Sudoku as Sets and Predicate Logic

Ren Lin

Abstract

This paper explores the combinatorial puzzle Sudoku through ideas from set theory and predicate logic. The rules of Sudoku, which require distinct values in each row, column, and subgrid, can be formalized using definitions in set theory and written as logical statements. Thinking of Sudoku as a grid related to sets and logic, and solving the puzzle by finding values that meet all logical rules, presents useful applications of mathematical ideas.

Contents

1 Introduction

Sudoku is a combinatorial puzzle that is solved by completing a partially filled 9×9 so that each row, column, and 3×3 grid contains each of the digits from 1 to 9 exactly once. Some see Sudoku as a simple game, but it has a deep mathematical structure that we can study using set theory and predicate logic. Solving a Sudoku puzzle requires satisfying certain conditions across rows, columns, and subgrids. These conditions required to solve a Sudoku puzzle can be described using sets and logical statements.

The discussion begins in Section 2 with an overview of set theory and predicate logic, focusing on ideas that will be essential for modeling Sudoku. In

Section 3, we will define the Sudoku grid and its rules using ideas from Section 2. Section 4 uses all past sections to solve a puzzle in Sudoku by checking conditions for solution uniqueness. Sections 5 and 6 include final reflections and references.

2 Preliminaries

Definition 2.1. (Set). A collection of distinct objects, called elements. Examples of sets include:

(1) **Empty:** The empty set, denoted $\emptyset$, contains no elements.

(2) **Subset:** A set A is a subset of B, written $A \subseteq B$, if every element of A is also an element of B.

(3) **Power Set:** The power set $\mathcal{P}(A)$ is the set of all subsets of A.

Definition 2.2. (Element). If a is an object and A is a set, we write $a \in A$ to indicate that a is an element of A.

Definition 2.3. (Ordered Pair). An ordered pair (a, b) is a pairing of two elements where the order matters. That is, $(a, b) \neq (b, a)$ in general.

Definition 2.4. (Relation). A relation is a set of ordered pairs. For example, a relation R between sets A and B is written as:

$$R = \{(a, b) \mid a \in A,\ b \in B\}.$$

Definition 2.5. (Predicate). A predicate is a logical statement involving one or more variables. The truth value of the statement depends on the values of the variables. For example, $P(x)$ might be true for some values of x and false for others.

Definition 2.6. (Quantifiers).

(1) The universal quantifier $\forall$ means "for all". For example, $\forall x \in A$, $P(x)$ means that $P(x)$ is true for every x in A.

(2) The existential quantifier $\exists$ means "there exists" or "there is at least one". For example, $\exists x \in A$, $P(x)$ means there is at least one x in A for which $P(x)$ is true.

Definition 2.7. (Satisfiability). A set of logical statements is satisfiable if there exists an assignment of values to its variables that makes all statements true.

3 Formalizing Sudoku

A Sudoku grid is a 9x9 grid composed of 81 cells. Each cell is located using a pair of coordinates (i, j) where i represents the row index and j represents the column index. The set of all cells in the grid is given by:

$$G = \{(i, j) \mid 1 \leq i \leq 9, 1 \leq j \leq 9\}$$

The set of possible digits in a Sudoku puzzle is:

$$D = \{1, 2, \ldots, 9\}$$

A Sudoku solution is a relation $R \subseteq G \times D$, where $G \times D$ is the Cartesian product of the set of cells G and the set of digits D. R is a set of ordered pairs $((i, j), d)$, where $(i, j) \in G$ and $d \in D$, such that $((i, j), d) \in R$ indicates that the cell (i, j) contains the digit d. The predicate $P(i, j, d)$ is defined as:

$$P(i, j, d) \text{ is true if and only if } ((i, j), d) \in R$$

The 3x3 subgrids are defined as sets of cells. For each $k, l \in \{1, 2, 3\}$, the subgrid $S_{k,l}$ contains cells (i, j) where:

$$S_{k,l} = \{(i, j) \mid 3k - 2 \leq i \leq 3k, 3l - 2 \leq j \leq 3l\}.$$

A valid Sudoku solution must satisfy the following constraints, expressed in terms of the predicate $P(i, j, d)$, which is true if cell (i, j) contains digit $d \in D = \{1, 2, \ldots, 9\}$:

1. Each row contains at most one instance of each digit.

   $$\forall i \in \{1, \ldots, 9\}, \forall d \in D, \forall j_1, j_2 \in \{1, \ldots, 9\}, (j_1 \neq j_2) \implies \neg(P(i, j_1, d) \wedge P(i, j_2, d))$$

   This ensures that if $P(i, j_1, d)$ and $P(i, j_2, d)$ both hold for the same digit d, then j_1 must equal j_2, preventing the same digit from appearing twice in a row.

2. Each column contains at most one instance of each digit.

   $$\forall j \in \{1, \ldots, 9\}, \forall d \in D, \forall i_1, i_2 \in \{1, \ldots, 9\}, (i_1 \neq i_2) \implies \neg(P(i_1, j, d) \wedge P(i_2, j, d))$$

   This ensures that for any two different rows $i_1 \neq i_2$, the same digit d cannot appear twice in the same column.

3. Each 3x3 subgrid contains at most one instance of each digit.

   $$\forall k, l \in \{1, 2, 3\}, \forall d \in D, \forall (i_1, j_1), (i_2, j_2) \in S_{k,l},$$
   $$(i_1, j_1) \neq (i_2, j_2) \implies \neg(P(i_1, j_1, d) \wedge P(i_2, j_2, d))$$

   This ensures that digit d appears at most once per 3x3 subgrid, where $S_{k,l} = \{(i, j) \mid 3k - 2 \leq i \leq 3k, 3l - 2 \leq j \leq 3l\}$.

4. Every cell contains exactly one digit.

$$\forall i \in \{1,\ldots,9\}, \forall j \in \{1,\ldots,9\}, \exists d \in D, P(i,j,d) \wedge \forall d' \in D, (P(i,j,d') \to d' = d)$$

This ensures that for each cell (i,j), there exists a digit $d \in D$ such that $P(i,j,d)$ holds, and any other digit $d' \in D$ that also satisfies $P(i,j,d')$ must be equal to d, guaranteeing each cell contains exactly one digit.

4 Applications and Analysis

In this section, we explore how the set-theoretic and predicate logic framework developed in Sections 2 and 3 can be applied to solve a Sudoku puzzle and analyze the uniqueness of its solution. By modeling Sudoku as a satisfiability problem, we can leverage logical constraints to systematically assign digits to cells and verify whether a given puzzle has a unique solution. This approach not only provides a mathematical foundation for solving Sudoku but also connects to broader applications in computer science and constraint satisfaction problems.

The formalization of Sudoku in Section 3 defines a valid Sudoku solution as a relation $R \subseteq G \times D$ that satisfies the constraints expressed by the predicate $P(i,j,d)$. These constraints (rows, columns, subgrids, and single-digit per cell) form a set of logical statements that must be simultaneously satisfied. This setup naturally lends itself to a satisfiability problem, where the goal is to find an assignment of truth values to $P(i,j,d)$ for all $(i,j) \in G$ and $d \in D$ such that all constraints hold.

Consider a partially filled Sudoku grid, where some cells $(i,j) \in G$ are assigned specific digits $d \in D$. These assignments correspond to a subset of the relation R, denoted as the set of "given" entries:

$$R_0 = \{((i,j),d) \mid \text{cell } (i,j) \text{ is pre-filled with digit } d\}.$$

The task is to extend R_0 to a complete relation R that satisfies all constraints in Section 3. In logical terms, we seek a model for the conjunction of the following:

1. The constraints from Section 3 (row, column, subgrid, and single-digit conditions).
2. The given entries: $\forall((i,j),d) \in R_0, P(i,j,d)$.

To illustrate, consider a simple 4x4 Sudoku puzzle (a smaller variant for brevity) with digits $D = \{1,2,3,4\}$ and 2x2 subgrids. Suppose the given entries are:

$$R_0 = \{((1,1),1),((2,2),2),((3,3),3),((4,4),4)\}.$$

This corresponds to the following grid:

1	.	.	.
.	2	.	.
.	.	3	.
.	.	.	4

The solver must assign digits to the remaining cells such that:

1. Each row $i \in \{1, 2, 3, 4\}$ contains each digit in D exactly once.
2. Each column $j \in \{1, 2, 3, 4\}$ contains each digit in D exactly once.
3. Each 2x2 subgrid $S_{k,l}$ (for $k, l \in \{1, 2\}$) contains each digit in D exactly once.
4. Each cell contains exactly one digit.

Using the predicate $P(i, j, d)$, we can encode these constraints and use a logical solver (or manual deduction) to find a satisfying assignment. For example, the constraint that row 1 contains digit 1 exactly once includes the fact that $P(1, 1, 1)$ is true (since $((1, 1), 1) \in R_0$), and thus $P(1, j, 1)$ must be false for $j = 2, 3, 4$. By iteratively applying the constraints, we can deduce values for each cell. For example, if we find that $P(1, 2, 2)$ is true, we update R to include $((1, 2), 2)$ and ensure that no other cell in row 1 or column 2 contains digit 2.

A valid solution to this 4x4 Sudoku puzzle, obtained by satisfying all constraints, is:

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

This solution can be verified by checking that each row, column, and 2x2 subgrid contains the digits $\{1, 2, 3, 4\}$ exactly once. For example, row 1 $(1, 3, 4, 2)$ and the top-left subgrid $((1, 1) = 1, (1, 2) = 3, (2, 1) = 4, (2, 2) = 2)$ satisfy the constraints.

This process is similar to algorithmic approaches like backtracking or constraint propagation used in computer science to solve Sudoku. The logical framework allows us to formalize these algorithms as searches for a satisfying assignment to the predicate P.

A well designed Sudoku puzzle typically has exactly one solution. Using the formalization from Section 3, we can analyze whether a given puzzle has a unique solution by checking whether there exists exactly one relation $R \subseteq G \times D$ that satisfies all constraints and includes the given entries R_0.

Formally, a puzzle has a unique solution if the set of logical statements (constraints plus given entries) is satisfiable, and any two satisfying assignments $R_1, R_2 \subseteq G \times D$ are identical. To verify uniqueness, we can attempt to find two distinct solutions. Suppose R_1 is a valid solution, such as the one provided above for the 4x4 puzzle. We can add a constraint to force a different solution:

$$\exists (i, j, d) \in G \times D \text{ such that } P_1(i, j, d) \neq P_2(i, j, d),$$

where P_1 and P_2 represent the predicates corresponding to R_1 and R_2. If the resulting set of constraints is unsatisfiable, then no second solution exists, and the puzzle has a unique solution.

5 Conclusion

This paper demonstrates how Sudoku, a popular combinatorial puzzle, can be rigorously modeled using set theory and predicate logic. By representing the puzzle's grid, digits, and rules as sets and logical constraints, we developed a framework to solve Sudoku and verify the uniqueness of its solutions. This approach transforms Sudoku into a constraint satisfaction problem (CSP), an important concept in computer science and artificial intelligence. The formalization enables the use of algorithms like backtracking and constraint propagation, which are applicable to real-world problems such as scheduling, graph coloring, and circuit design. This logical perspective connects Sudoku to automated reasoning and model checking, which is a practical way to explore set theory and predicate logic. Future research could extend this framework to larger puzzles, such as 16x16 Sudoku, or investigate variants like diagonal Sudoku.

6 References

[1] Robert R. Stoll, *Set Theory and Logic*, Dover Publications, 1979.