How Do YOU Solve Sudoku? A Group-theoretic Approach to Human Solving Strategies

Elizabeth A. Arnold [1]
James Madison University
Harrisonburg, VA 22807

Harrison C. Chapman
Bowdoin College
Brunswick, ME 04011

Malcolm E. Rupert
Western Washington University
Bellingham, WA 98225

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

Sudoku Puzzle P

How would you solve this Sudoku puzzle? Unless you have been living in a cave for the past 10 years, you are probably familiar with this popular number game. The goal is to fill in the numbers 1 through 9 in such a way that there is exactly one of each digit in each row, column and 3×3 block. Like most other humans, you probably look at the first row (or column or 3×3 block) and say, "Let's see.... there is no 3 in the first row. The third,

[1] Corresponding author: arnoldea@math.jmu.edu

sixth and seventh cell are empty. But the third column already has a 3, and the top right block already contains a 3. Therefore, the 3 must go in the sixth cell." We call this a Human Solving Strategy (HSS for short). This particular strategy has a name, called the "Hidden Single" Strategy. (For more information on solving strategies see [8].) Next, you may look at the cell in the second row, second column. This cell can't contain 5, 6 or 8 since these numbers are already in the second row. It can't contain 3 or 9 since these are already in the second column. And it can't contain 1, 2, or 7 since these numbers are already in the first block. Therefore, the only number this cell can contain is 4. This strategy is called the "Naked Single" strategy. If the Sudoku is "easy" enough, you can continue on in a strategic manner using the Naked and Hidden Single Strategies and solve the whole puzzle. If the Sudoku is harder, you will need to use more complicated logical strategies. We will discuss a few of these in Section 2.

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

Sudoku Puzzle Q

Now try to solve Sudoku Puzzle Q above. You may look at the last column and notice that there is no 7 in it. The third, fifth and seventh cell are available. Bt the third row already has a 7 and the bottom right block already has a 7. Therefore, the 7 must go in the fifth cell. Déjà vu? Do you get the feeling that you have already solved this puzzle? Well, in fact, you have! Sudoku Q is "essentially the same" puzzle as Sudoku Puzzle P that you already solved. It the same puzzle, just rotated 90° clockwise with the numbers relabelled by the permutation $1 \to 5, 2 \to 6, 3 \to 7, 4 \to 8, 5 \to 9, 6 \to 1, 7 \to 2, 8 \to 3, 9 \to 4$. In cycle notation, (159483726). In Section 1, we define precisely what we mean when we say two puzzles are "essentially the same", and by the end of this article, we will explain that feeling of déjà vu!

The way a human solves a Sudoku is strikingly different than the way a computer would solve the same puzzle. The computer doesn't use human logic or human strategies. The most efficient computer algorithm for solving Sudoku is a brute force, depth first search. A popular such algorithm is Knuth's "Dancing Links" version of Algorithm X [6]. The computer starts with the possibilities of 1-9 in each cell. It first eliminates possibilities in the cells according to what given clues are in the same row, column or box. The computer then locates the first open cell and tries a number in this cell, eliminating this number from the row, column and box. If the number causes a conflict, then it is discarded. The values in the row, column and box are reset, and it tries another number. If the number does not cause a conflict, the algorithm continues to the next cell. If all numbers give a conflict for a given cell, the computer backtracks to find the last cell where there was a choice, and tries a different choice. While this method would take a human a very long time, a computer can do it very quickly and efficiently since there is only one rule at every step. If there were more complicated rules of logic, such as those a human uses, the computer would be much less efficient.

1 Essentially the Same Sudoku Puzzles

So, what do we mean when we say two Sudoku puzzles are "essentially the same"? First we distinguish between a Sudoku "puzzle" and Sudoku "board". A Sudoku board is a 9×9 grid filled with the digits 1-9 according to the rules of the game. A Sudoku puzzle is any subset of a Sudoku board. We say that two Sudoku boards, B and C are essentially the same if we can perform some action on board B that maps it to board C where if we performed that action on *any* given board, it would not create any inconsistencies in the rules of the game. As we already mentioned in the introduction, rotating a puzzle 90 degrees will give another puzzle that is essentially the same to the original. Also, permuting the numbers used for clues in one puzzle leads to another essentially the same puzzle. What other actions can we do to an arbitrary puzzle that preserve the rules of the game?

First a bit of notation. The three three-row strips are called **bands** and the three three-column strips are called **stacks**. We can interchange any two rows in a band and not create any inconsistencies. Notice, however, that we cannot interchange two rows that are not in the same band. For example, consider interchanging row 1 and row 9 in Sudoku Puzzle P in the introduction. Now we have two 4's in the lower right block and two 7's in the lower left block, creating an inconsistency in the puzzle. Similarly, we can interchange any two colums in a stack, but we cannot interchange two columns in different stacks. We can also interchange any two entire bands and any two entire stacks. Finally, notice that we can reflect the entire puzzle across any of the lines of symmetry of the square: horizontal, vertical or either diagonal.

The Dihedral Group of Order 8 is the group of symmetries of the square, denoted D_4. (See for example [4], Ch. 1). The group elements are “actions” on the square itself. We are going to consider a similar group of actions on Sudoku boards that include the actions mentioned above. We know that D_4 can be generated by just two elements, namely a rotation and a reflection. We define the element r to be rotation 90 degrees clockwise and t to be the reflection across the diagonal from upper left to bottom right. Now we need to represent the band and stack swaps. There are three bands and three stacks. Define $B12$ to be the swap of band 1 with band 2 and $B13$ to be the swap of band 1 with band 3. Notice that $B12 \circ B13$ swaps bands 2 and 3. $r^{-1} \circ B12 \circ r$ swaps stack 1 and 2. Similarly, we can swap the other stacks using a conjugation of a band swap and a rotation. Now, define $R12$ to be the swap of row 1 and row 2 and $R13$ to be the swap of row 1 and row 3. Together with the band swaps, we can now swap any two rows. Together with rotation, we can swap any two columns. Now all of the position symmetries can be generated by these 6 elements: $r, t, B12, B13, R12, R13$. Using GAP [5], we calculate the size of the group generated by these elements to be 3,359,232. This group is non-Abelian. Recall that we can also relabel the entries in the cells. There are 9! ways to do this, represented by the group, S_9. The position symmetries commute with the relabeling symmetries. So the entire Sudoku symmetry group is $\mathcal{G} = \langle r, t, B12, B13, R12, R13\rangle \times S_9$. This group has been discussed in [2] and [7]. We give a much smaller generating set in the following theorem.

Theorem 1.1. *The Sudoku Symmetry Group, $\mathcal{G}$ can be generated by the three elements: $B12, R12$ and r together with S_9. $\mathcal{G} = \langle B12, R12, r\rangle \times S_9$*

Proof. We just need to show that the generators above can be written as a combination of these three generators. $B13$ can be obtained by rotating twice, swapping Band 1 and Band 2 and then rotating back. $B13 = r^2 \circ B12 \circ r^2$. $R13$ can be obtained by swapping Band 1 and Band 2, rotate twice, swap Row 1 and Row 2, rotate twice again, swap Band 1 and 2 again. $R13 = B12 \circ r^2 \circ R12 \circ r^2 \circ B12$. t is much more complicated, and we use GAP to find $t = r \circ B12 \circ r^2 \circ B12 \circ r^2 \circ R12 \circ B12 \circ r^2 \circ R12 \circ B12 \circ r^2 \circ R12 \circ B12 \circ r^2 \circ R12 \circ B12 \circ r^2 \circ R12 \circ B12 \circ r^2 \circ R12 \circ B12 \circ R12 \circ r^2 \circ R12 \circ B12 \circ R12$. Alternatively, since one set of generators is contained in the other, we can use GAP to compute the size of both groups. We find that they have the same size, indicating that they generate the same group. $\square$

Now we can use $\mathcal{G}$ to define precisely what we mean by "essentially the same" Sudoku boards.

Definition 1.1. *Two Sudoku boards are* **essentially the same** *if they are in the same equivalence class induced by the the action of the group, $\mathcal{G}$ on the set of Sudoku boards. In other words, Sudoku boards B and C are essentially the same, if there is a Sudoku symmetry, $g \in \mathcal{G}$ such that g applied to B yields C.*

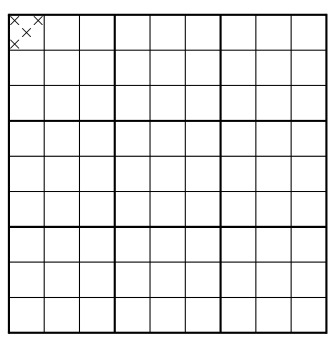
We would like to explain this feeling of déjà vu when solving two essentially the same puzzles. We will do this by defining another symmetry group relating to Sudoku: the group of solving symmetries. Before we do this, however, we talk about the idea of a "packet". A packet is basically a non-clue. Packets turn out to be a critical ingredient of solving symmetries.

2 Packets and Human Solving Symmetries

When you solve a Sudoku puzzle with pencil and paper, you will often look at a cell and decide which clues can NOT go in the cell. In the "Naked Single" strategy described in the introduction, we determine what clues cannot go in a cell and based on this, put the only possible remaining clue in the cell. We give these non-clues the name *packet*.

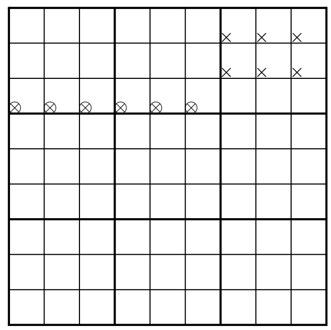
Definition 2.1. *A* **packet** *is a representation that a clue cannot be placed in a cell.*

For example, if we determine that 1, 3, 5 and 7 cannot go in the first cell, we represent this by putting an $\times$ in the first, third, fifth and seventh position in the first cell.



Packets in the first cell

We can expand on the Human Solving Strategy (HSS), "Naked Single", and discuss another HSS, "Naked Triple". In this Sudoku puzzle, we have packets in the 7-position in all but the last three cells in row 3. This implies that we must have packets in the 7-position in the top 6 cells of the top right block. In the diagram below, packets marked with $\otimes$ are the given packets and packets marked with $\times$ are implied packets.



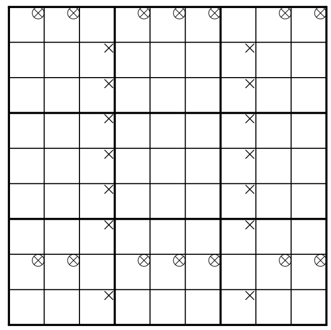
Packets representing a Naked Triple

Another HSS that takes advantage of packets is the "X-Wing" strategy. In this strategy, two rows have packets in all but two cells and these two cells are in the same column in the two rows. These given packets, represented by $\otimes$ in the figure below, imply packets in all but the two empty cells in the two columns. In the figure, the value 3 can only be in cell three or seven in both rows one and eight. Therefore, 3 cannot be in any of the other cells in column three and column seven. If three belongs in the third cell in the first row, then it must also be in the seventh cell in the eighth row. If it belongs in the seventh cell in the first row, then it must also be in the third cell in the eighth row. The possibilities for the 3 form an X, hence the name.

We will use the idea of packets and implied packets to algebraically define these human actions for solving Sudoku puzzles.

3 Solving Symmetries

Before defining solving symmetry, we first give a general context that will relate the group of solving symmetries to the Sudoku symmetry group, $\mathcal{G}$,



A Sudoku packet puzzle with an X-Wing

described in Section 1. There are 729 possible packets in a Sudoku board. We denote this set by $\mathcal{K}$. A Sudoku board can be thought of as a subset of 648 packets with eight packets in each cell. The packet that is missing representing the number that goes in the cell. A Sudoku puzzle is also a subset of $\mathcal{K}$. We call a subset of $\mathcal{K}$ a **packet set**. A packet set is an element of the power set of $\mathcal{K}$. Note that not all packet sets are Sudoku puzzles. There may be inconsistencies such as nine packets in a cell, or a packet in the 7-position of every cell in a given row. We call packet sets that are consistent and lead to a valid Sudoku board, **packet puzzles**. We summarize these definitions below.

Definition 3.1. *The set of all possible packets in a Sudoku board is denoted by $\mathcal{K}$. A subset of $\mathcal{K}$ is called a* **packet set***. The set of all packet sets is the power set of $\mathcal{K}$ denoted $\mathcal{P}(\mathcal{K})$. A* **packet puzzle** *is an element of $\mathcal{P}(\mathcal{K})$ that is consistent with the rules of Sudoku. A* **Sudoku board***, can be considered as an element of $\mathcal{P}(\mathcal{K})$ with all but one packet filled in each cell, consistent with the rules of Sudoku.*

Let $\mathcal{B}$ denote the set of all Sudoku boards. If a packet puzzle, P, is a subset of a board, $B \in \mathcal{B}$, then we say that P **completes to** B. Note that a packet puzzle can complete to more than one board. This would not be a fun game for us humans to play, but it is necessary to consider for our mathematical context. If a packet puzzle completes to only one board, as all Sudoku puzzles in the newspaper do, then we call this a proper packet puzzle or a puzzle with a unique completion.

Now we come to an important concept: the variety of a packet puzzle.

Definition 3.2. *The* **variety** *of a packet puzzle, P, denoted* $V(P)$ *,is the set of all boards to which* P *completes.* $V(P) \subseteq \mathcal{B}$ *is the set of solutions of* P*. If* P *is a proper packet puzzle, then* $V(P)$ *contains only one element.*

So why do we call this a variety when there are no ideals in sight? It turns out that we can represent the Sudoku constraints as polynomials. If we use Boolean polynomials in 729 variables, one for each packet, then we can create a packet puzzle ideal by adding the packet variables to the ideal generated by the Sudoku constraint polynomials. If we call the packet puzzle ideal P, then in fact, $V(P)$ *is* the variety of the ideal P which represents all possible Sudoku boards to which the puzzle P completes. This is all we will mention in this article about polynomials and ideals. See [1, 3] for more information about polynomial representation of Sudoku.

Note that if $B \in \mathcal{B}$ is a Sudoku board, and $P \in \mathcal{P}(\mathcal{K})$ is a packet puzzle such that $P \subseteq B$. Then $B \in V(P)$. We also have the following statement, based on the fact that $V(P)$ is an actual variety: Let P and Q be packet puzzles. Then $V(P \cup Q) = V(P) \cap V(Q)$.

This small, simple statement helps to show how a sudoku solver can take a given set of packets and add to it new packets. If the solver is adding a set of packets S to a starting set P, then we see that as long as $V(S) \subseteq V(P)$, then $V(P \cup S) = V(P)$; rather, the solutions to the puzzle are preserved after the adding of new packets. This brings us to the concept of an algebraic group which acts on packet puzzles while preserving their solutions. We call these actions **solving symmetries**.

In order to give context to both this group of solving symmetries and the group of Sudoku symmetries defined in Section 1, we define a universal set that encompasses all invertible functions on packet sets.

$$\mathcal{F} = \{f | f : \mathcal{P}(\mathcal{K}) \to \mathcal{P}(\mathcal{K}), f \text{ invertible}\}$$

It is easy to see that $\mathcal{F}$ is a group under composition. Since $\mathcal{P}(\mathcal{K})$ is finite with order 2^{729} The set of invertible functions on $\mathcal{P}(\mathcal{K})$ is isomorphic to the symmetric group on 2^{279} elements, $S_{2^{279}}$. The group of Sudoku symmetries, $\mathcal{G}$, described in Section 1 consists of invertible functions acting on packet sets. Therefore $\mathcal{G}$ is a subgroup of $\mathcal{F}$. Now we define another subgroup of $\mathcal{F}$ which is the topic of this paper.

Definition 3.3. *A* **solving symmetry** *is a invertible function $\sigma \in \mathcal{F}$ such that for all packet puzzles $P \in \mathcal{P}(\mathcal{K})$, $V(\sigma P) = V(P)$. In other words σ maps a packet puzzle to another packet puzzle with the same set of solutions. Let $\mathcal{S}$ denote the set of all solving symmetries.*

It is not hard to see that $\mathcal{S}$ is a group under composition.

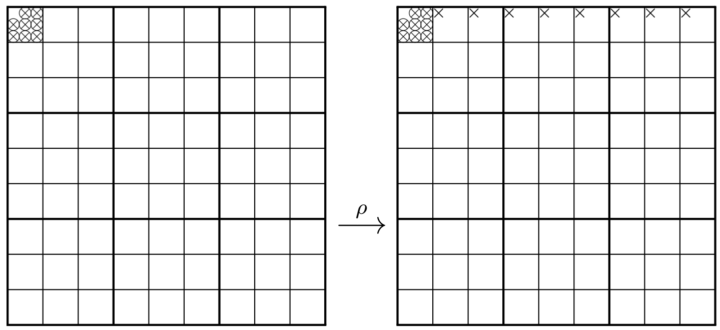
Theorem 3.1. *$\mathcal{S}$ is a subgroup of $\mathcal{F}$.*

Proof. Clearly the identity map, which maps P to P for any packet puzzle P, is a solving symmetry. Every solving symmetry is invertible by definition. Let σ and τ be solving symmetries. So $\sigma(P) = P$ and $\tau(P) = P$ for every packet puzzle, P. Therefore the composition of two solving symmetries is also a solving symmetry since $V(\sigma(\tau P)) = V(\sigma P) = V(P)$. $\square$

Here is an example of a human solving strategy that we would like to represent as a group element of $\mathcal{S}$. We can call the first cell in the Sudoku grid a and the remaining cells in the row b, c, d, e, f, g, h and i. In this example, we have a set of 8 packets in the first cell in all positions except the 1-position. Call this set $T = \{a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9\}$. From this information, the naked single strategy implies that the first cell must contain a 1. The human solver then concludes that a 1 cannot be present in any other cell in the first row. So we know that the packet set $S = \{b_1, c_1, d_1, e_1, f_1, g_1, h_1, i_1\}$ is implied by the packet set T. Now we can define

$$\rho : T \rightsquigarrow S$$

shown below.



ρ acting on its defining set

The solving symmetry ρ will act on a packet puzzle P if and only if P contains the set T. The problem is that ρ is not an invertible function on the set of all packet sets, $\mathcal{P}(\mathcal{K})$. Say, for example, a packet puzzle, P, already has a packet in the b_1 position. If $T \subseteq P$ and we apply ρ to P, then we get a new packet puzzle Q such that $Q = P \cup S$. If we define $\rho^{-1}(Q)$ to be $Q \setminus S$, then we see that $\rho^{-1}(Q) \neq P$, since it is missing the packet in the b_1 position. This issue, however, can be corrected by considering one packet at a time and taking the symmetric difference of P and S rather than the union.

Definition 3.4. *Suppose a packet set, T, implies a packet set S. By this we mean that $V(T) = V(T \cup S)$. Then we call T a* **triggering set** *for S and S the* **implied packet set** *for T.*

Next, we define the building block for Solving Symmetries.

Definition 3.5. *Given a triggering set, T, and a packet $k \notin T$, we define a* **primitive solving symmetry** *to be a function*
$F_{T,k} : \mathcal{P}(\mathcal{K}) \to \mathcal{P}(\mathcal{K}) \in \mathcal{F}$ as follows:

$$F_{T,k}(P) = \begin{cases} P\Delta\{k\} & \textit{if } T \subseteq P \\ P & \textit{if } T \not\subseteq P \end{cases}$$

where T is the triggering set for the single packet k. If the triggering set T is in the packet set P, then we form the symmetric difference, $P\Delta\{k\}$ and add *it to T. In other words, we add the packet k to P if it is not already there, and take it away if it is there. If the triggering set T is* **not** *a subset of the puzzle P, then we do nothing.*

Next we show that a primitive solving symmetry is indeed a solving symmetry.

Theorem 3.2. *Given a triggering set T and a single packet k implied by T, the primitive solving symmetry $\sigma = F_{T,k}$, is a solving symmetry.*

Proof. Let $\sigma = F_{T,k}$ be a primitive solving symmetry. σ is an invertible function from $\mathcal{P}(\mathcal{K})$ to $\mathcal{P}(\mathcal{K})$. Note that $\sigma^{-1} = \sigma$. Now consider an arbitrary packet set $P \in \mathcal{P}(\mathcal{K})$. To show that σ is a solving symmetry, we need to show that $V(\sigma P) = V(P)$. Suppose that $T \not\subseteq P$. Then $\sigma P = P$, so $V(\sigma P) = V(P)$.

Suppose now that $T \subseteq P$. We have two cases. Either the packet k is in P or it is not in P. Suppose $k \in P$. Then $\sigma(P) = P\setminus\{k\}$. We want to show that $V(P \setminus \{k\}) = V(P)$. Since T is a triggering set for $\{k\}$, $V(T \cup \{k\}) = V(T)$. Since $T \subseteq P$, we have that
$V(P) = V(P \cup T) = V(P \setminus \{k\} \cup \{k\} \cup T) = V(P \setminus \{k\}) \cap V(\{k\} \cup T) = V(P \setminus \{k\}) \cup V(T) = V(T \cup P \setminus \{k\}) = V(P \setminus \{k\})$ which is what we wanted to show.

Suppose now that $k \notin P$. Then $\sigma(P) = P \cup \{k\}$. We want to show that $V(P \cup \{k\}) = V(P)$. Now we have that $V(P \cup \{k\}) = V(P \cup T \cup \{k\}) = V(P) \cap V(T \cup \{k\}) = V(P) \cap V(T) = V(P \cup T) = V(P)$.

So in both cases, $V(\sigma P) = V(P)$ □

Does the set of primitive solving symmetries form a group? The answer is no. The identity is, in fact, not a primitive solving symmetry. And the composition of two primitive solving symmetries is not a primitive solving symmetry. The composition of two primitive solving symmetries is, however, a solving symmetry, which prompts our next definition.

Definition 3.6. *A* **simple solving symmetry** *is a primitive solving symmetry or the composition of two or more primitive solving symmetries with* *the same triggering set. If τ is a simple solving symmetry with triggering set T which implies a packet set, S, then*

$$\tau(P) = \begin{cases} P \Delta S & \textit{if } T \subseteq P \\ P & \textit{if } T \not\subseteq P \end{cases}$$

Note that two simple solving symmetries with the same triggering set commute. Simple solving symmetries with different triggering sets may not commute but their composition is still a solving symmetry. This gives us the following theorem.

Theorem 3.3. *Denote by $\mathcal{S}^3$ the set of compositions of simple solving symmetries together with the identity solving symmetry. Then $\mathcal{S}^3$ is a group under composition.*

Now any human solving symmetry can be represented by a simple solving symmetry in $\mathcal{S}^3$. In fact, given a puzzle that completes to a unique board, the set of all packets in the puzzle can be considered a triggering set, T, and the remaining packets needed to complete the board can be considered the set of implied packets, S. So any Sudoku puzzle can be solved in one fell swoop with a single solving symmetry! However, most of us humans are not so adept at recognizing such a large triggering set and implied packet set, so we solve the puzzle one small step at a time and use a composition of solving symmetries.

Now we have the group of simple solving symmetries, $\mathcal{S}^3$, sitting inside the group of all solving symmetries, $\mathcal{S}$, sitting inside the group of all invertible functions on packet sets, $\mathcal{F}$. An obvious question to ask is, "Are there any solving symmetries that are not compositions of simple solving symmetries?" In other words, are there other invertible functions on packet sets that are not compositions of these symmetric differences of packet sets? We do not know the answer, but we have a conjecture.

Conjecture 3.1 *There are no solving symmetries other than the composition of simple solving symmetries and the identity. In other words, $\mathcal{S} = \mathcal{S}^3$.*

4 Déjà Vu

Now we return to the question that we asked in the introduction. Why do we get a feeling of déjà vu when we solve two puzzles that are essentially the same? The answer is probably clear by now: Because we use the same solving symmetry to solve both!

P

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

$\overset{\pi}{\longrightarrow}$ $\underset{\pi^{-1}}{\longleftarrow}$

πP

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

$\downarrow \sigma$ $\qquad$ $\downarrow \pi\sigma\pi^{-1}$

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

B

$\overset{\pi}{\longrightarrow}$ $\underset{\pi^{-1}}{\longleftarrow}$

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

πB

Here is the example from the introduction. Suppose we have a Sudoku puzzle, P (here represented with clues rather than packets for readability) that completes to a Sudoku board, B. Now suppose there is a Sudoku symmetry, $\pi \in \mathcal{G}$, the Sudoku Symmetry Group, that maps P to another Sudoku puzzle Q. In our example, $\pi = r \circ (159483726)$, the permutation of the entries followed by a rotation of 90 degrees. So $Q = \pi(P)$. Q completes to the board $\pi(B) = C$. Now, suppose the solving symmetry needed to take P to B is

σ. Here is the explanation for the déjà vu: The solving symmetry needed to take Q to C is $\pi^{-1}\sigma\pi$. It is exactly the same solving symmetry, σ!

5 Conclusion

There is a wealth of mathematics to be discovered and explored concerning Sudoku puzzles. The idea of a packet enables us to define several algebraic groups that act on Sudoku boards and puzzles. This concept also allows us to mathematically codify how humans solve Sudoku puzzles. We hope to understand better this very large, but finite group of Solving Symmetries.

Acknowledgment This research was performed at a summer REU at James Madison University and supported by NSF DMS-1004516. We would like to thank our REU colleagues Bjorn Wastvedt, Eddie Tu, and Dr. Stephen Lucas for their help and support.

References

[1] E. Arnold, S. Lucas, & L. Taalman, Gröbner Basis Representation of Sudoku, *College Mathematics Journal*, **41**, 2010, 101-111.

[2] B. Felgenhauer & F. Jarvis, Mathematics of Sudoku I, *Mathematical Spectrum*, **39**, 2006, 15–22.

[3] J. Gago-Vargas, I. Hartillo-Hermoso, J. Martín-Morales, & J.M. Ucha-Enrq́uez, Sudokus and Gröbner bases: not only a divertimento, *Computer algebra in scientific computing*, 155–165, *Lecture Notes in Comput. Sci.*, **4194**, Springer, Berlin, 2006.

[4] J. Gallian, *Contemporary Abstract Algebra, 7th Ed.*, Brooks/Cole, California, 2010.

[5] The GAP Group, GAP – Groups, Algorithms, and Programming, Version 4.4.12; 2008. (`http://www.gap-system.org`)

[6] D.E. Knuth, *Dancing Links*, in: J. Davies, B. Roscoe & J. Woodcock, *Millennial Perspectives in Computer Science: Processings of the 1999 Oxford-Microsoft Symposium in Honour of Sir Tony Hoare*, Palgrave, 2000, 187–214.

[7] E. Russell & F. Jarvis, Mathematics of Sudoku II, *Mathematical Spectrum*, **39**, 2006, 54–58.

[8] Sudopedia, the free Sudoku reference guide, `www.sudopedia.org/wiki`, accessed September 21, 2010.

[9] L. Taalman, Taking Sudoku seriously, *Math Horizons*, September 2007, 5–9.