots, 9\}$. Abstrahiert man hier von den speziellen Indizes, <u>wo</u> in A diese jeweils 27 *Dreier* $d_1, \dots, d_{27} \in D$ vorkommen, oft mit Mehrfachnennung solcher d’s, dann induziert das zwei Abbildungen $v_A : D \to \mathbb{N}$ und $w_A : D \to \mathbb{N}$ mit den bei jenen v_l bzw. w_l auftretenden Multiplizitäten als Werten (manchmal auch “multi-set” genannt) und somit dann auch entsprechende *Partitionen* p_A und q_A von 27.

Abgesehen von der Transposition t, die Spalten mit Zeilen, also v_A mit w_A und p_A mit q_A vertauscht, bleiben diese Daten bei Anwendung beliebiger Operationen $g \in \hat{G}$ ungeändert, und selbst bei Hinzunahme von Permutationen $\pi \in \hat{Z}$ gilt immer noch $\{p_{\pi g(A)},\, q_{\pi g(A)}\} = \{p_A, q_A\}$ für jedes Sudoku A.

Die hiermit etablierten Sudoku-Invarianten werfen ganz neue Fragen auf, z. B. nach den Häufigkeiten ihres Auftretens oder dem Schwierigkeitsgrad zugehöriger Sudoku-Rätsel. Ferner lassen sich diese Invarianten bei Bedarf noch wesentlich verfeinern, indem man auch die Kardinalitäten der wechselseitigen Durchschnitte jener Dreier-Familien $d_1, \ldots, d_{27}$ zusätzlich mit in den Blick nimmt.

Nun zum *Beweis von Satz 1.1*: Es genügt, für just ein "X-beliebiges" (nicht allzu speziell gewähltes) Sudoku A die Implikation

$$(1.4) \qquad \hat{g}(A) = \hat{\pi}(A) \implies g = id_{9\times 9} \wedge \pi = id_\Sigma \quad \text{für alle } \ g \in G,\ \pi \in Z$$

zu zeigen. Das hat auf Anhieb gleich mit dem erstbesten Sudoku aus einem Rätselheft funktioniert, das U von Seite 2 hingegen taugt dafür nicht. Interessanterweise ist aber auch das in [J08] als "zufälliges Beispiel" genannte Sudoku A geeignet und soll uns jetzt zu diesem Zweck dienen, hier mit seinen Dreier-Mengen präsentiert:

```
 ---------------------------
| 2+8+1 | 9-6-5 | 7*3*4 |
| 5-6-9 | 4*7*3 | 8+1+2 |
| 7*3*4 | 2+1+8 | 6-5-9 |
 ---------------------------
| 9 2 6 | 5 8 4 | 1 7 3 |
| 3 5 7 | 1=2=6 | 4 9 8 |
| 1 4 8 | 3 9 7 | 2:6:5 |
 ---------------------------
| 6=1=2 | 8 3 9 | 5 4 7 |
| 4*7*3 | 6:5:2 | 9 8 1 |
| 8 9 5 | 7 4 1 | 3 2 6 |
 ---------------------------
```

Die Multiplizitäten v_A der *vertikalen* Dreier sind =3 für 678 (so statt $\{6,7,8\}$), dann =2 für 135, 167, 249, 358 und =1 für 124, 129, 139, 149, 179, 235, 245, 248, 257, 289, 345, 359, 368, 467, 468, 679, also $27 = 3+2+2+2+2+16\cdot 1$ als Partition p_A.

w_A für *waagerechte* Dreier hat die Werte =4 für 347 [mit $**$ markiert], =3 für 128 [++], 569 [--], =2 für 126 [==], 256 [: :] und =1 für 126, 137, 147, 189, 236, 269, 357, 379, 389, 457, 458, 489, 589, also $27 = 4+3+3+2+2+13\cdot 1$ als Partition q_A.

Mittels der in Abschnitt 1.2 genannten Erzeuger von G seien die in (1.4) zu diskutierenden $g \in G$ in der Form $g = c{\cdot}h$ mit (R/S)-Anteil h und lokalem (r/s)-Anteil c dargestellt. Der andere Fall $g = c{\cdot}h{\cdot}t$, wo g zuerst transponiert, scheidet hier sofort aus, weil bei diesem A die unter $\hat{Z}$ invarianten p_A und q_A verschieden sind. Um den (R/S)-Anteil zu prüfen, seien die $w_l(A_{i,j})$ in (1.3) nach Blockspalten und -zeilen zu

$$(1.5) \qquad V_j = \{w_l(A_{i,j}) : i, j < 3\}, \qquad W_i = \{w_l(A_{i,j}) : j, l < 3\}$$

gebündelt. Aus $\#V_0 = 8$, $\#V_1 = \#V_2 = 9$ und $\#(V_0 \cap V_1) = 4$, $\#(V_0 \cap V_2) = 3$ folgt dann, daß der Prämisse in (1.4) genügende $\hat{g}$ die Blockspalten nicht permutieren, und analog aus $\#W_0 = 3$, $\#W_1 = \#W_2 = 9$ und $\#(W_0 \cap W_1) = 0$, $\#(W_0 \cap W_2) = 1$, daß solche $\hat{g}$ auch die Blockzeilen fest lassen, also (R/S)-Anteil $h = id_{9\times 9}$ haben.

Bei den jetzt noch übrigen lokalen (r/s)-Anteilen von $g = c$ wird der Nachweis genügen, daß der auf die linke Blockspalte $A_{*,0}$ wirkende (s)-Anteil und der auf die mittlere Blockzeile $A_{1,*}$ wirkende (r)-Anteil von g beide trivial sein müssen, denn so läßt $\hat{g}$ den "westlichen" Block $A_{1,0}$ fest, was $\pi = id_\Sigma$, $\hat{\pi}(A) = A$ und dann endlich auch Trivialität *aller* lokalen Anteile von $g = c$ impliziert. Zur Realisierung dieses Plans ist die Prämisse in (1.4) mit (1.3) zu kombinieren, um die Wechselwirkungen zwischen $\hat{g} = \hat{c}$ und $\hat{\pi}$ genauer zu verfolgen.

Wenn der auf $A_{*,0}$ wirkende (s)-Anteil die Spalte $a_{*,l}$ zur Spalte k (mit $l, k < 3$) von $\hat{c}(A) = \hat{\pi}(A)$ werden läßt, hierbei von allen lokalen (r)-Anteilen absehend, dann muß π den Dreier $x_i = v_k(A_{i,0})$ auf den Dreier $y_i = v_l(A_{i,0})$ abbilden. Dies für die lokalen Operatoren s_0, s_1, s_2, s und dazu jeweils passend gewählte Indexpaare (i, l) nutzend erhält man folgende Tabelle solcher Einschränkungen von π auf Dreier.

Tabelle 1.4 s. Abbildung einiger Dreier $x_i = v_k(A_{i,0})$ auf $y_i = v_l(A_{i,0})$, für auf $A_{*,0}$ wirkende lokale (s)-Anteile *op* von g durch ein aus der Prämisse in (1.4) resultierendes π.

op	i,l,k	$x_i \xrightarrow{\pi} y_i$		*Folgerungen*
s_0	0 0 0	2 5 7	2 5 7	$\#(x_0 \cap x_1) = 1$ und
	1 1 2	6 7 8	2 4 5	$\#(y_0 \cap y_1) = 2$??
s_1	0 0 2	1 4 9	2 5 7	$\#(x_0 \cap x_1) = 1$ und
	1 1 1	2 4 5	2 4 5	$\#(y_0 \cap y_1) = 2$??
s_2	0 0 1	3 6 8	2 5 7	$\#(x_0 \cap x_1) = 1$ und
	1 1 0	1 3 9	2 4 5	$\#(y_0 \cap y_1) = 2$??
s	0 0 1	3 6 8	2 5 7	also $\{6,8\} \xrightarrow{\pi} \{2,5\}$ und
	1 1 2	6 7 8	2 4 5	$\pi\colon 3 \mapsto 7 \mapsto 4 \mapsto 3$, so aber
	2 2 0	4 6 8	2 3 5	unpassend zu $A_{1,1}$ und $A_{2,1}$

Der bei s für π hergeleitete Zyklus 3743 ist in (1.4) mit allen rein lokalen Wirkungen auf Block $A_{1,1}$ inkompatibel. Den anderen Fall ss reduziert man durch Übergang zu g^{-1} (mit π^{-1}) auf den schon erledigten lokalen Anteil $(ss)^{-1} = s$. So bleibt als auf die Blockspalte $A_{*,0}$ wirkender (s)-Anteil einzig die identische Abbildung möglich.

Die (r)-Anteile, jetzt nur auf die mittlere Blockzeile $A_{1,*}$ wirkend, lassen sich analog behandeln: Wenn solch (r)-Anteil r_0, r_1, r_2 oder r die Zeile $a_{3+l,*}$ zur Zeile $3+k$ (mit $l,k < 3$) von $\hat{c}(A) = \hat{\pi}(A)$ macht, jetzt von (s)-Anteilen absehend, dann muß π den Dreier $x_j = w_k(A_{1,j})$ auf den Dreier $y_j = w_l(A_{1,j})$ abbilden. Dies mit jeweils passend gewählten Indexpaaren (j,l) nutzend erhält man als zweite Tabelle

Tabelle 1.4 r. Abbildung einiger Dreier $x_j = w_k(A_{1,j})$ auf $y_j = w_l(A_{1,j})$, für auf $A_{1,*}$ wirkende lokale (r)-Anteile *op* von g durch ein aus der Prämisse in (1.4) resultierendes π.

op	j,l,k	$x_j \xrightarrow{\pi} y_j$		*Folgerungen*
r_0	0 0 0	2 6 9	2 6 9	$\#(x_0 \cap x_1) = 1$ und
	1 1 2	3 7 9	1 2 6	$\#(y_0 \cap y_1) = 2$??
r_1	0 0 2	1 4 8	2 6 9	$\#(x_0 \cap x_1) = 1$ und
	1 1 1	1 2 6	1 2 6	$\#(y_0 \cap y_1) = 2$??
r_2	0 0 1	3 5 7	2 6 9	$\#(x_0 \cap x_1) = 1$ und
	1 1 0	4 5 8	1 2 6	$\#(y_0 \cap y_1) = 2$??
r	0 0 1	3 5 7	2 6 9	also $\{3,7\} \xrightarrow{\pi} \{2,6\}$ und
	1 1 2	3 7 9	1 2 6	$\pi\colon 1 \mapsto 5 \mapsto 9 \mapsto 1$, so aber
	2 2 0	1 3 7	2 5 6	unpassend zu $A_{0,2}$ und $A_{2,2}$

völlig analog zur ersten, hier beim von r induzierten zyklischen Zeilentausch 3453 mit resultierendem π-Zyklus 1591, der aber z. B. auf Block $A_{2,2}$ mit allen rein lokalen (r/s)-Wirkungen inkompatibel ist. Wieder reduziert Invertieren das noch fehlende rr auf r, also bleibt als einzig möglicher auf die Blockzeile $A_{1,*}$ wirkender (r)-Anteil nur die identische Abbildung. □

Dieser letzte Schritt im Beweis von Satz 1.1 hätte übrigens, den Fall r betreffend, mit Blockzeile $A_{0,*}$ so nicht funktioniert, die mit ihren nur drei (jeweils 3-fachen) waagerechten Dreiern dann doch wohl nicht so ganz "zufällig" erscheint. Ähnliche Dreier-Strukturen werden im Teil 4 – dann immer in *allen* Blockzeilen und Blockspalten auftretend – beim Studium der Fixsudokus eine wesentliche Rolle spielen.

Schließlich sei noch angemerkt, daß – nach Ausschluß von t und nicht trivialer (R/S)-Anteile von g – die Überprüfung von (1.4) mit diesem A für die nur noch $6^6 = 46656$ rein lokalen $g = c$ natürlich auch rasch mit einem kleinen Programm erledigt werden könnte, da solch g das dafür jeweils einzige zu testende π ja schon mit der 0-ten Zeile von $\hat{g}(A)$ bestimmt.

2 Ein Fortsetzungsproblem

Eine der Grundfragen in [J08] war, kurz gesagt, ob jedes "blockdiagonale" Präsudoku (mindestens) eine Fortsetzung zu einem Vollsudoku besitzt – und wenn ja, wieviele? Genauer formuliert, hier der Terminologie aus Abschnitt 1.1 folgend, seien durch Angabe von 27 die Diagonalfelder $F_{0,0}, F_{1,1}, F_{2,2}$ belegenden Ziffern drei Bijektionen $A_{i,i} : F_{i,i} \to \Sigma$ (für $i = 0, 1, 2$) als potentielle Blockdiagonale beliebig vorgegeben. Dann gibt es in der Tat, wie hier zu berichten sein wird, stets dazu kompatible Vollsudokus A, sogar in großen Anzahlen – von 95 514 bis 283 576 variierend.

2.1 Diagonalblöcke, (e, u, v)-Normierung

Blockdiagonale Präsudokus P haben die "Felddiagonale" $FD = F_{0,0} \cup F_{1,1} \cup F_{2,2}$ als Definitionsbereich. Es liegt nahe, solche P als Tripel von 3×3 Blöcken $B_i : 3\times 3 \to \Sigma$ $(i = 0, 1, 2)$ zu spezifizieren, die angesichts der Sudoku-Bedingungen stets bijektiv sein sollen. Um daraus formal sauber das so bestimmte P als partielle Abbildung $P : 9\times 9 \supset FD \to \Sigma$ zu konstruieren, deute ich auch die B_i als (auf $F_{0,0}$ definierte) Präsudokus, benutze aus der Sudokugruppe G, die hier nun auch in ihrer Wirkung auf partielle Abbildungen verstanden sei, die Diagonalfelder **t**auschenden Operationen $t_j = R_j S_j$ (für $j = 0, 1, 2$, vgl. Abschnitt 1.2), setze dann, damit *definierend,*

$$P = B_0 \diagdown B_1 \diagdown B_2 := B_0 \cup t_2(B_1) \cup t_1(B_2) \tag{2.1}$$

und notiere dazu für späteren Gebrauch gleich auch noch die Tausch-Regeln

$$t_2(P) = B_1 \diagdown B_0 \diagdown B_2, \quad t_1(P) = B_2 \diagdown B_1 \diagdown B_0 \quad \text{und} \quad t_0(P) = B_0 \diagdown B_2 \diagdown B_1\,. \tag{2.2}$$

Die Menge all solcher P sei (wie in [J08]) mit Y bezeichnet. Auf Vollsudokus $A \in X$ angewandt stiftet die Einschränkung auf die Felddiagonale FD vermöge $A \mapsto A_{|FD}$ eine Abbildung $\Phi : X \to Y$, und deren Umkehrung ordnet jedem $P \in Y$ die Menge $\uparrow P \uparrow = \{A \in X : \Phi(A) = P\}$ seiner Urbilder zu. So wird das eingangs genannte Fortsetzungsproblem zur Frage, ob Φ *surjektiv* ist, ob also $\#\uparrow P \uparrow > 0$ für alle $P \in Y$ gilt, und nach Möglichkeit möchte man diese Anzahlen natürlich genau kennen.

Auf den ersten Blick wären das $\#Y = (9!)^3 \approx 4.78 \cdot 10^{16}$ zu untersuchende Fälle; diese abschreckend riesige Zahl läßt sich jedoch durch strukturelle Vorbetrachtungen wesentlich reduzieren. Dazu wird[2] in der totalen Mischgruppe G^* (auf Präsudokus wirkend) der Stabilisator H^* von Y betrachtet, also aller $\pi g \in G^*$ mit $g \in H \subset G$ als dem Stabilisator der Felddiagonalen FD. Mit der Transposition t, dem zu S_3 isomorphen Erzeugnis $\langle t_1, t_2 \rangle$ der Blockpermutationen und den am Ende von 1.3 erwähnten lokalen Untergruppen L_i erhält man H als das Kompositum $H = L_1 L_2 L_0 \langle t, t_1, t_2 \rangle$, somit nach Hinzufügen der Zifferngruppe Z jene Gruppe $H^* \subset G^*$ in der Form

$$H^* = ZH = MQ \quad \text{mit der Untergruppe} \quad Q = L_0 \langle t \rangle \langle t_1, t_2 \rangle \quad \text{von } G. \tag{2.3}$$

Beim Problem, die Anzahlen $\#\uparrow P \uparrow$ für so viele $P \in Y$ zu bestimmen, liegt der Nutzen dieser Betrachtungen darin, daß es wegen der Anzahl-Invarianz

$$\#\uparrow \pi g(P) \uparrow \; = \; \#\uparrow P \uparrow \quad \text{für alle } \pi g \in H^* \text{ und jedes } P \in Y \tag{2.4}$$

schon genügt, solche Anzahl für jeweils nur *ein* P aus jeder der H^*-Bahnen von Y

[2] wieder dem Vorbild [J08] folgend

zu berechnen. Es bestehen allerdings gewisse Schwierigkeiten, theoretisch volle Einsicht in die Natur dieser Bahnen zu gewinnen, weil es neben vielen der vollen Länge

$$\#H^* = \#M \cdot \#Q = 470\,292\,480 \cdot 432 \quad (= 9! \cdot 6^7 \cdot 2) \tag{2.5}$$

auch kürzere Bahnen gibt, bei ca. 1.5 % von Längen $\#M \cdot k$ mit echten Teilern k von $\#Q = 432$; weitere Einzelheiten dazu werde ich in Abschnitt 2.3 berichten.

Betrachten wir jedoch statt der H^*-Bahnen nur die (meist um den Faktor 432) kürzeren M-Bahnen von Y, dann führt uns das, weil die Mischgruppe M auf X und Y fixpunktfrei operiert, zu dem (von W. Jehne analysierten) glatten Fall, daß alle Bahnen die gleiche Länge haben. Meine jetzt folgenden teils recht technischen Erläuterungen, auch bezüglich passender Kodierung für eine effiziente Gestaltung meiner Programme, werden zunächst nur diese einfache Situation betreffen. Erst im Abschnitt 2.3 werde ich dann näher ausführen, wie man mittels der Wirkungen jener Gruppe Q der Ordnung 432 aus (2.3) den Rechenaufwand zur Bestimmung der Anzahlen um einen weiteren Faktor > 400 reduzieren kann.

Jedem Präsudoku $P \in Y$, wie in (2.1) als $P = B_0 \setminus B_1 \setminus B_2$ dargestellt, wird durch folgende **Normierung** ein wohlbestimmter Repräsentant $P_\star = \pi g_1 g_2(P)$ seiner M-Bahn zugeordnet, gemäß $M = Z L_1 L_2$ mit diesen $\pi \in Z$, $g_1 \in L_1$, $g_2 \in L_2$:

$$E = \begin{array}{|ccc|} \hline 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ \hline \end{array} \qquad B' = \begin{array}{|ccc|} \hline 1 & * & * \\ * & a & b \\ * & c & d \\ \hline \end{array} \quad \text{mit } a < b, c, d$$

Mit nebenstehendem "Einsblock" E wird π durch die Bedingung $\pi(B_0) = E$ bestimmt, und dann seien die lokalen Operatoren g_1, g_2 derart gewählt, daß damit die (passend verschobenen) Blöcke $B_1' = t_2 g_1 \pi t_2(B_1)$ und $B_2' = t_1 g_2 \pi t_1(B_2)$ von der nebenstehenden Form B' resultieren. Das ist jeweils auf genau eine Weise möglich, indem zuerst die Ziffer 1 an ihren Platz gebracht und dann ggf. (mittels lokalem r_0 oder s_0) Tausch der Zeilen oder Spalten 4, 5 bzw. 7, 8 bei g_2 auch noch die für die "Südost-Elemente" geforderte Zusatzbedingung erfüllt wird. So hat man schließlich $P_\star := E \setminus B_1' \setminus B_2'$ erreicht. Die Menge dieser Bahnrepräsentanten $P_\star \in Y$ sei mit $Y_\star$ bezeichnet, hiermit (primär aus typographischen Gründen) von den entsprechenden Notationen $\hat{A}$, $\hat{Y}$ aus [J08] abweichend. Bei komplizierterem Argument wie $P' = \psi g(P)$ (z. B. mit $\psi g \in H^*$) sei auch $\star(\psi g(P))$ als alternative Bezeichnung für $P_\star'$ verabredet.

Weil so normierte Blöcke B_1', B_2' in $P_\star$ jeweils genau $8!/4 = 10080$ verschiedene Belegungen zulassen, ist die obige "Schreckenszahl" $(9!)^3$ damit auf "*nur noch*"

$$\#Y_\star = 10\,080^2 = 101\,606\,400 \tag{2.6}$$

reduziert – leider aber immer noch unpraktikabel groß. Wollte man z. B. die Anzahl aller Sudokus, für die man wegen der uniformen Bahnlänge $\#M = 9! \cdot 6^4$ nach (2.4) die Summenformel $\#X = 9! \cdot 6^4 \cdot \sum_{P \in Y_\star} \#\!\uparrow\! P \!\uparrow$ erhält, durch direkte Berechnung dieser gut 101 Millionen $\#\!\uparrow\! P \!\uparrow$ finden, dann würde das mit meinem im nächsten Abschnitt vorgestellten Programm (mit ca. 1 sec pro P) einige Jahre dauern!

Erst mit der in Abschnitt 2.3 folgenden weiteren Reduktion der Zahl notwendiger $\#\!\uparrow\! P \!\uparrow$ Berechnungen wurde es überhaupt möglich, dieser Datenflut Herr zu werden. Selbst schon bei den Hilfsprogrammen zu dieser Reduktion, die dazu dienten, jene 101 Millionen Ausgangsdaten erst einmal auf praktikables Format zu komprimieren, war Verwendung einer programmtechnisch ökonomischen Darstellung der Blöcke solcher $P \in Y_\star$ wichtig, die jetzt noch erläutert werden soll.

Um die (normierten) 3×3 Blöcke B_1, B_2 solcher P jeweils in einem 32-bit Wort in bequemer, leicht handhabbarer Form zu speichern, ist es zweckmäßig, die Ziffern $1, \ldots, 9$ so umzubenennen, daß $2, \ldots, 9$ (um 2 vermindert) zu $0, \ldots, 7$ werden, und 8 dann für die 1 steht, deren Speicherung bei normierten Blöcken B ja entfallen kann. Für solch ein B sei das kodierende Wort dann

$$(2.7) \qquad w = \sum_{j=0}^{7} z_j \cdot 16^j \quad \text{mit Zuordnung dieser 'hexits' } z_j \text{ wie in } B = \begin{vmatrix} 8 & z_7 & z_6 \\ z_5 & z_4 & z_3 \\ z_2 & z_1 & z_0 \end{vmatrix}$$

mit der entsprechend transformierten Zusatzbedingung $(\beta): z_4 < z_0, z_1, z_3$, und auch die umgekehrte Zuordung $w \mapsto B(w)$ spezifiziert eine eindeutige Abbildung auf der Menge W aller 32-bit Worte mit jeweils 8 *verschiedenen* Hexadezimalen $z_j < 8$, die (β) genügen, was in Bitpositionen wie 2^3 oder 2^{31} Raum für andere nützliche Markierungen läßt. Kodierung der $P \in Y_\star$ erfolgt entsprechend mit Wortpaaren $(u, v) \in W^2$ durch die Zuordnung $(u, v) \mapsto P(u, v) := E \setminus B(u) \setminus B(v)$, und $N(u, v) := \# \uparrow P(u, v) \uparrow$ bezeichne dann die Anzahl möglicher Fortsetzungen.

Später wird es nützlich sein, auch den Einsblock E so kodiert zu denken, nämlich als $E = B(e)$ mit dem speziellen Hex-Wort $e = 76543210$ – hier sind solche Worte (wie im "TP-Book" [SGV]) stets 'from low to high' zu lesen[3]. Mehrfach wird uns speziell $P(e, e) = E \setminus E \setminus E$ in Sonderrollen begegnen, z. B. als Repräsentant der einzigen M-Bahn, die zugleich H^*-Bahn von Y ist, und zu der übrigens auch die Blockdiagonale $\Phi(U)$ des Ursudokus U gehört, wie auch als das (einzige) P mit maximaler Zahl $N(e, e)$ von Fortsetzungen – nebst weiterer Eigenheiten von $P(e, e)$ bei den Fixsudokus in Teil 4.

2.2 Programm zur Zählung möglicher Fortsetzungen

Grundlage dieses Abschnitts ist mein Programm zum Lösen von Sudoku-Rätseln, dessen technische Einzelheiten in [Sc2] ausführlich beschrieben sind. Hier wird es genügen, dessen zentrale Routine *INFER* mit den darin benutzten Datenstrukturen kurz vorzustellen. Ausgehend von dem anfangs gegebenen Präsudoku arbeitet sich *INFER*, die Sudoku-Regeln nutzend, mittels sukzessiv ausgeführter *Inferenzen* schrittweise vor, dabei den jeweils erreichten Wissensstand als sogenannte *Muster* speichernd. Das sind 81-Tupel K von Kennworten $k_n(i, j)$ für $i, j < 3$, $n \in \Sigma$, die als 9-stellige Bitmasken (mit Zahlenwerten < 512) mit ihren 1-bits jeweils anzeigen, *wo* Ziffer n im Feld $F_{i,j}$ nach derzeitigem Wissen noch stehen könnte. Zu Beginn wird das *leere Muster* mit 511 in allen 81 Worten erzeugt, das dann durch sukzessive Einarbeitung der Ziffern des vorgegebenen Präsudokus zu modifizieren ist.

Als ein (triviales) Beispiel betrachtet würde so Eingabe des Einsblocks E (als Präsudoku, hier nur das Feld $F_{0,0}$ belegend) zu einem Muster K' mit den Werten $k_n(0, 0) = 1, 8, 64,\ 2, 16, 128,\ 4, 32, 256$ (reinen Potenzen von 2) nebst weiteren wie z. B. $k_8(0, 1) = 219$, $k_8(1, 0) = 455$ (mit jeweils binärer Quersumme 6) führen, was so als Hinweis auf die Systematik der Kodierung solcher Muster genügen mag. Mit

$$(2.8) \qquad f(K) := \#\{(n, i, j) :\ i, j < 3\ \wedge\ 1 \leq n \leq 9\ \wedge\ k_n(i, j) \in \{2^\nu : \nu < 9\}$$

wird jedem Muster K die Anzahl der darin schon *fixierten* Ziffern zugeordnet – beim vorigen Beispiel also $f(K') = 9$ – und bei Inferenzschritten jeweils laufend angepaßt.

[3]dennoch wird *dieses* e nicht "transzendent"!

Wird so $f(K)= 81$ erreicht, dann ist damit die gesuchte Lösung gefunden. Es kann aber auch sein, daß solche Inferenzen zu einem $k_n(i,j) = 0$ führen, woraus dann folgt, daß sich das vorgegebene Präsudoku nicht zu einem Vollsudoku fortsetzen läßt.

Unser Hauptinteresse gilt hier dem dritten Fall, daß *INFER* nach Ausschöpfung all seiner Inferenzmöglichkeiten ein K erreicht hat, dessen $k_n(i,j)$ zwar allesamt positiv sind, dies aber mit $f(K) < 81$. Genauer gilt dann $f(K) \leq 77$, denn bei keinem Feld $F_{i,j}$ kann $k_n(i,j) > 1$ für nur genau ein n gelten, weil so die anderen 8 fixierten Ziffern diese eine auch fixiert hätten (mittels “Verdrängung”, vgl. [Sc2]). Ferner muß es neben einem so *defekten* Feld $F_{i,j}$ (mit maximal 7 fixierten Ziffern) in der gleichen Feldzeile $F_{i,*}$ oder gleichen Feldspalte $F_{*,j}$ mindestens noch ein weiteres defektes Feld geben, weil sonst die voll fixierten Belegungen jener anderen vier Felder auch auf $F_{i,j}$ alle Ziffern durch “Zeilungen” und “Spaltungen” fixiert hätten. Indem man diese Situation noch etwas weiter analysiert, gelangt man zu dem hilfreichen

Lemma 2.1. *Erreicht INFER nach Erledigung aller Inferenzen ein Muster K mit $f(K) = 77$, dann hat dies Fortsetzungen zu genau zwei verschiedenen Vollsudokus.*

Sonst, wenn *INFER* bei einem Muster K mit $f(K) < 77$ “stagniert”, zerlegt es eines der darin noch mehrdeutigen Kennworte $u = k_n(i,j)$ mit $2^\mu < u < 2^{\mu+1}$ in dessen höchste Zweierpotenz $u' = 2^\mu$ und $u'' = u - u'$, bildet daraus zur disjunkten Fallunterscheidung zwei alternative Submuster K', K'', indem es das $k_n(i,j)$ in K durch u' bzw. u'' ersetzt, und ruft rekursiv sich selbst ein- oder zweifach mit diesen K', K'' als neuen Anfangsmustern auf. Als mögliche Rückgabewerte eines Aufrufs von *INFER* sind dabei $m = 0, 1, 2$ vorgesehen, mit $m = 0$ im Falle der Nichtfortsetzbarkeit zu einem Vollsudoku, $m = 1$ bei eindeutig erreichtem Vollsudoku A, das *INFER* dann mitliefert, oder $m = 2$ bei mehrdeutiger Lösbarkeit. Falls der erste Aufruf (mit K') den Wert $m' = 2$ liefert, gibt das rufende *INFER* sofort $m = 2$ zurück, sonst wird auch m'' aus dem zweiten Aufruf mit K'' benötigt, danach mit Rückgabe von $m := 2$, falls $m'' = 2$, sonst $m := m' + m''$. — Terminierung dieser Arbeitsweise von *INFER* wird dadurch gesichert, daß jede Änderung des jeweils aktuellen Musters, aufgrund von Inferenzen oder bei solchen rekursiven Aufrufen, die totale binäre Quersumme all seiner 81 Kennworte jeweils *verkleinert*.

Jetzt ist mit wenigen Worten erklärt, wie ich diese Routine *INFER* zu einer *zählenden* Version *CINF* (mit C für <u>c</u>ounting) umprogrammiert habe: Es genügte, die Konventionen für die Rückgabewerte m und ihre Verarbeitung bei jenen rekursiven Aufrufen so zu ändern, daß m hier die Anzahl der verschiedenen $A \in X$ sein soll, zu denen sich das bei Aufruf von *CINF* vorgegebene Muster fortsetzen läßt. Dazu werden bei den rekursiven Aufrufen dann stets beide Zweige benötigt, um daraus den äußeren Rückgabewert $m := m' + m''$ zu bilden. Wegen dieser *vollen* Bearbeitung des ganzen Verzweigungsbaums ist der Zeitbedarf von *CINF* allerdings meist wesentlich höher als bei *INFER,* im Schnitt etwa 1.5 sec für die Berechnung eines Wertes $N(u,v)$ zu solchen blockdiagonalen Präsudokus $P(u,v)$ gegenüber nur 1 bis 2 msec bei *INFER* für das Lösen auch schwieriger Sudoku-Rätsel. – Immerhin gelang es, gestützt auf Lemma 2.1, durch “Abschneiden” aller dieser Doppelblüten jene durchschnittliche Zeit von 1.5 sec auf etwa 0.9 sec zu senken.

Abschließend sei hier am Rande noch erwähnt, daß all diese Programme im Rahmen des in [SGV] beschriebenen *TP*-Modells laufen, derzeit mittels Simulation auf einem etwas gealterten 2.4 GHz Pentium 4 aus dem Jahre 2005.

2.3 Weitere Reduktion durch Äquivalenzklassenbildung

Um die Zahl der $P(u,v)$, auf die *CINF* zur Bestimmung von $N(u,v)$ anzuwenden ist, zu verkleinern, betrachte ich jetzt die Bahnen des vollen Stabilisators H^* aus (2.3) und die damit in $Y_\star$ durch $P' \sim P \iff P \in H^*(P')$ induzierte Äquivalenzrelation, hier besser gleich für kodierende Wortpaare $(w,y),(u,v) \in W^2$ umformuliert, also

$$(w,y) \sim (u,v) \iff P(u,v) \in H^*(P(w,y))\,, \tag{2.9}$$

und als *der* Repräsentant solcher Klasse $(w,y)^\sim$ sei jeweils das darin *lexikographisch minimale* Paar (u,v) ausgezeichnet, also mit $u < w \vee (u = w \wedge v \leq y)$. Erinnert sei dabei nochmals an die aus (2.4) folgende Anzahl-Invarianz $N(w,y) = N(u,v)$. Mit Blocktausch $t_0 = R_0S_0 \in H^*$ folgt nach der letzten Regel in (2.2) unmittelbar $(u,v) \sim (v,u)$. Deshalb gilt bei jenen minimalen Repräsentanten stets $u \leq v$, und wegen $(w,y) \sim (y,w)$ genügt es, solche nach (2.9) zugeordneten (u,v) nur für die Paare in $WY = \{(w,y) \in W^2\colon w \leq y\}$ zu berechnen, was die große Zahl in (2.6) auf $\#WY = 5080 \cdot 10081 = 50\,808\,240$ reduziert, also auf etwa die Hälfte. Bei späterer Auszählung zur Bestimmung der Größen $m(u,v)$ all solcher Klassen $(u,v)^\sim$ sind dann die zugeordneten Paare (w,y) mit $w < y$ jeweils doppelt anzurechnen.

Um zu gegebenem (w,y) seinen Repräsentanten $(u,v) \sim (w,y)$ zu berechnen, werden im Prinzip einfach alle 432 Elemente der Gruppe Q aus (2.3) probiert. Nähere Einzelheiten der dabei benutzten Methode erläutere ich zuerst für die 72 Elemente g oder gt der Untergruppe $Q_0 = L_0\langle t\rangle$ von Q, mit lokalen Operationen $g \in L_0$. Auf $P(w,y) = E \setminus B(w) \setminus B(y)$ angewandt liefert solch gt (bzw. $g \neq id$) das Präsudoku $P' = gt(E) \setminus t(B(w)) \setminus t(B(y))$, — bzw. $g(E) \setminus B(w) \setminus B(y)$, — das dann aber wegen $gt(E) \neq E$ — bzw. $g(E) \neq E$ — neu zu normieren ist, beginnend mit der Permutation $\pi \in Z$, die $\pi gt(E) = E$ bzw. $\pi g(E) = E$ erfüllt. Programmtechnisch läßt sich das bequem mit einer Tabelle *PCW* von 72 "Perm-Code-Words" p für die hier möglichen πt und π verwalten. Durch die volle Normierung dieser P' gehört so zu jedem der 72 *PCW*-Elemente p vermöge der durch $P'_\star = E \setminus B(w') \setminus B(y')$ induzierten Zuordnung $w \mapsto w' = q(w)$ wie auch $y \mapsto y' = q(y)$ eine Abbildung $q : W \to W$ auf der Menge W der normierte 3×3 Blöcke kodierenden 32-bit Worte. Diese q's bilden übrigens eine zu Q_0 isomorphe Gruppe, hier ebenfalls Q_0 genannt. Jetzt auch noch $(w',y') \sim (y',w')$ berücksichtigend und vorerst solche Paare nur bzgl. Q_0 (lexikographisch) minimierend erhält man so

$$(u_0,v_0) = \min_{q \in Q_0} (\min(q(w),q(y)),\ \max(q(w),q(y))) \tag{2.10}$$

als einen ersten Kandidaten für den gesuchten Repräsentanten (u,v) von $(w,y)^\sim$. Bei Vorgabe von p und w benötigt mein Hilfsprogramm *PQW* zur Berechnung des Wortes $q(w)$ etwa 0.2 μs, und so geht Berechnung solch eines (u_0,v_0) in etwa 29 μs.

Zur Minimierung bzgl. der *vollen* Gruppe Q aus (2.3) sind auch die Operationen t_1 und t_2 einzubeziehen. Per Blocktausch t_2 entsteht nach der ersten Regel in (2.2) das Sudoku $t_2(P(w,y)) = B(w) \setminus E \setminus B(y)$, das meist (sofern nicht $B(w) = E$) erneuter Normierung bedarf. Anwendung der zuvor beschriebenen Methode auf $\star(t_2(P(w,y)))$ liefert dann aber ein zu (2.10) analoges Paar (u_2,v_2), ebenso per Anwendung auf $\star(t_1(P(w,y)))$ ein (u_1,v_1) und so endlich das gewünschte Gesamtminimum $(u,v) := \min((u_0,v_0),(u_1,v_1),(u_2,v_2))$ als Repräsentanten von $(w,y)^\sim$, und all das in einer Rechenzeit von weniger als 87 μs.

Dieser Algorithmus *WY2UV*, zu jedem Wortpaar $w \leq y$ dessen Repräsentanten (u, v) berechnend, könnte so variiert werden, daß er jeweils abbricht, wenn erstmals ein q mit $a = \min(q(w), q(y)) < w$ oder $a = w$ und $\max(q(w), q(y)) < y$ erreicht wird – entsprechend auch bei den q-Schleifen für $(u_1, v_1), (u_2, v_2)$, so daß er nur zuende kommt, wenn (w, y) selbst schon solch (u, v) ist, und dann könnte gezählt werden, wieviele verschiedene zu (w, y) äquivalente Paare dabei produziert wurden (maximal $3 \cdot 144 = 432$), um so die Klassengrößen $m(u, v)$ zu bestimmen.

Noch bequemer (und schneller, ohne viel weiteres Programmieren) war es jedoch, einfach für jedes der gut 50 Millionen Paare $w \leq y$ deren Repräsentanten (u, v) (in ca. 4000 sec) zu berechnen und in einer (ca. 400 MB langen) Folge *UVD* von Doppelworten zu speichern, bei $w < y$ im untersten hexit von v jeweils mit $v' = v + 8$ statt v durch solch “Verdopplungsbit” für dann 2-fache Zählung von (u, v) markiert. Mittels aufsteigender Sortierung dieser Wortpaare und Zählung aller Mehrfachnennungen ließ sich *UVD* dann einfach zur Ergebnisdatei *TPFUV* verdichten, die volle Information über diese Repräsentanten (u, v) aller Äquivalenzklassen von W^2 (als Kodierung von $Y_\star$) und deren jeweilige Größe $m(u, v)$ in diesem Format enthält:

119 Abschnitte $(u,\ vm\text{-Liste},\ 0)$ — nebst $nuv =$ totale Länge in Worten,

wobei die vm-Liste zum jeweiligen u die v's der Paare (u, v) in aufsteigender Folge enthält, als Einzelwort v bei der meist vorliegenden Standardgröße $m(u, v) = 432$, sonst als $\hat{v} = v + 2^{31}$ (Markierung im obersten hexit von v) und dann von weiterem Wort $m(u, v) < 432$ gefolgt. Jede vm-Liste endet beim nachfolgenden Wort 0, die Folge dieser u-Abschnitte bei $nuv = 240938 < e$, denn $e \leq u$ für alle $u \in W$. Hier nun einige wichtige Fakten aus *TPFUV* als Tabelle mit dem diesbezüglichen

Satz 2.2. *Die Äquivalenzrelation (2.9) zerlegt die Menge W^2 der $Y_\star$ kodierenden Wortpaare in $k^* = 237\,083$ Klassen. Deren Größen $m(u, v) = m$ sind stets Teiler von 432, mit Frequenzen $f(m)$ ihres Vorkommens wie in Tabelle 2.3. Diese $f(m)$ sind zugleich die Anzahlen der H^*-Bahnen in Y mit Längen $m \cdot \#M = m \cdot 9! \cdot 6^4$.*

Tabelle 2.3.
Aufteilung von $Y_\star$ in $k^* = 237\,083$ Äquivalenzklassen, mit jeweils $f(m)$ Klassenvertretern (u, v), zu denen es genau $m(u, v) = m$ zu $P(u, v)$ äquivalente normierte blockdiagonale Präsudokus $P(w, y)$ gibt.
Speziell für $u = e$ gibt es 170 solche Klassen, dort mit Anzahlen $f_e(m)$ für $m(e, v) = m$, insbesondere gilt $m(e, e) = 1$. — Bemerkenswert ist ferner, daß nur (e, e) als einziger Klassenrepräsentant in *TPFUV* die Form (u, u) hat.

m	$f_e(m)$	$f(m)$	$m \cdot f(m)$
1	1	1	1
3	1	1	3
4	–	2	8
6	1	2	12
12	2	8	96
24	–	4	96
27	2	2	54
36	4	14	504
54	9	19	1026
72	2	46	3312
108	24	118	12744
144	–	44	6336
216	124	3 356	724896
432	–	233 466	100 857312
total	= 170	= 237 083	= 101 606400

Zu wünschen bleibt, daß einige der hier noch rätselhaft erscheinenden Eigenschaften dieser Bahnlängendaten anregend für zukünftige weitere Studien wirken mögen.

2.4 Zählergebnisse

Wie Tabelle 2.3 zeigt, haben fast 98.5 % dieser k^* Klassen die maximale Größe 432, mit einer mittleren Zahl von ca. 428.6 Paaren (w, y) je Klasse. Etwa um diesen Faktor wird also die Rechenzeit zur Bestimmung der $N(w, y)$ für alle $P(w, y) \in Y_\star$ bei Nutzung von $N(w, y) = N(u, v)$ für $(w, y) \sim (u, v)$ reduziert. Insbesondere folgt so (mit den (u, v) in *TPFUV*) für $N_0 := \sum_{P \in Y_\star} \# \uparrow P \uparrow$ die Summenformel

$$(2.11) \qquad N_0 = \sum_{(u,v)} m(u, v) \cdot N(u, v) \quad \text{und} \quad \#X = 9! \cdot 6^4 \cdot N_0$$

als Gesamtzahl aller 9×9 Sudokus. Durchführung all dieser etwa 237 Tausend $N(u, v)$-Berechnungen mittels k^*-fachem Aufruf des Programms *CINF* (vgl. 2.2) kostete etwa 60 Stunden Rechenzeit, im Mittel also ca. 0.91 sec pro Aufruf. Die dabei nebenher laufende Akkumulation der Produkte in (2.11) lieferte

$$(2.12) \qquad N_0 = 14\,184585\,201152 = 2^9 \cdot p \quad \text{mit der Primzahl} \quad p = 27704\,267971$$

und so dann auch $\#X = 6670\,903752\,021072\,936960$, erfreulicherweise ganz in Übereinstimmung mit der schon aus [FJ5] bekannten Anzahl, was hoffen läßt, daß meine am Ergebnis (2.12) beteiligten Programme *WY2UV* und *CINF* korrekt sind.

Volle Information über diese Anzahlen $N(u, v)$ gibt eine weitere Ergebnisdatei *TPFNUV*, die für jede der 237 083 Klassen jeweils das Worttripel $(u, v, N(u, v))$ enthält, diese (hier als ganze Zahlen $u + v \cdot 2^{32} + N(u, v) \cdot 2^{64}$ gedeutet) in aufsteigender Folge sortiert, also mit Gesamtlänge von $237083 \cdot 12 = 2844\,996$ Bytes. Ganz oben zeigt diese Datei mit $u = v = e$ die maximale Anzahl $N(e, e) = 283576$, direkt darunter $N(e, v) = 278740$ mit kodierendem Wort $v = 15320764$, andererseits ganz unten das minimale $N(u, v) = 95514$ für $(u, v) = (74563120,\ 56240371)$ und direkt darüber — mit $(u, v) = (74563120,\ 43052671)$ — die *ungerade* Anzahl $N(u, v) = 96841 = 113 \cdot 857$, was voreilige Vermutungen zu widerlegen vermag.

Interessierte Leser sind eingeladen, sich *TPFNUV* und auch *TPFUV* zwecks eigener Studien anzuschauen, die (wie dieser Text) via Link `Sudoku` auf [Sc1] abrufbar sind. Diesen Teil abschließend werden hier jetzt noch zwei Tabellen zur Illustration der Werteverteilung dieser $N(u, v)$ präsentiert. Die eine zeigt, daß die berechneten $N(u, v)$-Werte sehr streuen, die andere gibt ein grobes Bild von deren Verteilung um einen Mittelwert $\approx 139\,707$, nebst $137\,451$ als Median.

Tabelle 2.4 h.
Häufigkeiten von $N(u, v)$'s : Sei $h_k := \#\{n : \kappa(n) = k\}$, worin $\kappa(n)$ die Repräsentanten (u, v) in *TPFUV* mit Werten $N(u, v) = n$ zählt. Stets gilt $\kappa(n) \leq 16$, mit monoton fallenden Werten

	h_k	$k \cdot h_k$		h_k	$k \cdot h_k$
$k = 1$	17084	17084	$k = 9$	1387	12483
2	11078	22156	10	762	7620
3	9450	28350	11	351	3861
4	8249	32996	12	158	1896
5	6851	34255	13	59	767
6	5269	31614	14	22	308
7	3590	25130	15	13	195
8	2288	18304	16	4	64

der h_k. Die Gesamtzahl *verschiedener* $N(u, v)$-Werte ist $\sum_k h_k = 66615$, und (zur Kontrolle) bestätigt man anhand dieser Zahlen auch $\sum_k k \cdot h_k = 237083$.

$k=$	$a_k=$									
96– 99							1	1	1	4
100–109	10	14	20	28	68	74	160	241	317	474
110–119	608	811	1186	1444	1794	2151	2647	3088	3492	3916
120–129	4280	4626	4740	5080	5196	5259	5328	5343	5370	5521
130–139	5543	5633	5702	5700	5693	5652	5843	5782	5779	5872
140–149	5718	5612	5519	5229	5055	4754	4471	4148	3927	3673
150–159	3270	3087	3088	2916	2613	2655	2672	2595	2527	2347
160–169	2522	2331	2255	2183	1912	1745	1550	1304	1175	992
170–179	816	693	697	595	546	588	494	496	439	495
180–189	479	563	586	595	505	492	447	285	294	215
190–199	154	101	104	82	75	67	71	65	61	60
200–209	68	67	53	40	64	137	176	142	115	76
210–219	91	92	44	14	12	20	18	9	9	2
220–229	1	7	11	4	6	6	3	5	9	6
230–239	8	6	40	66	62	21	4	0	0	1
240–249	0	0	0	2	1	0	0	0	1	0
250–259	0	0	1	0	0	0	1	0	3	3
260–269	4	15	9	10	6	8	1	1	0	1
270–279	0	1	1	0	0	0	0	0	0	1
280–284	0	0	0	0	1					

Tabelle 2.4 a. Verteilung der $N(u,v)$-Werte anhand der aus der Datei *TPFNUV* ermittelten Anzahlen $a_k := \#\{(u,v) : 1000 \cdot k - 500 \leq N(u,v) < 1000 \cdot k + 500\}$, hier für $k = 96, \ldots, 284$ tabelliert. Demnach liegen mehr als 99 Prozent dieser $k^* = 237\,083$ Werte zwischen $110\,500$ und $189\,500$.

3 Charakterisierende Abgrenzung der Sudokugruppe

Jetzt komme ich auf die schon am Ende von Abschnitt 1.2 gestellte Frage zurück, ob es außerhalb der dort eingeführten Sudokugruppe G in $S_{9\times 9} \setminus G$ noch weitere sudokutreue Permutationen geben kann. Entsprechend betrachte ich dazu nun

(3.1) $$H := \{h \in S_{9\times 9} : A \text{ ist Sudoku } \Rightarrow h(A) \text{ ist Sudoku}\}$$

als die größte sudokutreue Untergruppe der S_{81} auf der Menge $F = 9 \times 9$ aller Sudoku-Ziffernpositionen und werde – wie angekündigt – dafür $H = G$ beweisen. Zu diesem Ende zeige ich, auf die im Abschnitt 1.1 erläuterte Blockeinteilung von Sudokus und deren unterliegende Klasseneinteilung von F als 3×3 Feldermatrix $(F_{i,j})$ der Positionen Bezug nehmend, als ersten Schritt das

Lemma 3.1. *Jedes $h \in H$ bildet Felder in Felder ab, genauer induziert jedes $h \in H$ eindeutig eine Bijektion $\eta\colon 3 \times 3 \to 3 \times 3$ mit $(\mu,\nu) \in F_{i,j} \Rightarrow h(\mu,\nu) \in F_{\eta(i,j)}$.*

Indirekter Beweis. Bei einem h, das diese Aussage verletzt, kann man ohne Einschränkung annehmen, daß es das Feld $F_{0,0}$ nicht voll in nur ein Feld abbildet, ggf. nach Vorschaltung eines $g \in G$, das in $h\cdot g$ statt h passende Feldvertauschungen bewirkt. Als "Ziel" betrachte ich das Feld $F_{k,l}$, in das h Position $(0,0)$ abbildet, dessen Indizes $k, l \leq 2$ also durch $h(0,0) \in F_{k,l}$ bestimmt sind. So gibt es dann (inklusive $(0,0)$) $m > 0$ "konforme" $u \in F_{0,0}$ mit $h(u) \in F_{k,l}$ und $n = 9 - m > 0$ "Abweichler" $v \in F_{0,0}$, deren Bilder $h(v)$ nicht im Ziel liegen. Als Ersatz für diese sollte es, weil h als Bijektion das Ziel ja füllen muß, n externe $x \in F \setminus F_{0,0}$ mit $h(x) \in F_{k,l}$ geben. Als solche $x = (\mu, \nu)$ kommen jedoch nur die Fälle $\mu = 0$, $\nu \geq 3$ oder $\mu \geq 3$, $\nu = 0$ in Zeile 0 bzw. Spalte 0 von F in Betracht, weil es zu jedem anderen $(\mu, \nu) \in F \setminus F_{0,0}$ ein Sudoku A mit $a_{0,0} = 1 = a_{\mu,\nu}$ gibt[4], für das $h(A)$ dann mit zwei Einsen in $F_{k,l}$ kein Sudoku wäre.

Ähnlich erkennt man, daß diese externen x alle zu Spalte 0 oder alle zu Zeile 0 von F gehören müssen, denn sonst könnte man, von einem beliebigen Sudoku A ausgehend, weil dafür stets $\{a_{0,\nu} : \nu \geq 3\} \cap \{a_{\mu,0} : \mu \geq 3\}$ nicht leer ist, dessen Spalten 3–8 und Zeilen 3–8 mittels einer Operation $g \in G$ so permutieren, daß das Sudoku $A' = g(A)$ in zweien jener externen Positionen x die gleiche Ziffer hat und $h(A')$ somit kein Sudoku wäre. In der weiteren Diskussion sei unterstellt, daß die externen x alle zur *Spalte* 0 gehören – sonst hätte man ja gleich von Anfang an unter Vorschaltung der Transposition $h{\cdot}t$ statt h betrachten können. – Somit muß $n \leq 6$ gelten, und genauer sogar $n = 6$, denn mit $m \geq 4$ konformen u's wäre $\Delta = \{a_u : h(u) \in F_{k,l}\} \cap \{a_{\mu,0} : \mu \geq 3\}$ für jedes Sudoku A nicht leer und dessen Zeilen 3–8 ließen sich so zu einem Sudoku A' umordnen, für das $h(A')$ im Ziel eine Ziffer doppelt hätte, also kein Sudoku wäre. Schließlich folgt so auch, daß die $m = 3$ konformen u's zur Spalte 0 gehören müssen, weil sonst wieder $\Delta \neq \emptyset$ wäre, usw.

Bisher ist gezeigt, daß ein $h \in H$ (oder $h{\cdot}t$), das $F_{0,0}$ nicht voll in nur ein Feld abbildet, die Spalte 0 von F auf ein (3×3)-Feld $Q = F_{k,l}$ abbilden müßte. Dann könnte h aber auch $F_{1,0}$ nicht voll in nur ein Feld abbilden, denn bei Fixierung von $k', j' \leq 2$ durch $h(3,1) \in F_{k',l'}$ für $Q' = F_{k',l'}$ als neues Ziel sind die $v = (\mu, 0)$ für $\mu = 3, 4, 5$ diesbezüglich nun ja 'Abweichler'. Wie im Vorangehenden würde so folgen, daß h die Spalte 1 von F auf dieses Feld Q' abbildet, wobei die Alternative mit der *Zeile* 3 hier schon allein wegen $h(3,0) \in Q$, $h(3,0) \notin Q'$ entfällt – ohne neue Diskussion der Transposition! – Die übrigen 63 Positionen in den Spalten 2–8 von F bildet h bijektiv irgendwie auf $F \setminus (Q \cup Q')$ ab.

So vorbereitet nutze ich nun die lokale Spaltenoperation $s_2 \in G$, die in F die Spalten 0 und 1 vertauscht, und bilde damit $h' = h{\cdot}s_2{\cdot}h^{-1} \in H$ als *vermeintlich* sudokutreue Abbildung, die aus einer Bijektion $q : Q \to Q'$ nebst ihrer Umkehrung $q' : Q' \to Q$ und der Identität auf $F \setminus (Q \cup Q')$ besteht. Dieser indirekte Beweis von Lemma 3.1 endet mit dem Widerspruch, daß es zu solchem h' stets ein Sudoku A gibt, so daß $h'(A)$ kein Sudoku ist. Um letzteres zu zeigen, denke ich mir h' mittels passender Feldspalten- und Feldzeilen-Operationen $g \in G$ wenn nötig so transformiert, daß einer der Fälle $Q = F_{0,0}$, $Q' = F_{0,1}$ oder $Q = F_{0,0}$, $Q' = F_{1,1}$ vorliegt. Im ersten dieser Fälle sei $q'(0,3) = (\mu, \nu)$. Dann führt jedes Sudoku A mit $a_{\mu,\nu} = a_{3,3} = 1$ zu jenem Widerspruch. Im zweiten Fall sei $q'(3,3) = (\mu, \nu)$ und A irgendein Sudoku mit $a_{3,3} = 1$, $a_{\mu,\nu} = 2$. Hier kommt nun das Resultat aus 2.4 ins Spiel, daß sich beliebig vorgegebene Bijektionen $B_0 : F_{0,0} \to \Sigma$ und $B_1 : F_{1,1} \to \Sigma$, also blockdiagonale Präsudokus, jeweils zu einem vollen Sudoku fortsetzen lassen, und weil dessen Werte auf $F \setminus (Q \cup Q')$ die Werte auf Q und Q' schon eindeutig bestimmen, kann $h'(A)$ so kein Sudoku sein. □

[4] etwa aus dem Urbeispiel U von Seite 2 mittels Zeilen- und Spaltentausch leicht herleitbar

Im zweiten Schritt soll jetzt untersucht werden, welche Permutationen von 3×3, in Lemma 3.1 als η benannt, so von irgendeinem $h \in H$ induziert werden können, welche der insgesamt 9! Feldpermutationen bei den $h \in H$ überhaupt vorkommen. Vorbereitend sei als Terminologie verabredet, daß zwei Felder $F_{i,j}$ und $F_{i',j'}$ als *schräg zueinander* gelten oder auch *'schräges (Feld)paar'* genannt werden können, wenn (technisch formuliert) $i \neq i' \wedge j \neq j'$ gilt. Damit hat man als glatte Antwort

Lemma 3.2. *Permutationen $h \in H$ bilden schräge Feldpaare auf schräge Paare ab. Zu jedem solchen h gibt es in der von R, R_0, S, S_0 und der Transposition t erzeugten Untergruppe von G eine eindeutig bestimmte Feldpermutation g, so daß $h\cdot g$ jedes der Felder $F_{i,j}$ auf sich abbildet.*

Beweis. Für die Behauptung in der ersten Zeile sei $i \neq i' \wedge j \neq j'$, und mit dem von h nach Lemma 3.1 induzierten η gelte $\eta(i,j) = (k,l)$ und $\eta(i',j') = (k',l')$. Dann betrachte ich die Positionen $(\mu,\nu) \in F_{i,j}$ mit $h(\mu,\nu) = (3k,3l)$ und $(\mu',\nu') \in F_{i',j'}$ mit $h(\mu',\nu') = (3k',3l')$ und dazu ein Sudoku A mit $a_{\mu,\nu} = a_{\mu',\nu'} = 1$, das es ja zu beliebiger Vorgabe auf zwei zueinander schrägen Blöcken immer gibt. Weil $h(A)$ als Sudoku – hier direkt positiv argumentierend – weder in Zeile $3k$ noch in Spalte $3l$ zwei Einsen haben kann, folgt so $k \neq k' \wedge l \neq l'$, also Schrägheit der Bildblöcke.

Gestützt auf diese Eigenschaft solcher h läßt sich nun auch die weitere Behauptung einfach beweisen. Mittels passender Vorschaltung einer Blockpermutation g_0 aus der von R und S erzeugten Untergruppe von G wird erreicht, daß die von $h\cdot g_0$ induzierte Permutation η' zunächst einmal $\eta'(0,0) = (0,0)$ erfüllt und die zu $(0,0)$ schrägen Blockpositionen $(1,1)$, $(1,2)$, $(2,1)$, $(2,2)$ nur untereinander permutiert. Alsdann bewirkt passend vorgeschaltetes $g_1 \in \{1, R_0, S_0, R_0S_0\}$, daß die von $h\cdot g_0\cdot g_1$ induzierte Permutation η'' außer $\eta''(0,0) = (0,0)$ nun auch $\eta''(1,1) = (1,1)$ erfüllt und überdies auch die zu beiden schräge Blockposition $(2,2)$ fix läßt. Weil von den übrigen Positionen zu $(2,2)$ nur noch $(0,1),(1,0)$ schräg liegen, werden auch diese zwei von η'' fest gelassen oder aber vertauscht. In letzterem Fall wird (dann noch transponierend) $g = g_0\cdot g_1\cdot t$ zu dem g im Lemma, sonst ist das $g = g_0\cdot g_1$. ☐

Zum Beweis von $H = G$ wird es nach Lemma 3.2 nunmehr genügen, $h' \in G$ für solche "feldtreuen" $h\cdot g = h' \in H$ zu zeigen. Diesem letzten Schritt dient

Lemma 3.3. *Feldtreue $h \in H$ bilden Zeilen auf Zeilen und Spalten auf Spalten ab.*

Beweis. Um das zunächst für Zeile 0 von F zu zeigen, kann man vereinfachend $h(0,0) = (0,0)$ annehmen, ggf. mit auf $F_{0,0}$ wirkenden nachfolgenden lokalen Zeilen- oder Spaltenvertauschungen 010 oder 020, also mit $q \in \{1, r_1, r_2\} \cdot \{1, s_1, s_2\}$ und $q\cdot h$ anstelle von h. Dann sind noch zwei weitere $x \in F_{0,0}$ als die h-Urbilder zu $(0,1),(0,2)$ zu betrachten, ferner drei $x' \in F_{0,1}$ als Urbilder zu $(0,3),(0,4),(0,5)$ und drei $x'' \in F_{0,2}$ als Urbilder zu $(0,6),(0,7),(0,8)$. Daß bei sudokutreuem h alle diese x, x', x'' zu Zeile 0 gehören müssen, werde ich anhand der (nebenstehend nochmals wiedergegebenen) obersten Blockzeile des Beispiels U von Seite 2 zeigen.

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

Weil Zeile 0 des Sudokus $h(U)$ mit 1 beginnt, kann $(2,3)$ angesichts $u_{2,3} = 1$ keines der x' sein, gleiches folgt aber auch für $(2,4)$ und für $(2,5)$, wenn man mit lokaler Spaltenpermutation s für 3453 und $s(U)$ bzw. $ss(U)$ anstelle von U argumentiert. In gleicher Weise folgt, daß $(1,6),(1,7),(1,8)$ keine x'' sind. Ebenso scheiden

$(1,3),(1,4),(1,5)$ als mögliche x' und $(2,6),(2,7),(2,8)$ als mögliche x'' aus, wenn man so mit $S_0(U)$ – was die Blöcke $U_{0,1}, U_{0,2}$ vertauscht – anstelle von U schließt. Somit hat $h(U)$ in Zeile 0 jedenfalls schon alle Ziffern außer 2 und 3, weshalb die zwei Urbilder x nur $(0,1)$ und $(0,2)$ sein können.

Daß h auch Zeile 1 von F zeilentreu abbildet, läßt sich durch Vorschaltung von lokalem Zeilentausch mit $h{\cdot}r_2$ anstelle von h auf den Fall von Zeile 0 zurückführen, analog geht solches für die Zeile 2. Daß h ferner auch die Zeilen 3,4,5 jeweils zeilentreu abbildet, kann man dann mittels Betrachtung der feldtreuen Permutation $R_2{\cdot}h{\cdot}R_2 \in H$ auf das bisher Bewiesene reduzieren, analog geht solches für die Zeilen 6,7,8, und schließlich bildet h auch Spalten von F in Spalten ab, was sich unmittelbar aus der Zeilentreue der ebenfalls feldtreuen Permutation $t{\cdot}h{\cdot}t \in H$ ergibt. □

Jetzt ist $H = G$ einfach zu beweisen: Aus $h \in H$ wird mit passendem $g \in G$ nach Lemma 3.2 ein feldtreues $h{\cdot}g$, zu dem (wie im Beweis von Lemma 3.3) ein $q \in G$ existiert, so daß $q{\cdot}h{\cdot}g$ Zeile 0 (feldtreu) auf sich abbildet. Mit einem aus passenden lokalen Spaltenpermutationen in Spalten 0–2/3–5/6–8 von F zusammengesetzten $\hat{s} \in G$ wird $h' = \hat{s}{\cdot}q{\cdot}h{\cdot}g$ dann $h'(0,\nu) = (0,\nu)$ für alle ν erfüllen und alle Spalten jeweils feldtreu auf sich abbilden. Fügt man jetzt noch ein zu $\hat{s}$ analog aus passenden lokalen Zeilenpermutationen gebildetes (Zeile 0 fest lassendes!) $\hat{r} \in G$ hinzu, so daß $h'' = \hat{r}{\cdot}h'$ nun auch $h'(\mu,0) = (\mu,0)$ für alle μ erfüllt, dann muß dieses h'' nach Lemma 3.3 die Identität sein, was $h \in G$ impliziert. – Damit folgt als Hauptresultat

Satz 3.4. *Die in Abschnitt 1.2 eingeführte Sudokugruppe G stimmt mit der in (3.1) definierten maximalen sudokutreuen Untergruppe H der S_{81} auf $9{\times}9$ überein.*

4 Fixsudokus

4.1 Einleitendes

Satz 2.2 und Tabelle 2.3 zeigen unterschiedliche Bahnlängen bzgl. des Stabilisators H^* von Y. So wird es nicht überraschen, daß man auch bei Wirkung der Gruppe G auf die Menge X aller Sudokus G-Bahnen verschiedener Längen findet, es also Fixpunkte geben wird, hier formalisiert in dieser **Definition:** $A \in X$ sei *Fixsudoku* genannt, wenn es ein $f \in G \setminus \{id\}$ gibt, mit dem $f(A) = A$ gilt. – Dazu sogleich

Lemma 4.1. *Wenn A ein Fixsudoku ist und $B = \pi g(A)$ mit $\pi \in Z$ und $g \in G$ gilt, dann ist auch B ein Fixsudoku.*

Beweis. Man berechnet dazu $B = \pi g(A) = \pi g f(A) = g f g^{-1} \pi g(A) = f'(B)$ mit dem so gefundenen $f' = g f g^{-1} \neq id$ als passendem 'Fixierer' für B. □

Zur Einstimmung sei vorab erst einmal das Urbeispiel U von Seite 2 als solch ein Fixsudoku vorgestellt, das durch folgende "Umwälzung" in sich transformiert wird: Anwendung von SS – für Blockspaltentausch 0210 – erzeugt $SS(U)$ mit den Blockspalten $U_{*,1}, U_{*,2}, U_{*,0}$, und wendet man danach noch die *lokalen* zyklischen Zeilenvertauschungen $r \in L_0$, $r' \in L_1$, $r'' \in L_2$, also Zeilenpermutation 0120 3453 6786 an, dann erhält man mit $f_1 = rr'r''{\cdot}SS$ erstaunlicherweise wieder $f_1(U) = U$.
Später wird sich zeigen, daß dies ein sehr spezielles Beispiel sogenannter "Superfixe" ist, mit besonders großem Stabilisator als 9-elementiger Untergruppe von G_0.

4.2 Charakterisierende Eigenschaften von Fixsudokus

Als simple Methode, bei gegebenem Sudoku A zu entscheiden, ob es ein Fixsudoku ist, käme im Prinzip in Betracht, einfach mit allen von id verschiedenen $f \in G$ zu probieren, ob $f(A) = A$ gilt. Das wäre trotz der großen Zahl $\#G - 1 = 3359\,231$ bei der Rechenleistung heutiger PC's für ein einzelnes A (oder wenige A) durchaus praktikabel, würde aber bei vielen Milliarden Sudokus, die wir testen möchten, viel zu lange dauern. Methodisch nachteilig kommt bei dieser Art des Probierens hinzu, daß nur bei den relativ wenigen Fixsudokus vielleicht schon nach einigen Tausend Tests ein passendes f gefunden wird, während es sonst gerade bei den vielen anderen *neutralen* Sudokus mit negativem Ausgang jeweils "bis zum bitteren Ende" geht.

Aufgrund der jetzt folgenden Sätze über Fixsudokus wird es umgekehrt möglich sein, den Löwenanteil aller neutralen Sudokus ganz rasch als solche zu erkennen und auch sonst immer ziemlich schnell zu entscheiden, ob solch A ein Fixsudoku ist. Meine Analyse beginnt mit der Frage, welche $f \neq id$ mit $f(A) = A$ denn überhaupt in Betracht kommen. Angesichts der in Abschnitt 1.2 genannten Erzeuger von G sei $f \in G$ dazu mit (R/S)-Anteil h und lokalem (r/s)-Anteil c in einer der Formen $f = c{\cdot}h$ oder $f = c{\cdot}h{\cdot}t$ dargestellt – in letzterer, falls f zuerst transponiert.

Im Falle $f = c{\cdot}h \neq id$ mit $f(A) = A$ ist $h = id$ unmöglich, weil sonst die lokalen Permutationen alle Blöcke $A_{i,j}$ von A nicht ändern dürften, also auch $c = id$ wäre. Genauer müßte h mindestens Ordnung 3 haben, also R oder S enthalten, denn sonst ließe h immer mindestens einen der Blöcke $A_{i,j}$ fest, bei $h = R_1S_1$ oder $h = S_1$ z. B. $A_{1,1}$, was jedoch lokale Permutationen der Zeilen $\mu \in \{3, 4, 5\}$ ausschlösse, und das würde wegen $f(A) = A$ und der Wirkung von S_1 dann für die zugeordneten waagerechten Dreier-Mengen $w_0(A_{1,0}) = w_0(A_{1,2})$ implizieren – entgegen der Sudoku-Bedingung, daß die Zeile $a_{3,*}$ jede Ziffer nur einmal enthalten soll. Für andere Kombinationen $h = R_iS_j$, $h = R_i$ oder $h = S_i$ schließt man analog. Noch weitere Einschränkung der in Betracht kommenden (R/S)-Anteile h liefert das nächste Lemma, und daß die Transposition t in dieser Hinsicht ganz auszuschließen ist.

Lemma 4.2. *Jedes irgendein Sudoku A fixierende $f \in G \setminus \{id\}$ ist stets von der Form $f = c{\cdot}h \in G_0$ mit einem (R/S)-Anteil $h \neq id$ in der von R und S erzeugten 9-elementigen Untergruppe und einem jeweils passenden lokalen (r/s)-Anteil c.*

Beweis. Für den Fall $f \in G_0$ (also ohne t) wurde zuvor schon gezeigt, daß solch $f = c{\cdot}h$ im (R/S)-Anteil R oder S enthält. Stünde darin außerdem ein (einzelnes) S_i oder R_i, dann wäre auch $f^3 = \hat{c}{\cdot}h^3 = \hat{c}{\cdot}S_i$ bzw. $= \hat{c}{\cdot}R_i$ ein A fixierendes $\hat{f}$, mit variiertem (r/s)-Anteil $\hat{c}$, aber ohne R und S – Widerspruch!

Im anderen Falle $f = c{\cdot}h{\cdot}t$ mit $f(A) = c{\cdot}h(A^t) = A$ wird zunächst gezeigt, daß die "kleinen" 10 Unterfälle $h = id$ und $h = R_i,\ S_i,\ R_iS_i$ für $i = 0, 1, 2$ unmöglich sind, weil sonst einer der Diagonalblöcke dabei ungeändert bliebe, z. B. $B = A_{1,1}$ für $i = 1$ (oder $h = id$). Das aber würde (mit c' für die auf $B^t = A^t_{1,1}$ eingeschränkten lokalen Permutationen) für die waagerechten Dreiermengen (invariant unter c')

$$\{w_l(B) : l = 0, 1, 2\} = \{w_l(c'(B^t)) : l = 0, 1, 2\} = \{w_l(B^t) : l = 0, 1, 2\} \tag{4.1}$$

implizieren, was jedoch für keinen Sudoku-Block B gelten kann. So bleiben noch $26 = \#(S_3 \times S_3) - 10$ andere Unterfälle, von denen ich als nächste die 12 Fälle $h = RS_j,\ RRS_j,\ R_iS,\ R_iSS$ $(i, j = 0, 1, 2)$ diskutiere. Bei diesen wäre auch $\tilde{f} = f^2$

ein A fixierendes $\tilde{f} = c{\cdot}h{\cdot}t{\cdot}c{\cdot}h{\cdot}t = \tilde{c}{\cdot}h{\cdot}h^* \in G_0$ mit oft modifiziertem $\tilde{c}$ vom Typ (r/s) und $h^* = t{\cdot}h{\cdot}t$, worin die R's und S's aus h durch analoge S's bzw. R's ersetzt sind. Bei $h = RS_1$ würde so z. B. $h{\cdot}h^* = RR_1{\cdot}S_1S = R_0{\cdot}S_2$ entstehen, was nach den Überlegungen vor Lemma 4.2 aber zu keinem Fixsudoku paßt, und ebenso sind auch die anderen dieser 12 Unterfälle auszuschließen.

Nun zu den zwei Unterfällen $h = RRS$, RSS, die wegen $h{\cdot}h^* = id$ eine andere Art Beweis erfordern: Bei $f = c{\cdot}RRS{\cdot}t$ z. B. impliziert $A = f(A) = c{\cdot}RRS(A^t)$ für den Block $B = A_{0,1}$ (mit passendem c') die Gleichung $B = c'(B^t)$, die aber mittels (4.1) zum Widerspruch führt, sonst bei $f = c{\cdot}RSS{\cdot}t$ analog mit $B = A_{0,2}$.

Bei den jetzt noch übrigen 12 Unterfällen $h = R,\ S,\ RR,\ SS,\ RS,\ RRSS$ und $h = R_iS_j$ für $i \neq j$ betrachte ich wieder f^3 als[5] ein A fixierendes $\hat{f}$, hier von der Form $\hat{f} = c{\cdot}h{\cdot}t{\cdot}c{\cdot}h{\cdot}t{\cdot}c{\cdot}h{\cdot}t = \hat{c}{\cdot}h'{\cdot}t$ mit neuem $\hat{c}$ und $h' = h{\cdot}h^*{\cdot}h$, mit $h^* = t{\cdot}h{\cdot}t$. Bei den ersten 4 dieser Fälle ergäbe das jeweils ein $h' \in \{RRS, RSS\}$, die zuvor schon als "unfixierend" erkannt wurden, mit $h = RS$ oder $h = RRSS$ wäre $h' = id$, und bei den anderen 6 Fällen jeweils $h' = R_kS_k$ mit $k = 3 - i - j$, wie man z. B. bei $h = R_0S_1$ mit $h^* = R_1S_0$ an $h' = R_0R_1R_0 \cdot S_1S_0S_1 = R_2S_2$ erkennt. All diese Kombinationen $\hat{c}{\cdot}h'{\cdot}t$ gehören aber zu den "kleinen" Unterfällen, deren Unmöglichkeit sich schon an (4.1) zeigte. □

Als erste Folgerung aus Lemma 4.2 ist zu vermerken, daß im Stabilisator eines Fixsudokus A jedes $f \neq id$ die Ordnung 3 hat und daß solch Stabilisator jeweils kanonisch isomorph zur Gruppe H_A der (R/S)-Anteile h seiner Elemente ist. Wenn H_A die volle von R, S erzeugte Untergruppe von G_0 ist, sei A auch *Superfix* genannt, sonst ist H_A eine der in dieser vollen Gruppe der Ordnung 9 von R, S, RS oder RSS erzeugten Untergruppen der Ordnung 3. Entsprechend nenne ich A dann auch ein Fixsudoku *vom Typ* $[R], [S], [RS], [RSS]$ – und die Superfixe gelten als von jedem Typ. Als Beispiel sei hier nochmals jenes U erwähnt, das mit seinem am Ende von 4.1 genannten $f_1 = rr'r''{\cdot}SS$ ein Fixsudoku vom Typ $[S]$ ist, vermöge analog gebautem $f_2 = ss's''{\cdot}R$ mit $f_2(U) = U$ aber zugleich auch vom Typ $[R]$ und somit ein Superfix ist, dessen zu H_U isomorpher Stabilisator die Ordnung 9 hat. Auch das zugehörige System $vw(U)$ ist bei diesem U von bemerkenswerter Regularität: Es kommen darin nur je drei verschiedene v- und w-Werte mit den maximal möglichen Multiplizitäten 9 vor, also mit $27 = 9 + 9 + 9$ für die Partitionen p_U und q_U.

In Vorbereitung auf den gleich folgenden Hauptsatz dieses Abschnitts ist die im Beispiel bei f_1, f_2 benutzte adhoc Schreibweise für die (r/s)-Anteile c solcher $f = c{\cdot}h$ aus Lemma 4.2 nun zu systematisieren, indem jedes solche c als ein 6-Tupel $c = (b_0, b_1, b_2;\, c_0, c_1, c_2) \in S_3^3 \times S_3^3$ expliziert wird, worin (für $i, j, l < 3$) diese b_i jeweils eine lokale Permutation der Zeilen $\mu = 3i + l$ und die c_j eine Permutation der Spalten $\nu = 3j + l$ bezeichnen – und all diese Faktoren kommutieren miteinander.

Ferner seien (für die Diskussion eines festen Sudokus A) bei den Dreier-Mengen aus (1.3) folgende Kompaktschreibweisen verabredet: $v_{i,j}$ bezeichne das Mengen-Tripel $(v_0(A_{i,j}),\, v_1(A_{i,j}),\, v_2(A_{i,j})) \in D^3$, und $w_{i,j}$ in analoger Weise das Tripel $(w_0(A_{i,j}),\, w_1(A_{i,j}),\, w_2(A_{i,j})) \in D^3$, der Geometrie von $A_{i,j}$ entsprechend auch als 'Spaltenvektor' deutbar. Weil $A_{i,j}$ jede der Ziffern $1, \ldots, 9$ genau einmal enthält, sind die Komponenten solch eines Tripels jeweils paarweise disjunkt, und $v_{i,j}, w_{i,j}$ zusammen bestimmen die Ziffernpositionen in $A_{i,j}$ auf eindeutige Weise.

[5] diese Argumentation mit f^3 verdanke ich einem Hinweis von W. Jehne

Nach diesen Vorbereitungen soll nun der erste Fall $f = c{\cdot}R$ aus Lemma 4.2 analysiert werden. Mit $c = (b_0, b_1, b_2;\, c_0, c_1, c_2)$ folgt aus $c{\cdot}R(A) = A$ in Blockdarstellung

$$(4.2)\quad c{\cdot}R(A) = \begin{pmatrix} b_0c_0(A_{2,0}) & b_0c_1(A_{2,1}) & b_0c_2(A_{2,2}) \\ b_1c_0(A_{0,0}) & b_1c_1(A_{0,1}) & b_1c_2(A_{0,2}) \\ b_2c_0(A_{1,0}) & b_2c_1(A_{1,1}) & b_2c_2(A_{1,2}) \end{pmatrix} = \begin{pmatrix} A_{0,0} & A_{0,1} & A_{0,2} \\ A_{1,0} & A_{1,1} & A_{1,2} \\ A_{2,0} & A_{2,1} & A_{2,2} \end{pmatrix}$$

äquivalent zu 9 Gleichungen $b_0c_0(A_{2,0}) = A_{0,0}\,,\ b_1c_0(A_{0,0}) = A_{1,0}\,, \ldots\,,$ die wir nun in die Sprache der $v_{i,j}, w_{i,j}$ übersetzen. Dabei ist zu beachten, daß die Permutationen b_i keine Änderung der v-Werte und die c_j keine Änderung der w-Werte bewirken. So erweist sich (4.2) als äquivalent zu den 18 Gleichungen

$$[R]\qquad \begin{array}{lll} c_j\, v_{2,j} = v_{0,j}, & c_j\, v_{0,j} = v_{1,j}, & c_j\, v_{1,j} = v_{2,j}, \\ b_0\, w_{2,j} = w_{0,j}, & b_1\, w_{0,j} = w_{1,j}, & b_2\, w_{1,j} = w_{2,j} \end{array} \qquad \text{für } j = 0, 1, 2,$$

worin die b_i und c_j nun als Permutationen jener Tripel aus D^3 zu verstehen sind. Weil deren Komponenten paarweise disjunkt sind, bestimmt jede einzelne dieser Gleichungen – sofern überhaupt lösbar – das betroffene c_j bzw. b_i schon eindeutig, und $c{\cdot}R(A) = A$ gilt genau dann, wenn alle 18 Gleichungen *kompatibel* lösbar sind. Weil dann jede Komponente d dieser Tripel auch in den anderen Blockzeilen (bei jeweils gleichem Index j) vorkommt, folgt als einfaches *notwendiges Kriterium* für Fixsudokus A dieser Art so ganz nebenbei, daß alle Werte ihrer Multiplizitäten $v_A : D \to \mathbb{N}$ und $w_A : D \to \mathbb{N}$ Vielfache von 3 sein müssen.

Intuitiv ist klar, daß diese Art der Analyse auch für die anderen drei Fälle aus Lemma 4.2 zu entsprechenden Systemen von je 18 Gleichungen führen wird. Bevor die Details dazu ausgebreitet werden, sei hier zunächst das Hauptresultat formuliert.

Satz 4.3. *A ist genau dann ein Fixsudoku, wenn die zugehörigen Tripel $v_{i,j}$, $w_{i,j}$ von Dreier-Mengen hinsichtlich lokaler Permutationen $c = (b_0, b_1, b_2;\, c_0, c_1, c_2)$ eine kompatible Lösung von mindestens einem der Systeme $[R]$, $[S]$, $[RS]$, $[RSS]$ von je 18 kombinatorischen Gleichungen zulassen, die dann eindeutig ein $f = c{\cdot}h \in G_0$ mit $h \in \{R,\, S,\, RS,\, RSS\}$ bestimmt, das $f(A) = A$ erfüllt. – Notwendigerweise sind bei einem Fixsudoku A alle Multiplizitäten $v_A(d)$, $w_A(d)$ Vielfache von 3.*

Ähnlich einfach ist die Herleitung im zweiten Fall $f = c{\cdot}S$, mit Blockgleichung

$$(4.3)\quad c{\cdot}S(A) = \begin{pmatrix} b_0c_0(A_{0,2}) & b_0c_1(A_{0,0}) & b_0c_2(A_{0,1}) \\ b_1c_0(A_{1,2}) & b_1c_1(A_{1,0}) & b_1c_2(A_{1,1}) \\ b_2c_0(A_{2,2}) & b_2c_1(A_{2,0}) & b_2c_2(A_{2,1}) \end{pmatrix} = \begin{pmatrix} A_{0,0} & A_{0,1} & A_{0,2} \\ A_{1,0} & A_{1,1} & A_{1,2} \\ A_{2,0} & A_{2,1} & A_{2,2} \end{pmatrix}$$

äquivalent zu 9 Gleichungen $b_0c_0(A_{0,2}) = A_{0,0}\,,\ b_0c_1(A_{0,0}) = A_{0,1}\,, \ldots,$ die sich nach Übersetzung in die $v_{i,j}, w_{i,j}$ als äquivalent zu den 18 Gleichungen

$$[S]\qquad \begin{array}{lll} c_0\, v_{i,2} = v_{i,0}\,, & c_1\, v_{i,0} = v_{i,1}, & c_2\, v_{i,1} = v_{i,2}\,, \\ b_i\, w_{i,2} = w_{i,0}, & b_i\, w_{i,0} = w_{i,1}, & b_i\, w_{i,1} = w_{i,2} \end{array} \qquad \text{für } i = 0, 1, 2$$

erweisen. Wenn diese kompatibel lösbar sind, dann wird jede Komponente d dieser Tripel auch in den anderen Blockspalten (bei jeweils gleichem Index i) vorkommen, was wieder die Teilbarkeit der Multiplizitäten durch 3 impliziert.

Der dritte Fall $h = RS$ bedeutet zyklischen Shift in der Blockdiagonalen, also

$$(4.4)\quad c{\cdot}RS(A) = \begin{pmatrix} b_0c_0(A_{2,2}) & b_0c_1(A_{2,0}) & b_0c_2(A_{2,1}) \\ b_1c_0(A_{0,2}) & b_1c_1(A_{0,0}) & b_1c_2(A_{0,1}) \\ b_2c_0(A_{1,2}) & b_2c_1(A_{1,0}) & b_2c_2(A_{1,1}) \end{pmatrix} = \begin{pmatrix} A_{0,0} & A_{0,1} & A_{0,2} \\ A_{1,0} & A_{1,1} & A_{1,2} \\ A_{2,0} & A_{2,1} & A_{2,2} \end{pmatrix}$$

äquivalent zu 9 Gleichungen $b_0c_0(A_{2,2}) = A_{0,0}$, $b_0c_1(A_{2,0}) = A_{0,1}$, ... und nach Übersetzung in die $v_{i,j}, w_{i,j}$ dann auch äquivalent zu den 18 Gleichungen

$$[RS] \qquad c_j\, v_{i-1,j-1} = v_{i,j}, \quad b_i\, w_{i-1,j-1} = w_{i,j} \qquad \text{für } i = 0,1,2,\ j = 0,1,2,$$

jetzt mit der Konvention, daß Indexwerte -1 modulo 3 als 2 zu deuten sind. Bei kompatibler Lösbarkeit werden Komponenten d dieser Tripel diagonal zyklisch nach Südosten weitergereicht, also folgen dann auch hier $3|v_A(d)$ und $3|w_A(d)$.

Im vierten Fall $h = RSS$ schließlich folgt ganz ähnlich

$$(4.5)\quad c{\cdot}RSS(A) = \begin{pmatrix} b_0c_0(A_{2,1}) & b_0c_1(A_{2,2}) & b_0c_2(A_{2,0}) \\ b_1c_0(A_{0,1}) & b_1c_1(A_{0,2}) & b_1c_2(A_{0,0}) \\ b_2c_0(A_{1,1}) & b_2c_1(A_{1,2}) & b_2c_2(A_{1,0}) \end{pmatrix} = \begin{pmatrix} A_{0,0} & A_{0,1} & A_{0,2} \\ A_{1,0} & A_{1,1} & A_{1,2} \\ A_{2,0} & A_{2,1} & A_{2,2} \end{pmatrix}$$

äquivalent zu 9 Gleichungen $\cdots$ und dann auch äquivalent zu den 18 Gleichungen

$$[RSS] \qquad c_j\, v_{i-1,j+1} = v_{i,j}, \quad b_i\, w_{i-1,j+1} = w_{i,j} \qquad \text{für } i = 0,1,2,\ j = 0,1,2,$$

hier nun so, daß Indexwerte 3 als 0 zu deuten sind. Kompatible Lösbarkeit sichert (jetzt antidiagonal zyklisch shiftend) wiederum $3|v_A(d)$ und $3|w_A(d)$. □

Bei *einfachen* Fixsudokus A mit H_A von der Ordnung 3 bedeutet das Lösen des jeweils passenden Gleichungssytems explizite Konstruktion des zu H_A isomorphen Stabilisators von A. Am Rande sei ferner angemerkt, daß bei Lösbarkeit *mehrerer* dieser Gleichungssysteme $[R], [S], [RS], [RSS]$, wie es beim Urbeispiel U der Fall ist, die Multiplizitäten durch Kombination zweier solcher 'Weiterreich-Muster' notwendigerweise Vielfache von 9 sind, was aber andererseits kein *hinreichendes* Kriterium für Superfixe ist, wie man an dem folgenden leicht variierten Beispiels U' sehen kann: Man erhält solch ein nur noch einfaches Fixsudoku U' aus U, indem man dort <u>nur</u> in den Spalten 6,7,8 die Ziffern 1,4,7 gegen 3,6,9 tauscht. Das damit entstehende U' ist dann immer noch fix unter jenem f_1, also vom Typ $[S]$, genügt aber keinem der anderen Systeme $[R]$, $[RS]$, $[RSS]$, während sich bei den Invarianten mit $vw(U') = vw(U)$ keinerlei Änderung zeigt.

4.3 Algorithmus zum Erkennen und Zählen möglicher Fixsudokus

Ein volles Sudoku A, von dem entschieden werden soll, ob es ein Fixsudoku ist, mag auf sehr verschiedene Weisen gegeben sein. Hier nehme ich an, daß A durch Lösen einer Sudoku-Rätselaufgabe (mittels meines Programms *INFER*) oder durch einen eher theoretisch motivierten Fortsetzungsprozeß (wie bei der Variante *CINF*, vgl. Abschnitt 2.2) schrittweise erzeugt wird. So kann dann oft schon vorher am "wachsenden" A erkennbar sein, daß daraus kein Fixsudoku werden kann, wenn es z. B. bei den $v_l(A_{i,j})$ (soweit bis dahin bestimmt) schon mehr als 9 verschiedene Werte d gibt, also nicht alle Multiplizitäten Vielfache von 3 werden können.

Sonst liegt A dann schließlich als Vollsudoku vor[6], zuerst allerdings noch in der redundanten Kodierung der am Anfang von 2.2 beschriebenen Muster von Kennworten $k_n(i,j)$, die beim fertigen Vollsudoku A dann allesamt zu reinen Potenzen von 2 'abgemagert' sind. Im ersten vorbereitenden Schritt des Zähl-Algorithmus *FCINF* werden diese Daten in das für die Überprüfung jener Gleichungssysteme benötigte System $vw(A)$ umgerechnet, nun in Gestalt der 54 Binärzahlen

$$\text{(4.6)} \qquad \begin{aligned} v_{i,j}^l &= \textstyle\sum_{n=1}^9 \{2^{n-1} : q_n(i,j) \cap \ 7 \cdot 8^l \neq 0\}, \\ w_{i,j}^l &= \textstyle\sum_{n=1}^9 \{2^{n-1} : q_n(i,j) \cap 73 \cdot 2^l \neq 0\}, \end{aligned} \qquad \text{für } \ l,i,j = 0,1,2,$$

die als Kodierungen jener 54 Dreier-Mengen zu lesen sind. Dabei dienen $7 = 1+2+4$ (um $3l$ bit verschoben) und $73 = 1+8+64$ (um l bit verschoben) als Masken beim bitweisen AND – hier als Mengenoperation $\cap$ geschrieben. – Solch eine Umrechnung kostet ungefähr 3 μs. Es ist übrigens bemerkenswert zu sehen, wie man auf diese Weise die 81 Worte des Musters zu diesen 54 Worten verdichtet; sogar zu 54 Bytes ginge das, denn in (4.6) würde Summation bis $n=8$ ja auch noch genügen – indem die 9 dann jeweils zum Zweier gehörte!

Im zweiten Schritt wird nun erst einmal das in Satz 4.3 genannte notwendige Kriterium überprüft. Mit einer kleinen Zählroutine (mod 3 rechnend) kostet das für die v's und die w's je nur etwa 1 μs, wird ja aber schon bei den v's fast immer negativ enden. So werden neutrale Sudokus meist nach etwa 4 bis 5 μs als solche erkannt. Sonst sind schließlich die vier Gleichungssysteme $[R], [S], [RS], [RSS]$ der Reihe nach auf kompatible Lösbarkeit hin zu untersuchen, was dann nur noch programmiertechnische Routine erfordert:

Mit den Zahlen aus (4.6) sind die Tripel $(v_{i,j}^0, v_{i,j}^1, v_{i,j}^2)$ und $(w_{i,j}^0, w_{i,j}^1, w_{i,j}^2)$ als Kodierungen der in jenen Gleichungssystemen vorkommenden $v_{i,j}$ und $w_{i,j}$ gegeben. Ein kleines Unterprogramm *PERM*, das übrigens allein mit Register-Operationen arbeitend ganz ohne Speicherzugriffe auskommt, bestimmt zu (in zwei 32-bit Worten) gegebenen Tripeln $x = (x^0, x^1, x^2)$ und $y = (y^0, y^1, y^2)$ jeweils *die* Permutation $p \in S_3$, mit der $p\,x = y$ gilt – in naheliegender Weise durch eine Zahl $k(p) < 6$ kodiert – oder liefert ein Fehlersignal, wenn es solch p nicht gibt. Diese Routine *PERM* ist maximal 18-fach aufzurufen, um je solch ein Gleichungssystem zu prüfen, zweckmäßiger Weise in passender Reihenfolge nacheinander je dreifach für die gesuchten b_0, b_1, b_2, c_0, c_1, c_2, damit bei etwaigen Inkompatibilitäten (wenn nicht ohnehin solch Fehlersignal kommt) bald abgebrochen werden kann. Im (zeitlich) ungünstigsten Falle, daß A ein Fixsudoku ist, bei dem die geprüften 18 Gleichungen kompatibel lösbar sind, kostet diese Prüfung gut eine Mikrosekunde, der gesamte Test für alle vier Systeme also höchstens 5 μs.

Es erscheint schwer abschätzbar, ob drei vergebliche Tests vor dem vierten Test (des dann lösbaren Systems) tatsächlich so viel Zeit kosten können. Jedenfalls ist das Erkennen von (vollen) Fixsudokus mit dieser Methode immer in einer Rechenzeit von weniger als 10 μs möglich und damit viel schneller als etwa das Lösen selbst einfacher Sudoku-Rätsel.

[6] jetzt von der Shortcut Situation des Lemmas 2.1 absehend, die hier bei der Suche nach Fixsudokus voll auszureizen ist – es sei denn, man könnte zeigen, daß solche Stagnation von *CINF* bei einem Muster K mit 77 fixierten Ziffern Fortsetzbarkeit zu einem Fixsudoku ausschließt.

4.4 Zählergebnisse

Zuerst geht es in diesem Abschnitt darum, die Größe der Menge X_f aller Fixsudokus zu bestimmen. Lemma 4.1 gekoppelt mit (2.4) reduziert das Zählen aller $A \in X_f$ sofort auf die Anzahlen $N_f(u,v) = \#(\uparrow P(u,v)\uparrow \cap X_f)$ und vermöge der Anzahl-Invarianz $N_f(w,y) = N_f(u,v)$ für $(w,y) \sim (u,v)$ auf die nur noch $k^* = 237\,083$ Paare (u,v) in *TPFUV,* so daß wir, über solche (u,v) summierend, für die Anzahl N_f aller (e,u,v)-normierten Fixsudokus (vgl. 2.1) analog zu (2.11) die Formel

$$(4.7) \qquad N_f = \textstyle\sum_{(u,v)} m(u,v) \cdot N_f(u,v) \quad \text{und} \quad \#X_f = 9! \cdot 6^4 \cdot N_f$$

erhalten. Wie am Anfang von Abschnitt 4.3 schon angedeutet, kann *FCINF* bei Anwendung auf ein blockdiagonales Präsudoku $P(u,v)$ bei vielen der daraus erwachsenden größeren Präsudokus (die sich oft noch zu vielen verschiedenen Vollsudokus fortsetzen ließen) den Suchprozeß schon in relativ frühem Stadium abbrechen, weil z. B. ein 10-tes vertikales Tripel vorkommt, womit die nach Satz 4.3 für Fixsudokus notwendige Bedingung, daß alle Multiplizitäten durch 3 teilbar sind, bei den 27 Tripeln daraus entstehender Vollsudokus sicher nie gelten würde. Der damit erzielte Spareffekt ist enorm: Über sämtliche Aufrufe *FCINF* mit $P(u,v)$ für alle 237083 Paare (u,v) gezählt hatte mein globales Zählprogramm *FCNT* wegen derartig vorzeitigen Abbrechens nur noch bei $h = 2\,375835$ Vollsudokus A weitere Kriterien nach Satz 4.3 zu prüfen, im Durchschnitt also nur etwa 10 je Klasse $(u,v)^{\tilde{}}$ – statt der 139707 als dem Mittelwert der $N(u,v)$ – und all das in nur 12 Minuten!

Von jenen h finalen A's fielen bei dem vollen Test auf 3-Teilbarkeit der Multiplizitäten dann weiter noch 2 363846 (ca. 99.5 %) dieser A's durch, und von den übrigen 11989 A's erfüllten 8514 keines der Gleichungssysteme $[R], [S], [RS], [RSS]$ aus Satz 4.3. So lieferte diese Suche schließlich nur noch die geringe Zahl von 3475 Fixsudokus zu einigen wenigen Paaren (u,v) in *TPFUV*, und zwar in dieser Verteilung: Für das minimale Paar (e,e), $P(e,e) = E \setminus E \setminus E$ kodierend, gibt es die maximale Zahl $N(e,e) = 283576$ zugehöriger Vollsudokus, unter denen aber nur $N_f(e,e) = 64$ Fixsudokus, die ich unten noch weiter diskutiere. Weiters gibt es zwei speziell für den Typ $[RSS]$ wichtige Paare, $(u_1,v_1) = (57643210,\ 65743210)$ mit $N_f(u_1,v_1) = 26$ und $(u_2,v_2) = (43671520,\ 43172560)$ mit $N_f(u_2,v_2) = 17$, – beide mit gleicher relativ geringer Klassengröße $m(u_1,v_1) = m(u_2,v_2) = 12$ — sowie 19 Paare mit $N_f(u,v) = 16$ (und stark varianten $m(u,v)$-Werten, s. Tabelle 4.4) und weitere 383 Paare mit $N_f(u,v) = 8$, nebst $N_f(u,v) = 0$ für die übrigen 236678 Paare. Über jene 405 "Fixsudoku-haltigen" (u,v) nach (4.7) summierend folgt so

$$(4.8) \qquad N_f = 1\,161284 = 4 \cdot 41 \cdot 73 \cdot 97$$

und damit auch $\#X_f = 546\ 143132\ 344320$ als Gesamtzahl aller Fixsudokus, im Vergleich zu (2.12) mit relativer Häufigkeit ("Seltenheit") $N_f/N_0 = 0.000000\ 081$.

Die 383 Paare mit $N_f(u,v) = 8$ haben zugehörige $m(u,v)$ in diesen Häufigkeiten:

275-fach 432, 92-fach 216, 11-fach 72, 36 und 144 je 2-fach, 108 nur 1-fach.

Die Summe dieser 383 Multiplizitäten $m(u,v)$ ist 139932, zu den damit gezählten Blockdiagonalen (w,y) gibt es also insgesamt $8 \cdot 139932 = 1\ 119456$ Fixsudokus, das sind 96.4 Prozent von N_f in (4.8).

Tabelle 4.4. Die 19 Paare (u, v) mit $N_f(u, v) = 16$, nebst ihren $m(u, v)$-Werten. Deren Summe ist 2578, mit 16 multipliziert etwa 3.55 Prozent zu N_f beitragend.

u	v	$m(u, v)$
76543210	67543210	54
	57643210	36
	67542310	108
	57632410	12
57643210	67542310	216
	76532410	24
	67532410	216
	57632410	72
57632410	56742301	4
67543210	75643210	108
	76542310	216
	57642310	216
	76532410	432
	57632410	216
	76543201	36
	75643201	216
67542310	75632410	216
	56732410	72
	57643201	108

Weil mein Zählprogramms *FCNT* pro Durchlauf jeweils nur ca. 12 Minuten braucht, konnte es leicht, mit allerlei kleinen Zusatzabfragen variiert, mehrfach wiederholt werden, um genaue Anzahl-Informationen über das Vorkommen der verschiedenen Typen zu gewinnen und kleine Klassen spezieller Fixsudokus explizit zu tabellieren. Einige der so erzielten Befunde seien hier jetzt noch kurz dokumentiert.

Fast alle der 3475 beim Test als Fixsudokus erkannten A genügen jeweils nur einem der vier Gleichungssysteme aus Satz 4.3, und zwar in diesen Anzahlen:

(4.9) 1452 nur $[R]$, 1980 nur $[S]$, 36 nur $[RS]$, und 3 nur $[RSS]$.

Genauer findet man zwei dieser drei A's vom Typ $[RSS]$ in $\upharpoonleft P(u_1, v_1)\upharpoonright$ und das dritte in $\upharpoonleft P(u_2, v_2)\upharpoonright$, jeweils mit $m(u_1, v_1) = m(u_2, v_2) = 12$ multipliziert sind das also nur 36 der N_f normierten Fixsudokus aus (4.8). Die gleiche Anzahl erscheint in (4.9) für den Typ $[RS]$, und das ist in der Tat auch schon deren Gesamtzahl, denn all diese A's vom Typ $[RS]$ gehören zu $\upharpoonleft P(e, e)\upharpoonright$ mit Multiplizität $m(e, e) = 1$. Unter den übrigen 28 der $N_f(e, e) = 64$ Fixsudokus mit normierter Blockdiagonale $P(e, e) = E \diagdown E \diagdown E$ gibt es 12 einfache vom Typ $[R]$, 12 einfache vom Typ $[S]$, und genau 4 (als die einzigen, nur hier in $\upharpoonleft P(e, e)\upharpoonright$ existenten) blockdiagonal normierten *Superfixe* U_0, U_1, U_2, U_3, deren explizite Liste am Ende dieses Abschnitts folgt.

Die arg irregulär erscheinenden Einträge in Tabelle 4.4 wie auch die befremdliche Verschiedenheit der Anzahlen 1452 und 1980 in (4.9) resultieren zumindest teilweise aus der Willkür bei der am Anfang von Abschnitt 2.3 getroffenen Wahl der lexikographisch minimalen Paare (u, v) als Repräsentanten jener Äquivalenzklassen. Natürlich muß es einfache Fixsudokus vom Typ $[R]$ wie vom Typ $[S]$ in gleicher Zahl N' geben, denn mit Fixsudoku $A = cR(A)$ vom Typ $[R]$ ist dessen transponiertes $A^t = t{\cdot}cR(A) = \tilde{c}S(A^t)$ vom Typ $[S]$, und umgekehrt.

Anspruchsvoller hingegen ist es, jene zuvor nur empirisch belegten Anzahlen 4, 36 und N' theoretisch zu deuten. In dieser Richtung ist Fritz Ostermann (vgl. [FO]) eine vollständige kombinatorische Analyse der Fixsudokus gelungen.

Darüber hinausgehend wäre dann natürlich auch noch ein entsprechend gründliches Verständnis der Fixpunkte und Bahnlängen bzgl. der totalen Mischgruppe G^* aus 2.1 wünschenswert, was vermutlich aber noch andere Methoden erfordert.

Zum Schluß jetzt noch ein Blick auf die 4 normierten Superfixe. Das zuerst gezeigte U_0 ist die normierte Version des Urbeispiels U von Seite 2, hier durch die Normierung nun aber in schönerer Gestalt, wie auch bei den anderen U_j mit E und jeweils passend gewählten 3×3 Blöcken B, C von nebenstehender Form.

$$U_j = \left|\begin{array}{ccc} E & B & C \\ C & E & B \\ B & C & E \end{array}\right|$$

$$U_0 = \begin{array}{|ccc|ccc|ccc|} \hline 1&2&3&5&6&4&9&7&8 \\ 4&5&6&8&9&7&3&1&2 \\ 7&8&9&2&3&1&6&4&5 \\ \hline 9&7&8&1&2&3&5&6&4 \\ 3&1&2&4&5&6&8&9&7 \\ 6&4&5&7&8&9&2&3&1 \\ \hline 5&6&4&9&7&8&1&2&3 \\ 8&9&7&3&1&2&4&5&6 \\ 2&3&1&6&4&5&7&8&9 \\ \hline \end{array} \qquad U_2 = \begin{array}{|ccc|ccc|ccc|} \hline 1&2&3&9&7&8&5&6&4 \\ 4&5&6&3&1&2&8&9&7 \\ 7&8&9&6&4&5&2&3&1 \\ \hline 5&6&4&1&2&3&9&7&8 \\ 8&9&7&4&5&6&3&1&2 \\ 2&3&1&7&8&9&6&4&5 \\ \hline 9&7&8&5&6&4&1&2&3 \\ 3&1&2&8&9&7&4&5&6 \\ 6&4&5&2&3&1&7&8&9 \\ \hline \end{array}$$

$$U_1 = \begin{array}{|ccc|ccc|ccc|} \hline 1&2&3&8&9&7&6&4&5 \\ 4&5&6&2&3&1&9&7&8 \\ 7&8&9&5&6&4&3&1&2 \\ \hline 6&4&5&1&2&3&8&9&7 \\ 9&7&8&4&5&6&2&3&1 \\ 3&1&2&7&8&9&5&6&4 \\ \hline 8&9&7&6&4&5&1&2&3 \\ 2&3&1&9&7&8&4&5&6 \\ 5&6&4&3&1&2&7&8&9 \\ \hline \end{array} \qquad U_3 = \begin{array}{|ccc|ccc|ccc|} \hline 1&2&3&6&4&5&8&9&7 \\ 4&5&6&9&7&8&2&3&1 \\ 7&8&9&3&1&2&5&6&4 \\ \hline 8&9&7&1&2&3&6&4&5 \\ 2&3&1&4&5&6&9&7&8 \\ 5&6&4&7&8&9&3&1&2 \\ \hline 6&4&5&8&9&7&1&2&3 \\ 9&7&8&2&3&1&4&5&6 \\ 3&1&2&5&6&4&7&8&9 \\ \hline \end{array}$$

Literatur

[FJ5] B. Felgenhauer and F. Jarvis, *Enumerating possible Sudoku grids.* June 2005, 7 pages, — `http://www.afjarvis.staff.shef.ac.uk/sudoku/sudoku.pdf`

[Hal] P. R. Halmos, *Naive Mengenlehre.* Vandenhoeck & Ruprecht, Göttingen 1968. Nach der Originalausgabe *Naive Set Theory* (1960), Übersetzung aus dem Amerikanischen von M. Armbrust und F. Ostermann.

[J08] W. Jehne, *Zur mathematischen Theorie der Sudokus – Ein Entwurf.* Manuskript, 2+80 Seiten, Köln 2008 .

[FO] F. Ostermann, *Die Fixsudokus der Sudokugruppe.* Manuskript, 15 Seiten, Köln, September 2010, demnächst via `http://www.f-ostermann.de`

[Sc1] A. Schönhage, *Homepage,* `http://www.informatik.uni-bonn.de/~schoe/`

[Sc2] A. Schönhage, *SUDOC2 – Grundlagen zum Sudoku-Programm.* Manuskript, 8 Seiten, Januar 2007, abrufbar via Link `Sudoku` auf [Sc1].

[SGV] A. Schönhage, A. F. W. Grotefeld, E. Vetter. *Fast Algorithms — A Multitape Turing Machine Implementation.* BI Wissenschaftsverlag, Mannheim, 1994. — Neuere Informationen zu *TP* via Link `Turing Processing` auf [Sc1].