PATTERN POWER

The Secret Power of Subpatterns

By G. A. Aldrich

Version 3.1 June 30, 2015

Foreword

When I developed the idea of solution by subpattern, it was out of pure intellectual curiosity. It did not arrive at once; like a bolt out of the blue, but something that came together over a period of time. The very difficulty of not quite knowing how to go forward when I was presented with more than one choice was, in the end, the breakthrough. Choosing the way to proceed was the answer. Dealing with the choice of how to proceed led to the idea of pending boxes, which are all about choice.

Once I saw my way through the difficulties of carrying this method to the point of actually solving sudokus all by itself, I found myself using it whenever I could not succeed with standard methods. My dependence on it increased my understanding to the point that I began using it to solve sudokus which had previously taken me as much as an hour, often in as little time as fifteen minutes, and frequently with the first subpattern I tried.

The most important factor was the use of a separate sudoku form just for the pattern itself, and the integration of subpattern solution with the other single-candidate solving tools, especially n-wings and polarity chains. Once I had gone to the trouble of assembling the pattern in front of me, I wasn't eager, in those many instances when initial atempts at subpattern analysis failed, to just erase the pattern and go on to the next one. I saw the necessity of coordinating it with those other methods, in order to reduce the problems posed by complicated patterns.

At first it seemed like cheating to use an extra sudoku form, but it was so much easier to see the n-wings, especially the higher-level ones, compared to the effort of straining one's eyes and brain to distinguish them through the maze of other candidates, The experts constantly harped on the esthetics of training one's eye to tease out subtle relationships, and the virtues of keeping erasures to a minimum. Using an extra form allowed me to train myself more directly in mastering those subtleties. Most importantly, subpatterns were impossible to deal with otherwise, and subpatterns proved to be my road to success.

I should like to express my thanks to Nick Fittinghoff for editing the first draft of this manuscript, as well as offering some useful suggestions on the wording of certain descriptions, but most importantly for shaming me into somehow publishing this method, or, at the very least, getting it out there for the sudoku community to see.

Table of Contents

Table of Contents, continued

Chapter 1. Initial Information.

To present a sudoku, a certain set of “givens” is specified. These are the clues to the puzzle, the squares whose values are known.

This is the ***presentation*** of a sudoku.
The non-blank squares with digits
are the ***givens***: and this is the ***solution***:

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

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

Together, they create the puzzle which is called a ***sudoku***.

Look at the solution on the right. Pick out a row at random. Look at the digits in it. All nine digits, 1-9. Check out any column. Same answer. Look at one of the boxes. Again the same. All nine digits every time. This is the secret to solving sudokus. A systematic approach to solving is built upon it. You can always count on it

This simple quality of possessing all nine digits is so important that it deserves some special attention, as well as a special word. The term "**9-complete**" is used to describe it.

The rule of nine: Any collection of nine squares is called **9-complete** or **9-perfect** if every square in the collection contains one and only one digit between 1 and 9, and every digit from 1 to 9 is represented.

Nine different squares, nine different digits, one in each square
This is *9-completeness*

Notation:

The squares of a sudoku are labeled as follows:

(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)
(21)	(22)	(23)	(24)	(25)	(26)	(27)	(28)	(29)
(31)	(32)	(33)	(34)	(35)	(36)	(37)	(38)	(39)
(41)	(42)	(43)	(44)	(45)	(46)	(47)	(48)	(49)
(51)	(52)	(53)	(54)	(55)	(56)	(57)	(58)	(59)
(61)	(62)	(63)	(64)	(65)	(66)	(67)	(68)	(69)
(71)	(72)	(73)	(74)	(75)	(76)	(77)	(78)	(79)
(81)	(82)	(83)	(84)	(85)	(86)	(87)	(88)	(89)
(91)	(92)	(93)	(94)	(95)	(96)	(97)	(98)	(99)

There are only 81 squares, though their numbering goes to 99. Each square's number identifies its row and column, with the row number first.

The boxes are labeled as follows:

A	B	C
D	E	F
G	H	I

Each box is a 3 x 3 array of 9 squares:

These are the contents of the nine boxes:

Box A:

(11)	(12)	(13)
(21)	(22)	(23)
(31)	(32)	(33)

Box B

(14)	(15)	(16)
(24)	(25)	(26)
(34)	(35)	(36)

Box C

(17)	(18)	(19)
(27)	(28)	(29)
(37)	(38)	(39)

Box D:

(41)	(42)	(43)
(51)	(52)	(53)
(61)	(62)	(63)

Box E

(44)	(45)	(46)
(54)	(55)	(56)
(64)	(65)	(66)

Box F

(47)	(48)	(49)
(57)	(58)	(59)
(67)	(68)	(69)

Box G:

(71)	(72)	(73)
(81)	(82)	(83)
(91)	(92)	(93)

Box H

(74)	(75)	(76)
(84)	(85)	(86)
(94)	(95)	(96)

Box I

(77)	(78)	(79)
(87)	(88)	(89)
(97)	(98)	(99)

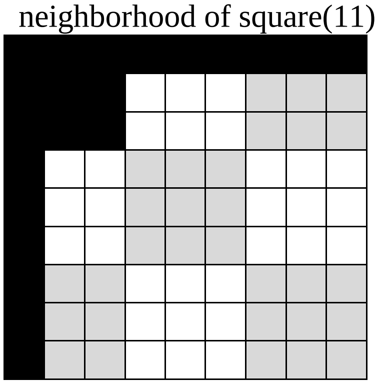
(27)	*means*	square (27)
square (45) = 8	*means*	the value of square (45) is 8
a 2-square	*means*	a square with the single digit 2
A	*means*	Box A
A→B	*means*	Box A precedes Box B in the order of boxes.
nine-perfect	*describes*	a row, box, or column that has all nine digits as established values.
n-wing	*means*	x-wing, swordfish, jellyfish, or squirmbag, [2-wing = x-wing, 3-wing = swordfish, 4-wing = jellyfish & 5-wing = squirmbag]

The **givens** are the original values in the sudoku, the clues to discovering the rest of the values. To the givens are joined the values of **solved squares,** solved by you, all together making up the **established values**, the single digits in the centers of squares. The **candidates** are the tiny digits at the tops of the **unsolved squares** that you penciled in during **annotation**. Squares with candidates are also called **annotated squares**.

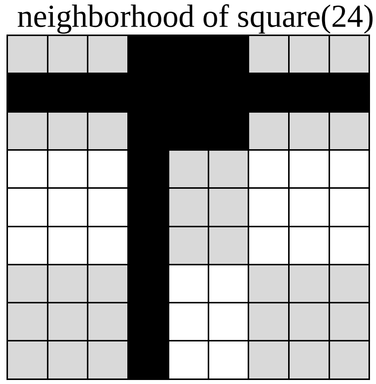
For any square, its **neighborhood** is the set of all squares in the same row, plus all those in the same column, plus all those in the same box. There are 21 squares in each neighborhood. Every neighborhood is unique. Since there are 81 squares in a sudoku, there are also 81 neighborhoods.

Fifteen Random Neighborhoods:

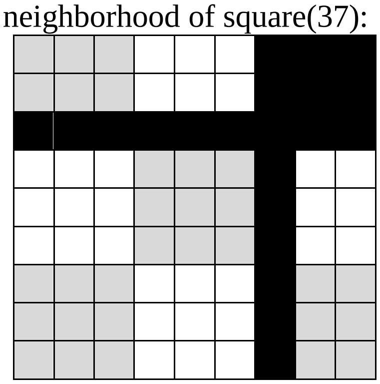
neighborhood of square(11):



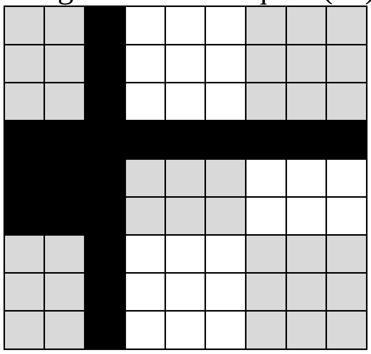
neighborhood of square(24):



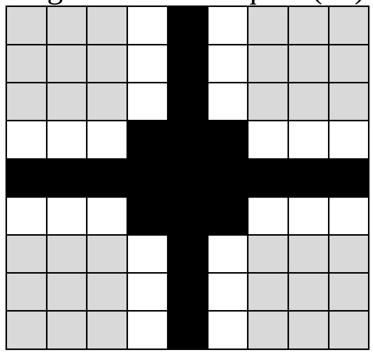
neighborhood of square(37):



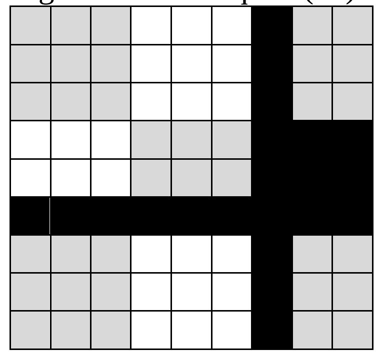
neighborhood of square(43):



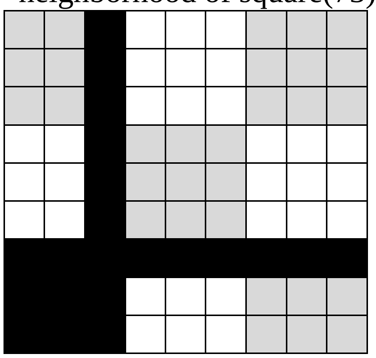
neighborhood of square(55):



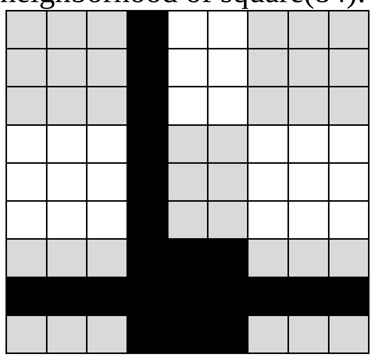
neighborhood of square(67):



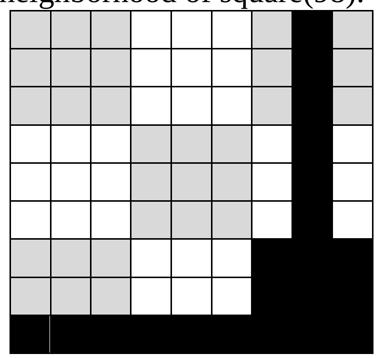
neighborhood of square(73):



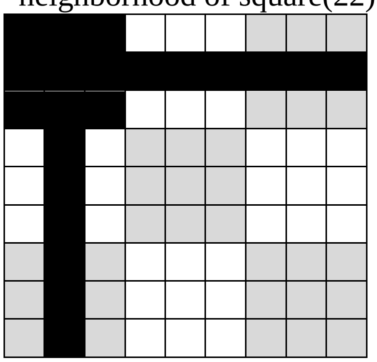
neighborhood of square(84):



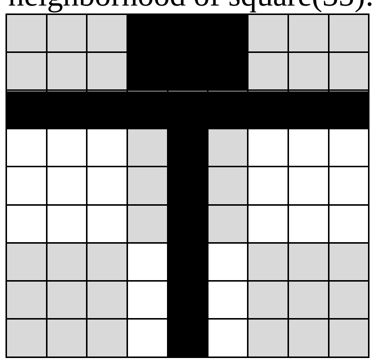
neighborhood of square(98):



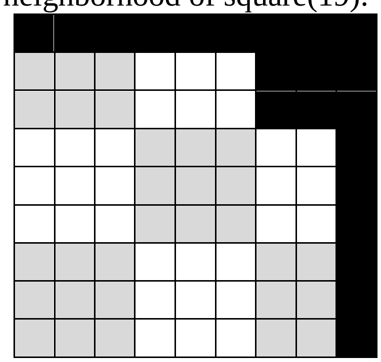
neighborhood of square(22):



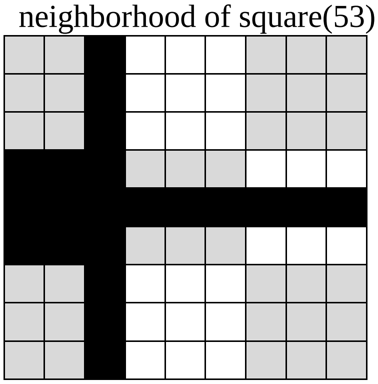
neighborhood of square(35):



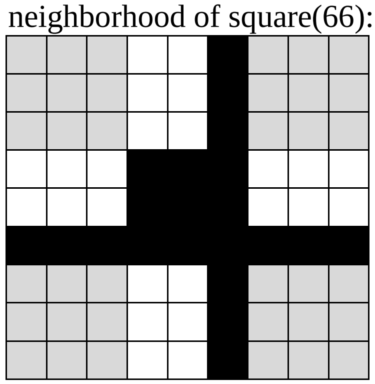
neighborhood of square(19):



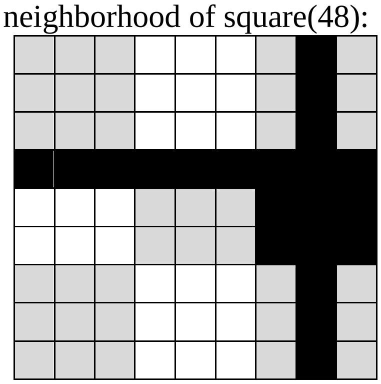
neighborhood of square(53):



neighborhood of square(66):



neighborhood of square(48):



Comments on the proofs in this book:

The attempt has been made in this book to use proofs which give insight into why something is true. They are proofs which rely on the understanding of the relationships between boxes, squares, and subpatterns. This understanding is not generally analytic. It is related to the method outlined in this book, and is readily understood after the relationships between subpatterns and their sharing or non-sharing of squares has been explained. The reader would do well to try to understand these "proofs" because in the process a number of related matters are laid out in what I hope is an understandable way, giving the reader a relaxed approach to the use of the method. If you understand the rules of the method and why they work, then you can all the more readily use the method successfully. There is nothing esoteric to them, as they depend on common sense.

It is for reasons like the above that is it difficult to determine which group of sudoku solvers are most likely to find this method a worthy addition to their arsenal of sudoku weapons. It is not really a matter of beginner versus advanced. It is a matter of confortableness with concepts and ideas. Instead of formulas to calculate, there are relationships to be perceived. What is required is a meditative frame of mind rather than a contest-winning one. The reader should find the ideas challenging and interesting. They are the product of imagination and creativity rather than of perseverance and determination, although these last characteristics must also be qualities of the solver seeking a greater mastery of this method.

The logic of solving sudokus:

1. *The two basic assumptions about sudokus are:*

a. All the rows, columns, and boxes of a sudoku obey the Rule of Nine.
b. Every sudoku has a unique solution, which agrees with the givens. There is only one set of possible values for the blank squares, and they must be logical consequences of the givens.

2. *We add to these the rule of logic:*

Any assumption which leads to a contradiction is false, and its logical opposite is true.

3. *And, combining these, we get the basic principle:*

If the assumption that a square has a particular value leads to a logical contradiction, then that square does not have that value.

Contradiction: An illogical situation, one which is not allowed by the rules of nine perfection, or one which contradicts the assumption that a subpattern being tested is the correct one, or one which contradicts the assumption that the sudoku has but one solution. This last situation occurs when a fatal four is encountered. It is universally agreed upon today by all sudoku creators that a sudoku must have one and only one solution. You may always make this assumption, that there is exactly one solution. Therefore the occurrence of a fatal four is a contradiction, and the assumption which led to this situation must be false.

Chapter 2. Patterns

Until I became acquainted with x-wings, I had seen no discussions of the patterns of candidates in an annotated sudoku. From the outset of my solving, I was bemused by them, and wondered why so little was said about them. I assumed they were too complex for anyone to consider analyzing them. Even x-wings and their higher siblings didn't address digit patterns, except to exclude candidates in certain rows or columns of the sudoku.

I first encountered the word "unthreading" in Michael Mepham's "The Book of Sudoku," published in 2005. In a section titled "Ariadne's thread," followed by another called "Tough Sudokus," the idea of branching, akin to what I call "pending," was introduced to me, the situation of having more than one choice presented for a subpattern, although neither the term nor the concept of subpattern was used..

If you look at the pattern of a particular digit in a fully solved sudoku, you will see exactly one instance of that digit in every row, column, and box. This property connects all the members of a digit pattern. If we define two squares as being **related** when they are in the same row, column, or box, then we can say that all the squares in a digit pattern must also be **unrelated**, else they could not complete the digit pattern for that digit. If we define two squares as being **connected** when they are unrelated, then we can say that the squares of a digit pattern are all **connected** to one another. An alliance of anarchists.

When we study a candidate pattern, we want to identify the candidates which will become established (solved) values. They will form a single subpattern, which must have the same characteristic as the partial pattern of established digits, and this single subpattern will then complete the existing partial pattern.

The study of a candidate pattern in a sudoku would be useless if there were no solved squares, either initially as givens, or with the candidate digit promoted to the status of a solved digit, because all of the remaining squares would have that digit as a candidate, turning our search for the correct candidates into a nightmare.

To find this successful subpattern, we identify all possible subpatterns, so that we may then test them, seeking the single one which does not fail. All these subpatterns must have the same characteristic, that their member squares be **connected** with, and therefore **unrelated** to one another.

Below is a partly solved sudoku, which has been through the basic solving techniques. We'll arbitrarily look at the 5-candidates. In a later chapter, we'll discuss the reasons for choosing a particular candidate for the purpose of studying its pattern.

To see the 5-candidates by themselves, let's copy the sudoku, omitting all digits except for 5. We're interested only in the candidate 5's, not the established 5's, but we need to know where the established 5's are, so we'll **replace all the established 5's by X's**. We do this in order not to get confused between solved values and candidate values.

Then we'll move the 5-candidates to the centers of their squares, and omit all the other candidates, so we can still more easily see the 5-candidates by themselves.

This is the sudoku we want to complete. On the next page is the 5-pattern.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	3	7	258	14	28	6	9	25	14
row2	468	9	28	5	1248	3	14	26	7
row3	46	45	1	49	249	7	3	256	8
row4	1	2	6	7	3	5	8	4	9
row5	48	3	58	149	149	2	15	7	6
row6	7	45	9	6	14	8	125	3	12
row7	9	1	7	2	5	4	6	8	3
row8	5	6	3	8	7	9	24	1	24
row9	2	8	4	3	6	1	7	9	5

This is the pattern of the 5-candidates. The X's show the squares where the already determined established 5's exist. They are represented as X's to distinguish them from the 5-candidates. The correct subpattern will be a collection of 5-candidates which will complete the existing pattern of X's.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1			5					5	
row2				X					
row3		5						5	
row4						X			
row5			5				5		
row6		5					5		
row7					X				
row8	X								
row9									X

Note that there are two 5’s in each unsolved box. Accordingly, there must be more than one thread in this candidate pattern. Each of these threads will form a subpattern, which we need to examine as a possible completion of the already existing established 5's, shown as X's in the diagram above.

When we find the right subpattern, we will promote its 5-candidates into established values, and they must be capable of coexisting with the already established 5's.

The established 5's already adhere to the rule that no two of them may be in the same row, column, or box.

The correct subpattern must therefore also follow this same rule, that no two 5-candidates can be in the same row, column, or box. Therefore we must choose for our subpattern, 5-candidates which are not in the same row, column, or box with each other. The way we express this succinctly is to say that they must not be **related**, that no two of them may be in the same neighborhood (row, column, and box).

In determining a subpatten, we need to start somewhere. We'll go **box by box**, because boxes are easier to examine than rows or columns, and as we go from one box to the next, we must choose as subpattern members those which are not in the same neighborhood as any of the already existing subpattern members. **If we go from box to box, the requirement of not being in the same neighborhood is reduced to simply requiring them to be in a different row or column in the next box, because they are already in a different box.**

As we go through the boxes, we'll add one square from each box to the subpattern, so that **it grows as we go**. In each box, we'll choose among the 5's those we find acceptable. Related squares will be rejected because they share rows or columns with the already chosen subpattern members.

We'll have to visit the boxes in some sort of order. How shall we proceed?

In the diagram below, we notice that if we were to proceed from box F to box C, or from C to F, there would be two choices to make in the second box. This is because none of the 5's are in the same columns in the two boxes, since the 5's in box C are in column 8, while the 5's in box F are in column 7:

A	B	C
D	E	F
G	H	I

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

The same problem does not exist if we go from box C to box A, where rows 1 and 3 contain shared 5-candidates.

Therefore, to avoid multiple choices, we should start in C and finish in F or vice versa. We have only two choices of box order: C→A→D→F or F→D→A→C, one the reverse of the other. We'll choose the first one. This choice will make C the **starting box**, and we'll arbitrarily choose square (18) as the first square of the first subpattern.

Let's look again at the box order we've chosen: C→A→D→F, with box C as the starting box.

A	B	C
D	E	F
G	H	I

Our choice in box C for the first member of subpattern-1 was square (18). Going on to box A, we see that square (13) is related to (18), so the only unrelated square is (32), which becomes the second member of subpattern-1.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1			5					5_1	
row2				X					
row3		5_1						5	
row4						X			
row5			5				5		
row6		5					5		
row7					X				
row8	X								
row9									X

.
The next box is D, where square (62) is in the same column as the second member of the subpattern, so the only unrelated square in D is (53), which becomes the third member.

In the final box, F, square (57) is in the same row as (53); so the only square in F unrelated to subpattern-1 is square (67), the fourth member. We have completed subpattern-1, because it has a single member in every box of the pattern that lacks an established value 5.

A	B	C
D	E	F
G	H	I

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1			5					5_1	
row2				X					
row3		5_1						5	
row4						X			
row5			5_1				5		
row6		5					5_1		
row7					X				
row8	X								
row9									X

We still have a 5-candidate in each of the four pattern boxes, C, A, D, and F. So we begin a new subpattern-2 with the 5-candidate in square (38). In the same manner as before, its extension in A is (13), in D it is (62), and in F it is (57). This completes subpattern-2.

A	B	C
D	E	F
G	H	I

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1			5_2					5_1	
row2				X					
row3		5_1						5_2	
row4						X			
row5			5_1				5_2		
row6		5_2					5_1		
row7					X				
row8	X								
row9									X

Now we have come to the big question: Which subpattern is correct, subpattern-1 or subpattern-2? To settle this, we go back to the original sudoku and test the subpatterns.. We'll first test subpattern-1, which is composed of the squares (18), (32), (53), and (67). We'll accomplish this testing with a **temporary** update.

There is a big difference between updating sudokus in the usual way and performing a **temporary update** to test a subpattern. To test subpattern-1, we insert the 5's of subpattern-1 into the sudoku. They go into the centers of the boxes, just like the established digits.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	3	7	258	14	28	6	9	25 **5**	14
row2	468	9	28	5	1248	3	14	26	7
row3	46	45 **5**	1	49	249	7	3	256	8
row4	1	2	6	7	3	5	8	4	9
row5	48	3	58 **5**	149	149	2	15	7	6
row6	7	45	9	6	14	8	125 **5**	3	12
row7	9	1	7	2	5	4	6	8	3
row8	5	6	3	8	7	9	24	1	24
row9	2	8	4	3	6	1	7	9	5

As a temporary update begins, we have unsolved squares with candidates at their tops and solved squares, each with an established value at its center and no candidates. During a temporary update, only unsolved squares with candidates receive temporary values at their centers. These values will need to be erased if the subpattern proves incorrect.

When an unsolved square is updated, the candidates at its top should be copied to the center before the update is made. If unsolved squares are not updated, their centers remain blank, and their candidates remain their current possible values. The term "solved squares" should be expanded to include the givens.

Never alter the candidates at the tops of squares during a temporary update. At the end of the update all the digits at the centers of the squares will be erased, **except for those squares with no digits at their tops**. These are the original solved squares.

When a square has temporary values at its center, the candidates at its top are ignored.

When a digit is deleted from a square, the deletion should be performed among the digits at the center. Again, never alter the candidate lists at the top during a temporary update. They can be temporarily ignored when there are candidates or temporary solved values at their centers.

When a subpattern proves correct, it is the key that unlocks the puzzle. All the squares of a subpattern are given a solved value, all at the same time. Briefly the solving is slow; then, suddenly, the gridlock breaks, solving gains momentum, and the solving speeds to an end.

We'll take a brief respite from the testing of subpattern-1 to review some important procedures of temporary updating.

The word "**area**" is used in this chapter to mean **9-string**.

a) Promotion: A candidate is promoted through elimination of the other candidates in its square. It becomes the established value and eliminates all like candidates from its neighborhood. This is the main procedure, and sometimes the only one. One such promotion can start a wave of eliminations and promotions, and such a wave is often the main effect of updating with a subpattern.

b) Uniqueness: Through eliminations, a candidate becomes solitary in its row, column, or box. It thereby becomes the established value for its square and again eliminates all like candidates from its neighborhood.

c) Candidate pairs: Two pairs of candidates in the same row, column or box—for example, two squares that each contain only a 2 and a 7-- eliminate all like candidates from the same area (row, column, or box).

d) Group separation. Several squares in an area (row, column or box) prove to be the only squares containing an equal number of individual candidates equal to the number of squares. These candidates claim those squares, eliminating all like candidates in other squares in the area.

Sometimes a square that receives an insertion proves to be one of only two squares in its area that contain a particular other candidate. Thus the insertion causes that candidate to become unique in the other square and to become its established value. Causing such a promotion is the most effective way that a subpattern helps solve a sudoku.

Another, more common way that a subpattern candidate succeeds is by eliminating its own occurrence in another square that holds only one other candidate. The other candidate, now solitary in its square, becomes the square's established value..

Ideally, subpattern insertions **immediately** update the squares surrounding the insertions. A subpattern works well only when at least some of its digits produce quick changes in squares in their areas. As noted above, waves of values that imply other values make a subpattern succeed. As you solve, it is vitally important to keep all the implied changes current, square by square. Otherwise, it's easy to go wrong, even when the subpattern you are testing is correct.

We'll return soon to testing subpattern-1 of the 5-pattern. First, however, some comments on subpattern testing need to be made.

Some advice on temporary updates

When performing a temporary update, it is easy to leap into the task too carelessly, because the updates are so obvious, and there is no end to them. It is a feast of simple food. But it is not wise to progress too quickly. because then it becomes an almost unconscious process, since it is so easy, and it is in allowing oneself to go unconscious that errors occur. It is dull work, but it must be precise. What will you do if every subpattern fails? Start over again:? And, if so, do you trust the first result you get, when before you were in error?

It is best to avoid such situations. Remember all the tedious work you went through, just to get to this endgame. Must it all go to waste because of haste? Take your time. It is not the most exciting work, but it is what you have chosen. So clear your mind of extraneous thoughts, and set to the work with a grim will. No stray thoughts. Just the threat of doing it all for naught. And when the end comes, it will be a successful one. You can have a smoke, or a glass of wine, instead of pinned to the puzzle, still puzzling, making up for lost time.

You must believe in yourself. Then you become more confident in performing your task, so you begin to enjoy the process, so you pay more attention to it and avoid errors, helping you to believe in yourself.

Now we'll return to the testing of subpattern-1 of the 5-pattern, with 5's in (18), (32), (53), and (67). We place 5's in the centers of those squares, and then see whether the insertions solve the sudoku..

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	3	7	258	14	28	6	9	25 **5**	14
row2	468	9	28	5	1248	3	14	26	7
row3	46	45 **5**	1	49	249	7	3	256	8
row4	1	2	6	7	3	5	8	4	9
row5	48	3	58 **5**	149	149	2	15	7	6
row6	7	45	9	6	14	8	125 **5**	3	12
row7	9	1	7	2	5	4	6	8	3
row8	5	6	3	8	7	9	24	1	24
row9	2	8	4	3	6	1	7	9	5

We first update the top three rows by themselves:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	3	7	258 **28**	14	28	6	9	25 **5**	14
row2	468 **46**	9	28	5	1248	3	14	26	7
row3	46	45 **5**	1	49	249	7	3	256 **26**	8

1. The 5 in square (18) reduces square (13) to 28.
2. The pair of 28's in squares (13) and (23) reduces square (21) to 46.
3. The 5 in square (18) reduces square (38) to 26.

Updating the three middle rows by themselves:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row4	1	2	6	7	3	5	8	4	9
row5	48 **8**	3	58 **5**	149 **49**	149 **49**	2	15 **1**	7	6
row6	7	45 **4**	9	6	14 **1**	8	125 **5**	3	12 **2**

4. The 5 in square (67) makes square (62) = 4, which makes square (51) = 8.
5. That same 5 makes square (57) = 1, which makes square (69) = 2.
6. The 4 in square (62) makes square (65) = 1.
7. The 1 in square (65) makes the pair of squares (54) & (55) both = 49.

There are no updates that can be made within the three bottom rows by themselves, so we put the top six rows together:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	3	7	258 **28**	14	28	6	9	25 **5**	14
row2	468 **46**	9	28	5	1248	3	14 **1**	26	7
row3	46	45 **5**	1	49	249	7	3	256 **26**	8
row4	1	2	6	7	3	5	8	4	9
row5	48 **8**	3	58 **5**	149 **49**	149 **49**	2	15 **1**	7	6
row6	7	45 **4**	9	6	14 **1**	8	125 **5**	3	12 **2**

8. The 1 in square (65) reduces square (25) to 248.
9. Step 8 causes square (27) to have the only 1-candidate in square (27).
10. This results in two solved 1's in squares (27) and (57).

This is a contradiction because column 7 is not nine perfect.
Therefore subpattern-1 is invalid.

Next, we will test subpattern-2, composed of 5-candidates in squares (38), (13), (62), and (57):

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	3	7	258 **5**	14	28	6	9	25	14
row2	468	9	28	5	1248	3	14	26	7
row3	46	45	1	49	249	7	3	256 **5**	8
row4	1	2	6	7	3	5	8	4	9
row5	48	3	58	149	149	2	15 **5**	7	6
row6	7	45 **5**	9	6	14	8	125	3	12
row7	9	1	7	2	5	4	6	8	3
row8	5	6	3	8	7	9	24	1	24
row9	2	8	4	3	6	1	7	9	5

We'll first update the top three rows by themselves:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	3	7	258 **5**	14	28	6	9	25	14
row2	468	9	28	5	1248	3	14	26	7
row3	46	45	1	49	249	7	3	256 **5**	8

1. The 5 in square (13) reduces square (18) to a 2.
2. The 2 in square (18) makes square (28) = 6.
3. That same 2 also makes square (15) = 8.
4. That 8 in (15) reduces (25) to 124.
5. The 5 in (13) makes (32) = 4, and then (31) = 6.

6. Then the 6 and 4 in box A reduce the candidates in (21) to an 8.
7. This in turn makes (23) = 2, which makes (28) a 6.
8. That 2 in (23) also reduces the candidates in (25) to 1 and 4.
9. Now we have a pair of 14's in squares (14) and (25) in box B.
10. The 2 candidate in (35) is the only one in box B, so it becomes a solved 2.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	3	7	258 **5**	14	28 8	6	9	25	14
row2	468 **8**	9	28 **2**	5	1248 **14**	3	14	26 **6**	7
row3	46 **6**	45 **4**	1	49 **9**	249 **2**	7	3	256 **5**	8

We next update the middle three rows:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row4	1	2	6	7	3	5	8	4	9
row5	48 **4**	3	58 **8**	149 **19**	149 **19**	2	15 **5**	7	6
row6	7	45 **5**	9	6	14 **4**	8	125 **12**	3	12

11. The 5 in (62) makdes (53) an 8. This 8 makes (51) a 4.
12. And then that 4 reduces squares (54) and (55) to a pair of 19's.
13. That pair of 19's in box E reduces square (65) to a 4.
14. And finally, the 5 in square (57) reduces (67) to 12, making (67) & (69) a pair of 12's.

Now we put all the rows together, and complete the entire sudoku with ease:

15. The important first update is to use the 4 in square (65) to reduce square (25) to a 1. That makes (14) = 4, which makes (19) a 1, which makes (27) a 4, which makes (87) a 2, which makes (89) a 4.
16. The 1 in (25) makes (55) a 9, which makes (54) a 1.
17. Finally the 2 in (87) makes (67) a 1, which makes (69) a 2.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	3	7	258 **5**	14	28 8	6	9	25	14
row2	468 **8**	9	28 **2**	5	1248 **14**	3	14	26 **6**	7
row3	46 **6**	45 **4**	1	49 **9**	249 **2**	7	3	256 **5**	8
row4	1	2	6	7	3	5	8	4	9
row5	48 **4**	3	58 **8**	149 **19**	149 **19**	2	15 **5**	7	6
row6	7	45 **5**	9	6	14 **4**	8	125 **12**	3	12
row7	9	1	7	2	5	4	6	8	3
row8	5	6	3	8	7	9	24	1	24
row9	2	8	4	3	6	1	7	9	5

The result of these updates is shown on the next page. A scan of all rows, columns and boxes reveals all of them to be nine perfect. Subpattern-2 is the correct subpattern, and the solution is shown below.

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

Now I can complete the remark I made earlier, about why I became interested in subpatterns. I saw a simple relationship between squares and their candidates and patterns and their subpatterns. If I could test every possible value of a square, according to the candidates at its top, and logically eliminate every value but one, why could I not equally test every subpattern of a pattern, and logically eliminate them all but one. Then I would have a new method of solving sudokus.

qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq

There is one question remaining from the preceding solution, and that arose during the choice of the order of boxes, because if we were to make the box order C→F→D→A, it would give us the choice of two possibilities when we go from C to F.

A		C
D		F

Let's return to the subpattern analysis, using this different box order, and see what happens when we have to exercise a choice.

Box C is again the starting box, but in the new box order, F is the second box, and there are two choices, squares (57) and (67). We must choose one of them. What about the other? We must make a choice now; so we arbitrarily choose square (57). We also pencil a P into box F to remind us that a choice is pending. The square we passed up is (67), so as a further reminder we give its 5 the impossible subscript 0.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1			5_1					5_1	
row2				**X**					
row3		**5**						**5**	
row4						**X**			***P***
row5			**5**				5_1		
row6		5_1					5_0		
row7					**X**				
row8	**X**								
row9									**X**

We progress through the rest of the boxes in the order of boxes, arriving at the situation above, with two 5's in the same subpattern. This situation is unacceptable. The choice of square (57) has led to a contradiction, so we must backtrack to box F, where the pending choice was made, erase the subscript 1 from the 5 in square (57), and assign the 5 in square (67) as the next member of subpattern-1.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row4						**X**			~~*P*~~
row5			**5**				**5**		
row6		**5**					5_1		

The rest of the updates are easily calculated, resulting in the situation below. Subpattern-1 has been completed, and subpattern-2 begun in square (38). When we reach box F, we again have the two choices, (57) & (67). As before, we choose (67), and the other updates are made as before.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1			**5**					**5_1**	
row2				**X**					
row3		**5_1**						**5**	
row4						**X**			
row5			**5_1**				**5**		
row6		**5**					**5_1**		
row7					**X**				
row8	**X**								
row9									**X**

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1			**5**					**5_1**	
row2				**X**					
row3		**5_{12}**						**5_2**	
row4						**X**			***P***
row5			**5_{12}**				**5**		
row6		**5**					**5_{12}**		
row7					**X**				
row8	**X**								
row9									**X**

We again have a contradiction in row 3, so we backtrack again to the pending box F, where this time we choose square (57). This leads to the situation below.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1			5_2					5_1	
row2				**X**					
row3		5_1						5_2	
row4						**X**			
row5			5_1				5_2		
row6		5_2					5_1		
row7					**X**				
row8	**X**								
row9									**X**

This time, there is no problem, and the extensions of subpattern-1 are clear. The pending square was really a phantom. But it still had to be accounted for. Even ghosts have to be checked out. We must be absolutely certain at the end of the analysis that we have an incontestable set of all possible subpatterns. No stone unturned. No possible subpattern overlooked.

So pending squares sometimes create phony contestants, which clog up the process, multiplying the number of subpatterns to be tested. This is why we chose the original order of boxes the way we did. This does not mean that we shall always be able to avoid pending situations, because we cannot avoid them in general. Multiple choices (other than in the starting box) always result in pending situations. This is a good reason for choosing for starting box the one with the most choices, because, from a philosophical point of view, the starting box has similarities to a pending box, in that it is always pending until the calculation of subpatterns is complete. We don't mark it as such, nor do we think about it in such a way, but when we choose the order of boxes, it is best to choose the box with the most pending squares for the starting box; otherwise it will indeed become a pending box, with many choices every time we encounter it.

Exercise 1. Determine the subpatterns of the 5-pattern; then solve the sudoku with one of them. You may find it helpful to use a copy of my own solving form, which appears in the back of the book. Solution on page 37.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	2	5	13	9	36	4	7	8	16
row2	7	49	6	8	25	1	25	3	49
row3	19	1349	8	23	2356	7	1259	245	1469
row4	8	6	9	7	4	5	3	1	2
row5	4	12	5	6	12	3	8	9	7
row6	3	7	12	12	9	8	4	6	5
row7	6	239	7	4	8	29	1259	25	139
row8	19	8	123	5	7	29	6	24	349
row9	5	29	4	13	13	6	29	7	8

Exercise 2. Determine the subpatterns of the 7-pattern, then solve the puzzle with one of them. Solution on page 38.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	9	1	4	268	26	2678	237	5	36
row2	257	25	6	125	3	2579	4	8	19
row3	2578	38	23	1256	1269	4	27	179	169
row4	1	46	9	346	5	36	8	2	7
row5	25	245	8	7	249	1	6	39	34
row6	3	246	7	246	8	269	5	19	149
row7	2468	38	123	9	1246	2368	37	367	5
row8	246	9	5	2346	7	236	1	36	8
row9	68	7	13	13568	16	3568	9	4	2

Exercise 3. Determine the subpatterns of the 6-pattern; then solve the puzzle with one of them. Solution on page 39.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	**6**	**9**	**5**	**7**	**3**	**4**	**1**	**8**	**2**
row2	17	**2**	17	569	589	58	**4**	**3**	69
row3	**8**	**4**	**3**	**1**	69	**2**	**5**	69	**7**
row4	**2**	138	167	56	157	**9**	378	**4**	3568
row5	1379	138	1679	**2**	**4**	1357	3789	579	35689
row6	379	**5**	**4**	**8**	67	37	**2**	679	**1**
row7	**4**	13	**2**	59	78	**6**	3789	1579	3589
row8	1359	**6**	19	**4**	**2**	78	3789	1579	3589
row9	59	**7**	**8**	**3**	159	15	**6**	**2**	**4**

Answer to Exercise 1:

This is the analysis of the 5-pattern:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1		X							
row2				5_{12}			5_{3}		
row3					5_{3}		5_{1}	5_{2}	
row4						X			
row5			X						
row6									X
row7							5_{2}	5_{1}	
row8				X					
row9	X								

The 5-subpattern-1 solves the sudoku, and this is the solution:

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

Answer to Exercise 2:

This is the analysis of the 7-pattern:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1						7_{12}	7_{3}		
row2	7_{12}					7_{3}			
row3	7_{3}						7_{1}	7_{2}	
row4									X
row5				X					
row6			X						
row7							7_{2}	7_{13}	
row8					X				
row9		X							

The 7-subpattern-2 is the correct subpattern, and this is the solution:

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

Answer to Exercise 3:

This is the analysis of the 6-pattern:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	X								
row2				6_{12}					6_{3}
row3					6_{3}			6_{12}	
row4			6_{2}	6_{3}					6_{1}
row5			6_{13}						6_{2}
row6					6_{12}			6_{3}	
row7						X			
row8		X							
row9							X		

Subpatterns-1 &-2 fail, making subpattern-3 the correct one, and it leads to the solution below:

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

Chapter 3. Pattern Analysis.

Notation and review of nomenclature dealing with patterns

For any digit, 1-9, a ***digit pattern***, or what is often simply called a ***pattern***, is the set of all unsolved squares of a sudoku that have that digit as a candidate. This means, of course, that it excludes squares that have that digit as an established value, for such squares have no candidates. Here we prefix a digit pattern with the digit for which it is the pattern, and which is called the ***pattern digit***. For the digit 3, a 3-pattern is the set of all squares with the candidate 3. Every sudoku has up to 9 patterns, one for each digit. A completely solved sudoku has no patterns, since all of its squares have established values.

For any digit, 1-9, a ***digit subpattern***, or simply ***subpattern***, is a subset of a digit pattern, no two members of which are in the neighborhood of each other. A ***valid subpattern*** is a ***digit subpattern*** that has ***exactly one*** member in every row, column, and box of the sudoku that lacks the pattern digit as an established value. The term ***subpattern digit*** is simply a specialized use of ***pattern digit***.

Subpatterns are numbered consecutively, with ***subpattern numbers***. The **name** of a particular subpattern is composed of the term **"subpattern"**, to which are appended a hyphen and the ***subpattern number***. If there are five subpatterns in a particular pattern, then the first is called "subpattern-1," and the last, "subpattern-5."

Note that there is **no connection** between **subpattern digit** and **subpattern number**. One may speak of "subpattern-3" of the "5-pattern", or, more succinctly, of the "5-subpattern-3", meaning the third subpattern of the 5-pattern.

Preliminary comments on determining subpatterns

The method chosen in this approach is to progress box by box, according to a pre-set order of boxes, because it makes candidate patterns easier to see. Boxes are compact and two-dimensional, more easily seen in their entirety, while rows and columns are linear and extended. Two-box patterns are often too simple to be useful, though they have solved plenty of sudokus. Usually, three-box and four-box patterns work well. Patterns in more boxes are often complicated to analyze and tedious to use, because they often have six or seven subpatterns. However, when the number of subpatterns is small, the number of boxes in the pattern is not so much of a problem.

The order of boxes should make determining subpatterns as straightforward as possible; in particular, it should depend on the relationships among boxes. If two boxes have no rows or columns in common, they should not be adjacent in the progression. Each box in the sequence should follow and precede boxes with which it has rows and columns in common.

The minimum number of members of a pattern is four; the maximum perhaps 20, but such a large pattern often has several hundred subpatterns. Small patterns with a small number of subpatterns are desirable. The number of subpatterns often greatly exceeds the number of pattern members, because each pattern square can participate in multiple subpatterns. Determining the subpatterns can reveal many distinct subpatterns that vary only slightly.

In rows and columns, pattern members occur most often in pairs, frequently in threes, less often in fours, and rarely in fives or sixes, at least in practical patterns.

Pattern unthreading is versatile; it can solve almost every sudoku. To use it well, a solver should become adept at analyzing patterns and at producing accurate analyses. Accuracy requires the ability to **discover every subpattern**. Otherwise the unthreading would be inaccurate and incomplete.

It is mandatory in determining subpatterns to have a blank sudoku form at hand. Subpattern determination should never be attempted without one, except when the pattern is very simple and small.

As a subpattern grows from box to box, more and more squares become related to it, and squares that can enlarge it become fewer and fewer.

In determining subpatterns, it is extremely important to consider all possible pairings of squares. Otherwise, the subpattern needed to solve the sudoku might not be found. It is critical to try out every possible combination of pairs of squares not in each other's neighborhood. When the sudoku is solved, the subpattern that solves it must be one of those revealed by the analysis.

It is not enough to try out all possible pairings; **it is necessary to keep track of them as well.** If more than one square is found that can continue a subpattern, one of them must be chosen, but those not chosen must be noted for later attempts at inclusion.

Hierarchy of choices.

In progressing through boxes, finding all the possible subpatterns, a solver encounters boxes where more than one square may be chosen for the current subpattern. In some cases a series of boxes occurs, each one of which involves a choice. As it happens, the order of choices is important. In what order should these several choices be made? To answer this question, we can start with the following analogy.

Imagine a journey with a beginning, an end, two midpoints, and three pairs of alternate legs. Below, 1 and 2 are alternate legs, A and B are alternate legs, and x and y are alternate legs. How many different routes are there from the beginning to the end of the journey?

```
              ____1→____          ___A→____          ____x→____
beginning→ /              \→mid1→/             \→mid2→/              \→end
            \____2→_____/        \___B→_____/        \____y→_____/
```

To iterate these routes, we can start by taking leg 1, followed by leg A, followed by leg x; and call this route 1Ax. Then there are routes 1Ay, 1Bx, 1By, 2Ax, 2Ay, 2Bx, and 2By: eight routes altogether. The table below summarizes these routes:

<table>
<tr><td colspan="4">1</td><td colspan="4">2</td></tr>
<tr><td colspan="2">1A</td><td colspan="2">1B</td><td colspan="2">2A</td><td colspan="2">2B</td></tr>
<tr><td>1Ax</td><td>1Ay</td><td>1Bx</td><td>1By</td><td>2Ax</td><td>2Ay</td><td>2Bx</td><td>2By</td></tr>
</table>

In the journey, there are three points where a choice is made. We first choose 1, next A, then x. The second time we make the journey, we again choose 1, then A, but this time y, so for the first two journeys, it is the third choice that is made twice.

For the third journey, we again choose 1, next B, then x. For the fourth journey, it is 1, next B, then y. So for the next two journeys it is again the third choice that is made twice.

Therefore, for the first four journeys, we always choose 1 first, but then A followed by x, then y; then it is B followed by x, then y.

Similarly, for the last four journeys, we always choose 2 first, but then A followed by x, then y; then B followed by x, then y.

For all eight journeys, the first choice is the major one, the second choice minor to that, and the third choice minor to both. Minor choices should always be completed before choices above them in the hierarchy..

<u>This will be our principle</u>: When a series of choices is to be made, the order of boxes where the choices exist must dictate the hierarchy of choices. A box earlier in the sequence will be major, boxes later in the sequence will be minor to it. <u>Minor choices are always made first</u>.

This principle will ensure that every possible subpattern is considered.

Pending boxes:

When a pending box gives rise to multiple choices, each choice must have its own subpattern. Such subpatterns could be called ***sister subpatterns***. They share all the squares leading up to the pending box, where they diverge. It's quite possible that they meet again to share a square further on in the order of boxes. Sister subpatterns are the only subpatterns that can share a square in the starting box. Every one of them has a different subpattern number beginning in the shared square in the starting box.

When a growing sister subpattern arrives at a box that has multiple choices (and therefore becomes the new highest pending box), it gives rise to the next generation of sister subpatterns, which could be called daughter subpatterns of sister subpatterns and cousin subpatterns of one another. In this way pending boxes become the spawning grounds of subpattern family trees.

If no pending boxes occur during subpattern determination, the only sources of different subpatterns are the different pattern squares in the starting box.

The starting box and pending boxes are the only generators of different subpatterns.

Downstream is a convenient term meaning later in the order of boxes, and ***upstream*** will denote a box earlier in the order.. The **highest pending box** is always **downstream** from all other pending boxes.

In a series of pending boxes it's only the highest pending box that gets new assignments. All the other boxes, including any lower pending boxes upstream, should share the same subpattern number in the same square as the preceding subpattern until the highest pending box is complete (and no longer pending). Then the next higher pending box (always upstream) becomes the highest pending box and receives the same VIP treatment until it too is complete.

If a pending box exists, no other pending box can occur upstream from that pending box.

Since we don't test patterns that exist in more than five or six boxes, the number of pending boxes is limited to five or six. In actuality three pending boxes at any given time are the most we encounter. The number is seldom more than two at any point in the subpattern determinations.

When a new pending box is encountered downstream from the highest pending box, it becomes the highest pending box. The previous highest pending box is demoted to a mere pending box until the new highest pending box is complete (and therefore no longer pending). Then the demoted box regains its identity as highest pending box.

Square sharing:

Note again that subpatterns may share squares. Although it’s sometimes convenient to think of subpatterns as sisters, daughters, or cousins, these “relatives” are oblivious of one another. Each subpattern follows its individual logic, the **unrelatedness** of its growing set of squares.

Blind Alleys

When we arrive at a box where every square is related to the ongoing subpattern, so that there is no square to continue the subpattern, that subpattern must be abandoned as invalid. We must go back through the boxes in reverse order, erasing the subscript of the abandoned subpattern, until we arrive at either the starting box or a pending box. The latter will always be the highest pending box, and the square most recently assigned to it is **always** the cause of the blind alley. That square must be removed from the squares available in that box for the subpattern. After choosing another square to continue the subpattern we may go downstream once more.

If no eligible square remains in the pending box, that box is no longer pending. We must erase the pending P and continue back through the boxes, erasing the subscript of the invalid subpattern until we reach the starting box or the next pending box (which is now the highest pending box). We then act as before, selecting another eligible square for the subpattern and resuming our forward journey through the boxes.

If we reach no pending box in our backwards journey, we will end at the starting box, where the subpattern number must be erased from the square where the subpattern began. That square may not begin any further subpatterns.

If another square in the starting box is still unassigned, *that* square must start a new subpattern, which should receive the same number as the abandoned subpattern (so no gap occurs in the sequence of subpattern numbers).

If, when we return to the starting box, no square remains to start a new subpattern, we erase the invalid subpattern's number from its starting square. We will have completed all the subpatterns for this pattern digit.

Proof that this strategy discovers all subpatterns:

Could there be a subpattern which this strategy ***doesn't*** discover?
Let's call it a ***spurious*** subpattern.
If there ***were*** such a subpattern,
it would have to have a member in every box.
In particular, it would have to have a member in the starting box.
Say that its square was ***removed*** from the starting box
at the end of the analysis for want of a completed subpattern
starting with that square.
This means that square had to have been tested by the analysis
and found to lack an extension in one of the subsequent boxes.
Let's look at that box.
The spurious subpattern has a square in it,
and that square ***must be connected*** to the spurious subpattern,
so it was available to it during the process of analysis as well.
Therefore the rules of analysis were not followed.
This is a contradiction to the assumption that those rules ***were***
followed.
Therefore there ***had to have been such an extension***.
The same is true of all subsequent extensions of the spurious
subpattern.
All of its squares ***had to have been available*** to the original analysis.
Therefore the process ***had to have discovered*** this spurious subpattern.
Otherwise the initial square of this spurious subpattern
was ***not removed*** from the starting box, which could only happen if
the spurious subpattern ***were*** discovered by the analysis.
Therefore all subpatterns are discovered by this analysis.

Any box may be chosen as the starting box.

If two different starting boxes led to different sets of subpatterns, then by the proof above, neither starting box could reveal a subpattern not discovered by the other.

You can't change the starting box.

The importance of the starting box is that the proof offered in this section depends on the starting box never changing during the subpattern analysis. The reason for this is that the starting box is our method of accounting for all the subpatterns. Imagine that the sudoku form is the map of a state, that each box represents a county, that the pattern squares represent hotels in that county, and that a subpattern number represents a particular individual who always frequents the same hotel when he is in that county. In one county, an individual stays at a rooming house with only one room for rent, and in another county stays at an hotel at which other travelers also stay. Our purpose is to identify all the traveling individuals in the state, and we do it by canvassing all the hotels and rooming houses in one particular county, chosen at random, making a head count in each. If we don't stick to one county, but do a partial head count in one county, and then switch to another to complete the count, it clearly isn't going to work, because some pairs of travelers have rooms in the same hotel in one county and rooms in different hotels in another county.

The point of this analogy is that we can't start to count subpatterns in one pattern box, and then switch our survey mid-way to another pattern box

When a return to the starting box is made, and there are no pending conditions, then the order of boxes may be changed.

First of all, any two squares are independent of one another if they share no subpatterns. Similarly, any two sets of squares are independent of one another if they share no subpatterns.

Secondly, any two box orders produce the same set of subpatterns from the same set of squares in the starting box.

Suppose box order #1 were used on one set of squares in the starting box, completing the subpatterns for these squares, in that no pending conditions remained.

Then we propose to change the order of boxes to box order #2. We could pretend that we had actually used box order #2 for the first set of squares. After all, we'd have ended up with the same set of subpatterns, anyway. So continuing to use box order #2 wouldn't constitute a change. We'd get the same result. So in this case it's legal to pretend.

The remainder of this chapter will be spent on pattern analysis alone, with no attempt to use subpatterns to solve sudokus. We'll next begin the analysis of the 1-pattern below:

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

We'll choose the order of boxes as A→C→G→I.

A		C
G		I

In the starting box A, there are three choices for the first member of subpattern-1. We'll choose square (21). In the second box C, square (29) is in the same row as (21), so the only unrelated square in box C is (39), which becomes the second member of subpattern-1. See the diagram on the next page.

A		C
G		I

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1				**X**					
row2	**1_1**	**1**							**1**
row3	**1**								**1_1**
row4								**X**	
row5					**X**				
row6			**X**						
row7			***P1***			**X**			
row8	**1**	**1_1**					**1**		
row9		**1_0**					**1_1**		

In box G, there are two unrelated squares, (82) and (92), so we choose (82) as the third member of subpattern-1, penciling in a P1 (for "pending"), and giving square (92) the subscript 0 to indicate that it is the next pending square in box G.

In box I, only one square, (97), is unrelated to subpattern-1, and it becomes the fourth member of subpattern-1.

We now begin subpattern-2 in box A. Because we have a pending situation, we choose the **same** square (21) as the first member of subpattern-2. We do likewise in box C. In box G, the pending box, the pending square (92) becomes the third square of subpattern-2. We replace the zero subscript with a 2 and erase the pending P1. In box I, the only square unrelated to subpattern-2 is (87), which becomes the final member of subpattern-2.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1				**X**					
row2	$\mathbf{1_{12}}$	$\mathbf{1_{3}}$							$\mathbf{1_{4}}$
row3	$\mathbf{1_{4}}$								$\mathbf{1_{123}}$
row4								**X**	
row5					**X**				
row6			**X**						
row7			***P2***			**X**			
row8	$\mathbf{1_{3}}$	$\mathbf{1_{14}}$					$\mathbf{1_{2}}$		
row9		$\mathbf{1_{20}}$					$\mathbf{1_{134}}$		

Back at the starting box A, we choose the unassigned square (22) as the first member of subpattern-3. Its extension in box C is (29). In G only (81) is unrelated to subpattern-3, and it becomes the third member. In I only (97) is unrelated to subpattern-3, and it completes the subpattern..

Back at A (31) remains unassigned; so it becomes the first square of subpattern-4. The only unrelated square in C is (29), which becomes the second member of subpattern-4.

In box G are two unrelated squares, (82) and (92), so we must again mark this box as pending, penciling in a P2. We choose square (82) as the third square of subpattern-4, adding the subscript zero to the pending square. The extension of subpattern-4 in box I is (97). This completes subpattern-4.

Because G has a pending square, we need a new subpattern. We start subpattern-5 in the same square as the preceding subpattern-4, square (31). In C only (29) is unrelated to the new subpattern-5; so it becomes the second member. Next, in the pending box G we continue subpattern-5 in square (92), replacing the zero subscript with the subscript 5 and erasing the P2. In the final box, only (87) is unrelated to subpattern-5; so it becomes the final member of subpattern-5. The starting box has no unassigned squares nor do any of the other boxes. No zero subscripts remain. We are done.

Final analysis of the 1-pattern:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1				X					
row2	1_{12}	1_{3}							1_{45}
row3	1_{45}								1_{123}
row4								X	
row5					X				
row6			X						
row7			~~P2~~			X			
row8	1_{3}	1_{14}					1_{25}		
row9		1_{25}					1_{134}		

In the next example we have a five-box 9-pattern. We choose box E as the starting box, because it has the most candidates, and we make the order of boxes E→D→G→H→B.

	B	
D	E	
G	H	

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

We make square (54) the first square of subpattern-1. In the second box, D, square (53) is in the same row as (54); so the only unrelated square is (61), which becomes the second member of subpattern-1. In G (91) is in the same column as (61); so (73) becomes the third member. In H (74) and (76) are related to (73), but (95) is unrelated, and it becomes the fourth member. Finally in B (25) is related to (95); so (96) becomes the fifth and last member of subpattern-1. We next begin subpattern-2 in square (56). Its extensions in boxes D, G & H are all forced. But box B has no unrelated square, so square (56) may not be the beginning of subpattern-2. But then, since all the updates for ***any*** subpattern which starts in (56) will encounter the same problem, it is evident that square (56) cannot be the beginning of any subpattern. We'll give it the subscript K (for killer) to remind us to remove the candidate 9 from square (56).

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1		X							
row2					9_2	9_1			
row3							X		
row4									X
row5			9	9_1		9_2			
row6	9_{12}			9	9	9			
row7			9_{12}	9		9			
row8								X	
row9	9				9_{12}				

We must remove all the subscripts 2. We backtrack from square (95) in box H, removing the 2 subscripts from (95), (73), (61), and finally (56). Now we must start a new subpattern-2 from another square in E. We'll choose square (64) as the new beginning of subpattern-2. Square (61) is the only extension in D, and (91) the only extension in G, In H, because (74) is in the same column as (64) and (95) in the same row as (91), the only choice is (76), which becomes the fourth member. This makes (25) the only choice in box B, and (25) completes subpattern-2.

Since subpattern-2 was forced, there are **no pending boxes.** <u>Therefore we may change the order of boxes</u>. We shall make the new order E→B→H→G→D

	B	
D	E	
G	H	

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1		X							
row2					9_{2}	9_{1}			
row3							X		
row4									X
row5			9_{23}	9_{1}		9_{K}			
row6	9_{1}			9_{2}	9_{3}	9			
row7			9_{1}	9		9_{2}			
row8								X	
row9	9_{23}				9_{1}				

We can't start any subpattern with (56), because any extension of (56) will create the same impasse as the previously aborted subpattern-2. So we'll begin subpattern-3 in square (65). Its extension in D is (53), and in G it is (91). Box H, however, appears to hold two choices, (74) and (76). What to do? Let's glance downstream to box B. We can foresee that (26) will continue subpattern-3 there; so (23) forestalls the choice of (76) in H. Thus the only continuation in H is (74), which completes subpattern-3.

The order of boxes is E→D→G→H→B.

	B	
D	E	
G	H	

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1		X							
row2					9_{24}	9_{13}			
row3							X		
row4									X
row5			9_{23}	9_1		9_K			
row6	9_1			9_2	9_3	9_4			
row7			9_1	9_{34}		9_2			
row8								X	
row9	9_{234}				9_1				

Now we start a new subpattern-4 in the remaining square (66). Its extension in box D is (53); in box G, (91); and in box H, (74). Finally, in box B, it is (25). We have found four subpatterns, and they complete the search.. Back at the starting box, we see that square (56) is not a member of any pattern. It is what I call a **killer candidate**, and it receives the subscript K to remind us to remove it from square (56) of the sudoku we are solving.

Killer Candidates

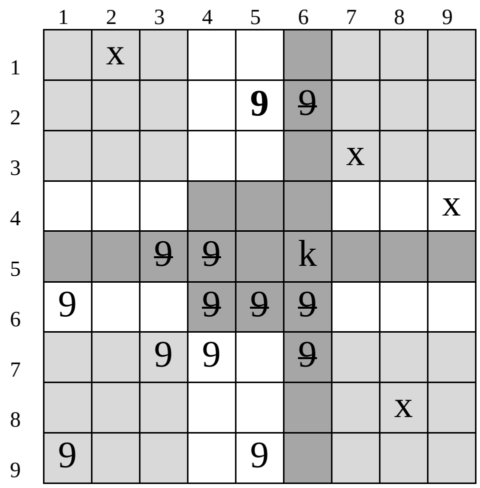
The term ***killer candidate*** needs explanation. One might be tempted to argue that any impermissible candidate will cause violations of the nine-perfect rule, including all the candidates removed while solving, which might also be called killers.

But that argument fails. A killer candidate is impermissible in a **pattern**, and every pattern is **abstract**, not limited to any one sudoku. A killer candidate resembles the candidates removed by n-wings. Its existence is a matter of pattern **geometry**.

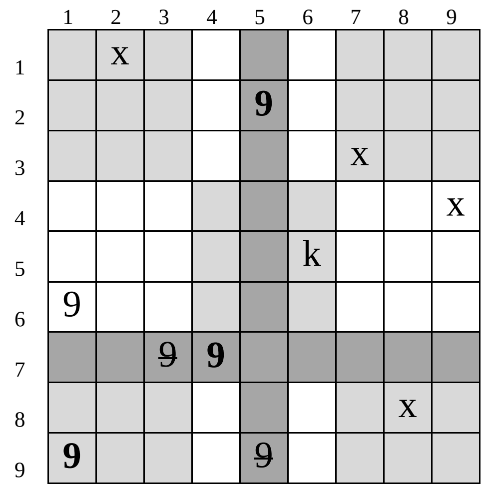
There is an easy way to verify that a candidate is a killer, a way that depends not on the sudoku itself but **only on the pattern.** On a blank sudoku form, identify the killer candidate with a k (here, short for a 9_k), into the pattern, and remove all other subscripts.

Now lets make the killer candidate the established value for its square and check the results. First, it eliminates all the 9-candidates in its neighborhood (box, row, and column). These eliminations result in single candidates that eliminate other 9's.

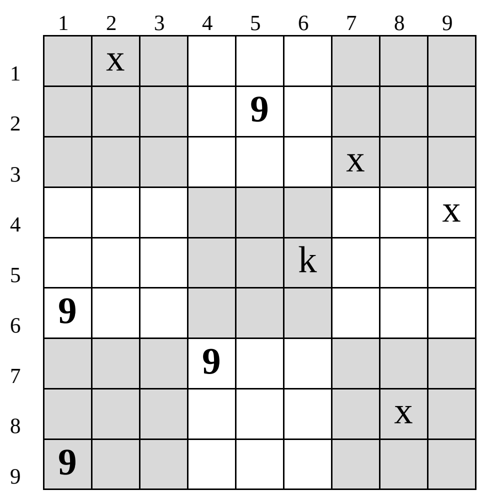
Step 1



Step 2



Step 3



The eliminations end in a standoff between the nines in column 1, because eliminating either one creates a box with no 9-candidate, either box D or box G. If you try this procedure with any normal subpattern candidate, you'll discover it creates no such violations.

A serendipitous effect of identifying a killer candidate is that removing it from its square in the sudoku. can occasionally, by itself, lead to the solution.

Solving by eliminating a killer candidate:

This is a wrapup in solving a sudoku. The 6-pattern looks do-able, even though there are only two solved squares, (18) and (65).

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	1	7	35	2	35	8	9	6	4
row2	9	24	56	1	57	346	27	3	8
row3	36	24	8	67	9	346	257	1	57
row4	36	1	2	4	8	5	36	7	9
row5	8	36	7	9	1	2	356	4	356
row6	5	9	4	3	6	7	1	8	2
row7	7	36	9	8	2	1	4	5	36
row8	4	5	36	67	37	9	8	2	1
row9	2	8	1	5	4	36	367	9	367

This is the 6-pattern. On the next page we'll try determining its subpatterns.

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

A	B	
D		F
G	H	I

We choose B as the starting box, (26) as the beginning square of subpattern-1, and the intial box order B→A→G→D→H→I→F

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1								X	
row2			6_2			6_1			
row3	6_1			6_2		6			
row4	6_2						6_1		
row5		6_1					6_2		6
row6					X				
row7		6_2	→	→	→	→	→	→	6_{12}
row8			6_1	6					
row9						6_{12}	6		6

We'll start subpattern-2 in square (34), with the slightly different box order B→A→D→G→H→I→F. Every choice in the sequence of boxes is forced, yet the members of subpattern-2 in boxes G and I are in the same row. This is a contradiction, and it means that square (34) cannot begin a subpattern. It is therefore a killer, so the 6-candidate in square (34) must be removed from the pattern. On the next page, we'll see what the results of this removal are.

Removing the 6-candidate from square (34) leaves the single candidate 7, which, since it is the only candidate, **must be promoted**, which we indicate by moving it to the center of the square and giving it a large numeral. Square (34) is shaded dark to make it stand out. When we go to update the surrounding squares, and then go on to update the rest of the sudoku, we discover that this single 7 solves the entire sudoku all by itself! And in only 27 simple steps, like a row of cascading dominos. Check this out yourself, as it is a good exercise in updating a sudoku. Note that this is an example of a single killer candidate solving an entire sudoku.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	1	7	35	2	35	8	9	6	4
row2	9	24	56	1	57	346	27	3	8
row3	36	24	8	**7**	9	346	257	1	57
row4	36	1	2	4	8	5	36	7	9
row5	8	36	7	9	1	2	356	4	356
row6	5	9	4	3	6	7	1	8	2
row7	7	36	9	8	2	1	4	5	36
row8	4	5	36	67	37	9	8	2	1
row9	2	8	1	5	4	36	367	9	367

Two alternate pattern analyses:

The following 2-pattern shows the established 2's as large X's:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1			X						
row2				X					
row3								X	
row4		X							
row5					2		2		
row6						2	2		2
row7	2				2	2	2		2
row8	2				2				2
row9						2	2		

Since the pattern exists only in the lower six rows, we shall, for simplicity, remove the top three rows from the discussion which follows on subsequent pages. This discussion will have two parts, for two different subpattern evaluations, called "method 1" and "method 2," proceeding from different box orders, each "method" having a different starting box.

In the first method, the same box order will be used for the first three subpatterns, and the fourth subpattern by a different box order. This is possible because there are no pending squares.

In the second method, the same box order will be used throughout, as the choice of starting box results in pending situations, which do not allow changes in box order.

Method 1: Starting box is I. The order of boxes is chosen to be I→G→H→E→F. Square (77) is chosen as the starting square.

	E	F
G	H	I

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row4		**X**							
row5					2_{12}		2		
row6						2	2_{2}		2_{1}
row7	2				2	2	2_{1}		2_{2}
row8	2_{12}				2				2
row9						2_{12}	2		

Using this order of boxes, every choice for subpattern-1 is forced, obtaining the results above. This box order has been specially chosen so as to avoid pending boxes.

Note that in going from box G to box H, row 7 in H is excluded by the subpattern-1 member in square (77) in I, and row 8 in H is excluded by the subpattern-1 member in square (81) in G. leaving square (96) the only available square in H for subpattern-1.

Going from box H to box E, column 6 is excluded by the subpattern-1 member in square (96), and in going from box E to box F, column 7 is exclude by the subpattern-1 member in square (76).

For subpattern-2, We choose square (79) as the first square of subpattern-2, As before, every choice in this order of boxes is forced, by almost the same logic as for subpattern-1, with column 9 replacing the role of column 7.

Subpattern-3 will be started in square (89), with the same order of boxes, I→G→H→E→F. Every extension of subpattern-3 is forced, in the same manner as for the first two subpatterns.

	E	F
G	H	I

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row4		**X**							
row5					2_{1234}		2_{K}		
row6						2_{K}	2_{23}		2_{14}
row7	2_{3}				2_{K}	2_{4}	2_{1}		2_{2}
row8	2_{124}				2_{K}				2_{3}
row9						2_{123}	2_{4}		

For subpattern-4, since there are no pending boxes, we are allowed to change the order of boxes to I→F→E→H→G. There is only one unassigned square, (97), so it becomes the first member of subpattern-4. All extensions of subpattern-4 are forced, with the results as above.

Since all the squares of the starting box I have been included in subpatterns, we are done, and all remaining unselected candidates must be killers, so we give them all the subscript K.

Method 2 is shown on the next page.

Method 2. This time, the starting box is E. The box order will be E→F→I→G→H. We shall choose square (55) as the first square of subpattern-1.

	E	F
G	H	I

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row4		**X**							$\boldsymbol{P_1}$
row5					2_1		2		
row6						2	2_1		2•
row7	2				2	2	2		2_1
row8	2_1				2				2•
row9						2_1	2		$\boldsymbol{P_2}$

There are two choices in box F for the second member of subpattern-2, namely (67) and (69). We'll choose (67), and mark the box as pending. We also put a small dot to the right of the 2-candidate in square (69) to indicate that it will be the next choice in pending box F.

The next box in the order of boxes is I. There are two choices, squares (77) and (97). We choose (77) as the third member of subpattern-1, mark box I as pending, and append a dot to the other choice, and insert a pending P2.

The next box is G, and there is only one choice, square ((81).

The final box is H, and again there is only one choice, square (96). This completes subpattern-1. Subpattern-2 is dealt with on the next page.

The box order is still E→F→I→G→H. It may not be changed, because of the pending situation.

	E	F
G	H	I

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row4		**X**							$\boldsymbol{P_1}$
row5					2_{12}		2		
row6						2	2_{12}		2•
row7	2_2				2	2	2		2_1
row8	2_1				2				2_2
row9						2_{12}	2		$\cancel{P_2}$

Back at the starting box E, we note there is at least one pending situation, so we begin the third subpattern in the same square as the second subpattern, square (55), by adding the subscript 2. The next box, F, is not the highest pending box, so we add the subscript 2 to the same square as for subpattern-1

The next box, I, is the highest pending box, so we choose the remaining square, namely square (89), which is marked with a dot. We remove the dot, assign the subscript 2 to the 2-candidate in box (89). We also erase the pending P2.

The next box is G, there is only one choice, square (71), and in the final box, H, there is again only one choice, square (96), which becomes the final member of subpattern-2.

On the next page, we shall calculate subpattern-3

Since there is a pending situation in box F, we must begin subpattern-3 in the same beginning square in the starting box E as subpattern-2. The order of boxes is unchanged, still E→F→I→G→H.

The extension of subpattern-3 in box F is the remaining pending square, (69), which is marked with a dot. We replace the dot with the subscript 3. We also erase the pending P1.

	E	F
G	H	I

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row4		**X**							$\cancel{P_1}$
row5					2_{123}		2		
row6						2	2_{12}		2_3
row7	2_2				2	2	2_3		2_1
row8	2_{13}				2				2_2
row9						2_{123}	2•	P_1	

In box I, there are two choices for subpattern-3, namely squares (77) and (97). We choose square (77), place a dot next to the other choice, and mark the box with a new pending P1.

There is only one choice in the next box G, namely square (81), and again only one choice in box H, namely square (96). This completes subpattern-3.

Back at the starting box E, there is a pending condition, so we begin subpattern-4 in the same square as subpattern-3, namely square (55). This is shown on the next page.

Going to the next box in the order of boxes, namely box F, we note the higher pending condition in box I, so we choose the same square (69) as for subpattern-3 for the second member of subpattern-4. When we reach the pending box I, we choose the remaining pending square (97), replace the dot with the subscript 4, and erase the pending P1. The order of boxes is still E→F→I→G→H.

	E	F
G	H	I

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row4		**X**							
row5					2_{1234}		2		
row6					↓	2	2_{12}		2_{34}
row7	2_{24}	→	→	→	2	2	2_{3}		2_{1}
row8	$2_{13\bullet}$				2				2_{2}
row9		$\cancel{P_2}$				2_{123}	2_{4}	$\cancel{P_1}$	

In box G, there are two choices, namely squares (71) and (81). We choose square (71) as the next member of subpattern-4, place a dot next to the other choice, and mark the box with a pending P2.

But now, when we go to the final box H, there is no available square for subpattern-4, since squares (75) and (76) are in the same row as (71) in box G, square (85) is in the same column as square (55) in box E, and square (96) is in the same row as (97).

Therefore the extension in box G, in square (71) is not valid, so we cannot choose square (71) as the 4th member of subpattern-4, and instead choose square (81) and erase the pending P2, since there is only one choice, and it has now been made. See the next page.

Back at box G, where subpattern-4 has been continued in square (81), when we progress to the final box, H, there is only one choice for the extension of subpattern-4, namely square (76), since squares (75) & (85) share the same column as (55), and since square (96) shares the same row as (97).

	E	F
G	H	I

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row4		**X**							
row5					2_{1234}		2_{5}		
row6						2_{5}	2_{12}		2_{34}
row7	2_{2}				2	2_{4}	2_{3}		2_{15}
row8	2_{1345}				2				2_{2}•
row9						2_{123}	2_{4}		$\boldsymbol{P_1}$

Back at the starting box, there is no pending condition, and square (66) has not yet been assigned to a subpattern, we must begin a new subpattern-5 in square (66). Its extension in box F is forced, in square (57).

There are two choices in box I, namely squares (79) and (89). We'll choose (79), mark (89) with a dot to show it is a pending square, and mark box I with a pending P1.

In box G, there is only one available square for subpattern-5, namely square (81). But now when we go to the final box H, there is no square to continue subpattern-5, since both (75) & (85) are in the same column as (55), and both (76) & (96) are in the same column as (66).

Therefore there is no extension of subpattern-5 in box G, since both choices in G have no continuation in box H. Therefore subpattern-5 cannot be started in square (66), since it cannot be completed. We must therefore remove the subscript 5 from square (66). We have now completed all the 2-subpatterns, and we have the following situation:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row4		**X**							
row5					2_{1234}		$2_{\mathbf{K}}$		
row6						$2_{\mathbf{K}}$	$2_{\mathbf{12}}$		$2_{\mathbf{34}}$
row7	$2_{\mathbf{2}}$				$2_{\mathbf{K}}$	$2_{\mathbf{4}}$	$2_{\mathbf{3}}$		$2_{\mathbf{15}}$
row8	$2_{\mathbf{134}}$				$2_{\mathbf{K}}$				$2_{\mathbf{2}\bullet}$
row9						$2_{\mathbf{123}}$	$2_{\mathbf{4}}$		$\boldsymbol{P_1}$

As before, the 2-candidates with no subscripts are all killers, and should be given the K-subscript to indicate that they are killers.

How do we deal with this? Clearly, the 2-candidate in square (55) is an established value, and, in the way established values are shown in subpattern diagrams, should be replaced with a large X.

Although the top three rows of the pattern were not shown in the above discussions, they are resurrected below for the sake of clarity:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1			**X**						
row2				**X**					
row3								**X**	
row4		**X**							
row5					**X**				
row6							$2_{\mathbf{12}}$		$2_{\mathbf{34}}$
row7	$2_{\mathbf{2}}$					$2_{\mathbf{4}}$	$2_{\mathbf{3}}$		$2_{\mathbf{1}}$
row8	2_{134}								$2_{\mathbf{2}}$
row9						2_{123}	$2_{\mathbf{4}}$		

In the sudoku for which this is the 2-pattern, the squares with killer candidates will have had their 2-candidates removed.

The results of Method 1 have a similar result (see the sudoku below).

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1			**X**						
row2				**X**					
row3								**X**	
row4		**X**							
row5					**X**				
row6							2_{23}		2_{14}
row7	2_{3}					2_{4}	2_{1}		2_{2}
row8	2_{124}								2_{3}
row9						2_{123}	2_{4}		

Note that the final results of the two methods have the digits 2 & 3 reversed. This is not important, as the subscripts are simply aspects of the naming convention.

The Law of Total Subscripts.

The subscripts in every row, column, and box always include every subscript once and only once. This stems from the fact that every row, column and box must represent every subpattern, and no subpattern can have more than one member in each row, column, or box.

This is an important fact, and c**an be used to check the calculated subpatterns**. If a subscript is **missing** from any row, column, or box, the calculated subpattern is **in error**.

Let us look at the following pattern solution from page 51:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1		X							
row2					9_{24}	9_{13}			
row3							X		
row4									X
row5			9_{234}	9_{1}		9_{K}			
row6	9_{1}			9_{2}	9_{3}	9_{4}			
row7			9_{1}	9_{34}		9_{2}			
row8								X	
row9	9_{234}				9_{1}				

In row 2, the subpattern members are 9_{24} **&** 9_{13}

In row 5, the subpattern members are 9_{234} **&** 9_{1} ,

In row 6, the subpattern members are 9_{1}, 9_{2}, 9_{3} **&** 9_{4}

In row 7, the subpattern members are 9_{1}, 9_{34} **&** 9_{2}

In row 9, the subpattern members are 9_{234} **&** 9_{1}

In column 1, the subpattern members are 9_{1} **&** 9_{234}

In column 3, the subpattern members are 9_{234} **&** 9_{1}

In column 4, the subpattern members are 9_{1}, 9_{2} **&** 9_{34}

In column 5, the subpattern members are 9_{24}, 9_{3} **&** 9_{1}

In column 6, the subpattern members are 9_{13}, 9_{4} **&** 9_{2}

In box B, the subpattern members are 9_{24} **&** 9_{13}

In box C, the subpattern members are 9_{234} **&** 9_{1}

In box D, the subpattern members are 9_{1}, 9_{2}, 9_{3} **&** 9_{4}

In box G, the subpattern members are 9_{1} **&** 9_{234}

In box H, the subpattern members are 9_{34}, 9_{2} **&** 9_{1}

In every case, the subscripts include every subscript from 1 to 4 once and only once. Check out the subscript law in the other patterns we've solved so far.

Note also that this is not a proof of subpattern analysis. Consider the two 5-patterns on the next two pages. Only one of them is correct, but you couldn't tell it using this law.

Warning – Look at the pattern analysis below and compare it to the one on the next page:

Pattern analysis 1:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1									
row2	5_2						5_{13}		
row3	5_1		5_3					5_2	
row4									
row5	5_3		5_{12}						
row6									
row7		5_3		5_1			5_2		
row8				5_2				5_{13}	
row9		5_{12}		5_3					

Pattern analysis 2:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1									
row2	5_{1}						5_{2345}		
row3	5_{23}		5_{45}					5_{1}	
row4									
row5	5_{45}		5_{123}						
row6									
row7		5_{35}		5_{24}			5_{1}		
row8				5_{1}				5_{2345}	
row9		5_{124}		5_{35}					

In both cases, the subscript law holds. Which one is correct?

The important answer here is that the subscript law by itself is not a proof that a pattern analysis is correct. It is only useful for spotting one that isn't. As an exercise, calculate the subpatterns and find out for yourself which one is the right one.

Derivative boxes:

When a pattern box shares rows or columns with **exactly one other pattern box**, then it could be called ***derivative*** in that, using the law of total subscripts, the subscripts of its subpattern members can be entirely deduced from those in the squares it is derivative to. It cannot have more than one square in the shared row or column.

Example 1: In this pattern, it is clear that the subscripts of the 4's in box A can be seen as entirely derivative to those of the 4's in box G. Similarly, the subscripts of the 4's in box E are entirely derivative to the 4's in box H.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1									
row2									
row3	*4*	*4*							
row4									
row5				*4*		*4*			
row6									
row7	4					4			
row8	4								4
row9		4		4		4			4

We can therefore calculate the subpatterns using only the bottom three boxes: We'll make box G the starting box, with the order of boxes G→H→I

G	H	I

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row7	4_{12}					4			
row8	4				*P*				4
row9		4		4_1		4_0			4

We'll choose square (71) as the first square of subpattern-1. There are two choices in box H, (94) and (96). We'll choose (94) as the second square of subpattern-1, noting that (96) is the pending square, and marking box H as pending. In box I, there is only one choice for subpattern-1, and that is square (89).

Returning to the starting box G, we note the pending condition, so we begin subpattern-2 in the same starting square (71), and in the pending box we now choose square (96) as the second square of subpattern-2, erasing the pending P. The extension of subpattern-2 in box I is the same square as for subpattern-1.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row7	4_{12}					4_{34}			
row8	4_{3}				~~P~~				4_{124}
row9		4_{4}		4_{1}		4_{2}			4_{3}

Back at the starting box G, there is no pending situation, so we choose the next available square (81) for the start of subpattern-3. Since there is no pending square, we may change the order of boxes to G→H→I. There is only one unrelated square to subpattern-3, namely (99). Its extension in box H is square (76).

Back at the starting box, there is still one available square, (92), which we make the first square of subpattern-4. Following the same order of boxes, the extensions of subpattern-4 are squares (89) in I and (76) in H. We have finished. Almost. Now we must put the entire pattern back together.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1									
row2									
row3	*4*	*4*							
row4									
row5				*4*		*4*			
row6									
row7	4_{12}					4_{34}			
row8	4_{3}								4_{124}
row9		4_{4}		4_{1}		4_{2}			4_{3}

Using the law of total subscripts, we calculate the subscripts for the pattern members in boxes A & E.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1									
row2									
row3	4_{4}	4_{123}							
row4									
row5				4_{234}		4_{1}			
row6									
row7	4_{12}					4_{34}			
row8	4_{3}								4_{124}
row9		4_{4}		4_{1}		4_{2}			4_{3}

Note that box I is not derivative to a single box, but to two boxes, G & H. It cannot be dealt with in a similar manner because it enters into the logic of boxes G & H, and is not a passive player.

Example 2.

This pattern exists only in the top six boxes. In this case there are two derivative boxes, plus a derivative of a derivative. The entire pattern can be computed just using box D.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1		1	1						
row2									
row3								1	1
row4		1	1					1	
row5									
row6		1	1						1

	col 1	col 2	col 3
row4		1	1
row5			
row6		1	1

The pattern is extremely easy. The four squares are independent of each other, so the pattern is:

	col 1	col 2	col 3
row4		1_1	1_2
row5			
row6		1_3	1_4

Pasting the pieces back together, we have:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1		1	1						
row2									
row3								1	1
row4		1_1	1_2					1	
row5									
row6		1_3	1_4						1

Using the law of total subscripts, the subscripts for boxes A and F follow immediately from the subscripts in box D. The subscripts for box C follow from those in box F.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1		1_{24}	1_{13}						
row2									
row3								1_{12}	1_{34}
row4		1_1	1_2					1_{34}	
row5									
row6		1_3	1_4						1_{12}

Be very careful in using this method. It can sometimes be extended to more complex situations, but there are many pitfalls. For simple situations, it can be useful.

Remember the guidelines: One derivative square for row or column, one box determining the derivative square.

Pattern Problem No. 1

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

Pattern Problem No. 2

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

Pattern Problem No. 3

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

Pattern Problem No. 4

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

Pattern Problem No. 5

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

Pattern Problem No. 6

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

Pattern Problem No. 7

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

Pattern Problem No. 8

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

Pattern Problem No. 9

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

Pattern Problem No. 10

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

Pattern Problem No. 11

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

Pattern Problem No. 12

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

Pattern Solution No. 1

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1									
row2									
row3									
row4					4_{12}	4_{3}			
row5							4_{2}		4_{13}
row6									
row7					4_{3}	4_{1}			4_{2}
row8						4_{2}	4_{13}		
row9									

Pattern Solution No. 2

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	9_{1}			9_{23}					
row2			9_{23}	9_{1}					
row3									
row4									
row5									
row6									
row7					9_{3}		9_{12}		
row8	9_{2}		9_{1}				9_{3}		
row9	9_{3}				9_{12}				

Pattern Solution No. 3

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1									
row2									
row3							1_{4}		1_{123}
row4	1_{1}		1_{2}			1_{34}			
row5			1_{34}				1_{12}		
row6				1_{12}			1_{3}		1_{4}
row7	1_{234}		1_{1}						
row8				1_{34}		1_{12}			
row9									

Pattern Solution No. 4

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1									
row2	5_{1}	5_{23}							
row3									
row4	5_{23}		5_{1}						
row5									
row6									
row7		5_{1}	5_{2}						5_{3}
row8			5_{3}						5_{12}
row9									

Pattern Solution No. 5

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1									
row2				9_{1}	9_{2}	9_{3}			
row3									
row4						9_{12}	9_{3}		
row5									
row6				9_{3}			9_{12}		
row7									
row8									
row9				9_{2}	9_{13}				

Pattern Solution No. 6

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1		1_{12}			1_{3}				
row2		1_{3}		1_{1}	1_{2}				
row3									
row4	1_{1}			1_{23}					
row5	1_{23}				1_{2}				
row6									
row7									
row8									
row9									

Pattern Solution No. 7

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1									
row2	1_{1234}		1_5						
row3									
row4						1_4	1_3		1_{125}
row5						1_{1235}	1_4		
row6									
row7	1_5		1_1				1_2		1_{34}
row8									
row9			1_{234}				1_{15}		

Pattern Solution No. 8

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	7_{1235}	7_{46}							
row2									
row3				7_{23}	7_{1456}				
row4									
row5									
row6									
row7		7_{123}		7_{456}					
row8	7_{6}				7_{3}				7_{1245}
row9	7_{4}	7_{5}		7_{1}	7_{2}				7_{36}

Pattern Solution No. 9

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1									
row2		8_{5678}						8_{1234}	
row3	8_{1234}						8_{57}		8_{68}
row4	8_{56}			8_{3478}			8_{1}		8_{2}
row5		8_{1234}						8_{5678}	
row6	8_{78}			8_{1256}			8_{3}		8_{4}

Pattern Solution No. 10

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1				2_{123}	2_{45}		2_{67}		
row2									
row3					2_{67}		2_{12345}		
row4		2_{1357}	2_{246}						
row5									
row6									
row7		2_{246}	2_{357}		2_{1}				
row8									
row9			2_{1}	2_{4567}	2_{23}				

Pattern Solution No. 11

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1									
row2									
row3									
row4	1_{123}				1_{456}				
row5							1_{14}	1_{2356}	
row6	1_{456}				1_{123}				
row7			1_{36}				1_{25}	1_{14}	
row8			1_{1245}				1_{36}		
row9									

Pattern Solution No. 12

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	4_{1}		4_{23}					4_{4567}	
row2	4_{4}		4_{56}	4_{1237}					
row3	4_{7}			4_{456}				4_{123}	
row4									
row5									
row6									
row7									
row8	4_{25}		4_{147}		4_{36}				
row9	4_{36}				4_{12457}				

Chapter 4. Choosing the right subpattern.

Reducing the number of candidates.

In everything that follows, it is assumed that the sudoku being assesed for possible subpattern analysis has been reduced to the point of having only a small number of candidates in most squares. Otherwise, although subpattern analysis can still often solve the sudoku, it may on occasion entail a great deal of drudgery. It is therefore best to stick to the tried-and-true methods normally used to reduce the number of candidates. It is at the end of such a process that a subpattern can best be used to solve the sudoku.

N-wings

Reminder: For convenience, x-wings and their higher siblings (swordfish, jellyfish, and squirmbags) are lumped together under the single term ***n-wings***. The identification of, and use of n-wings is a mandatory first step to complete before using subpattern analysis.

Using an extra sudoku form

Since patterns are so important in this approach, I recommend the use of two sudoku forms, side by side, the right one for the sudoku itself, and the left one for patterns. Included on the website is a template for the sudoku form I use for solving, including a ***distribution map*** (see next paragraph). I always transfer the sudoku from the website or newspaper or puzzle book to this form. It is important to be able to erase candidates as they are eliminated, and newspapers, for one, do not lend themselves to this. It is also far easier to study patterns on a separate form, especially extensive patterns.

Distribution map

At the bottom of the sudoku form (see above) is a set of nine small 3x3 digit grids, one grid for each of the nine digits 1-9. Each grid contains nine small cells, one for each box of the sudoku. When a cell contains a digit, it means the corresponding box of the sudoku contains that digit as an established value. This is called a ***distribution map***. It is given its initial values from the givens, and updated every time a square is solved. It is a record of solved values by box. An example of it is on the next page, at the bottom of the sudoku. It's main purpose is to assist in choosing digit patterns for subpattern analysis.

Using the proper tools

Since the erasers that pencils come with are quickly used up, I find it mandatory to keep on hand boxes of erasers, which can be found at stationery stores, as well as at chain stores like Office Depot. To ensure that an eraser stays on the pencil, it is sometimes effective to wind a small strip of Scotch tape around the pencil end so as to increase its diameter. If the eraser is too long, lop a small strip off the end of it off with a pair of scissors to ensure greater control.

It is also best to use No. 2 pencils or softer for annotation. Pencils which are too hard make for difficult erasures. Those too soft create smudges.

What to look for in the digit and its pattern

There are two requirements to look for in selecting a digit whose pattern is to be analyzed (in order to find the correct subpattern that leads to the solution). The first is that the pattern not be so extensive that it is difficult to analyze. The second is that a pattern be extensive enough to free up sufficient other digits to lead to a solution. Some sudokus, when they are reduced sufficiently to be analyzed successfully, have many digits which lead to solutions, but other sudokus, when they reach this state, have but one digit pattern which is both analyzable and yet extensive enough to free up sufficient other digits. Since there are nine digits altogether, and since the optimum seems to be a balance of the number of boxes in the digit pattern and the number of boxes not yet solved for the digit, it is best for the number of pattern boxes and the number of non-pattern boxes to be equal. This suggests a ratio of four to five or five to four. My experience is that five pattern boxes is close to ideal.

Example No. 1

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	57	457	8	6	25	9	1	24	3
row2	1	3	2	4	8	7	5	6	9
row3	9	45	6	35	1235	12	8	24	7
row4	358	589	7	2	59	4	6	1389	158
row5	35	2	1	58	569	68	39	7	4
row6	6	589	4	1	7	3	2	89	58
row7	4	178	5	78	16	168	39	39	2
row8	78	6	9	378	123	128	4	5	18
row9	2	18	3	9	4	5	7	18	6

1		**1**
1	**1**	

2		
2	**2**	**2**
2		**2**

3		**3**
	3	
3		

	4	
4	**4**	**4**
4	**4**	**4**

		5
5	**5**	**5**

6	**6**	**6**
6		**6**
6		**6**

	7	**7**
7	**7**	**7**
		7

8	**8**	**8**

9	**9**	**9**
9	**9**	

Just look at box F and the updates provided by the insertion of a 1,3 or 5 digit in any square and the ensuing updates in box I. The entire middle of the sudoku in boxes D, E & F is immediately solved. The effects of a 2, 4, 6 or 7 in most of the boxes look meager. I would choose the 1-pattern, just because 1 is first in the 5-box patterns. It is quite likely that several patterns would work, but the 1, 3 & 5 look to be the easiest.

6-box	5-box	4-box	3-box	2-box
8-pattern	1-,3-,5-pattern	9-pattern	2- ,7-pattern	4-, 6-pattern
too complicated	just right	too simple?	too simple	too simple

Example No. 2

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	3	4	1	9	57	2	8	57	6
row2	7	25	8	3	15	6	9	125	4
row3	6	259	259	157	4	8	17	1257	3
row4	2	3	49	6	19	7	14	8	5
row5	5	69	7	18	189	4	3	16	2
row6	8	1	46	2	3	5	47	67	9
row7	1	267	26	4	267	3	5	9	8
row8	9	258	3	58	258	1	6	4	7
row9	4	5678	56	578	5678	9	2	3	1

1		
1		
1	**1**	**1**

	2	
2	**2**	**2**
		2

3	**3**	**3**
3	**3**	**3**
3	**3**	**3**

4	**4**	**4**
	4	
4	**4**	**4**

5	**5**	**5**
		5

6	**6**	**6**
	6	
		6

7		
7	**7**	
		7

8	**8**	**8**
8		**8**

	9	**9**
		9
9	**9**	**9**

The 5-pattern doesn't seem as effective as the 7-pattern. I'd opt for the 7-pattern.

5-box	**4-box**	**3-box**	**2-box**
5-,7-pattern	1-,2-,6-,8-pattern	9-pattern	4-pattern
just right	too simple	too simple	too simple

Example No. 3

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	6	9	5	7	3	4	1	8	2
row2	17	2	17	569	589	58	4	3	69
row3	8	4	3	1	69	2	5	69	7
row4	2	138	167	56	157	9	378	4	3568
row5	1379	138	1679	2	4	1357	3789	579	35689
row6	379	5	4	8	67	37	2	679	1
row7	4	13	2	59	78	6	3789	1579	3589
row8	1359	6	19	4	2	78	3789	1579	3589
row9	59	7	8	3	159	15	6	2	4

	1	**1**
		1

2	**2**	**2**
2	**2**	**2**
2	**2**	**2**

3		**3**
	3	

4	**4**	**4**
4	**4**	**4**
4	**4**	**4**

5		**5**
5		

6		
6	**6**	**6**

	7	**7**

8		**8**
	8	
8		

9		
	9	

Here, the 6-pattern and the 8-pattern look like the best choices. I'd opt for the 6-pattern, since it seems to free up more of the other digits.

7-box	6-box	5-box
7-,9--pattern	1-,3-,5-pattern	6-,8-pattern
too complicated	too complicated	just right

Example No. 4.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	9	2	6	3	1	58	4	7	58
row2	3	7	48	46	25	258	19	56	1589
row3	1	48	5	46	7	9	3	26	28
row4	45	1	234	8	2349	6	7	25	2359
row5	58	9	238	7	23	1	6	4	235
row6	6	34	7	5	2349	234	19	8	1239
row7	7	345	1	2	345	345	8	9	6
row8	48	3458	348	9	6	345	2	1	7
row9	2	6	9	1	8	7	5	3	4

1	**1**	
1	**1**	
1	**1**	**1**

2		
2	**2**	**2**

3	**3**	**3**
		3

		4
		4
		4

5		
	5	
		5

6		
6	**6**	**6**
6	**6**	**6**

7	**7**	**7**
7	**7**	**7**
7	**7**	**7**

	8	**8**
	8	**8**

9	**9**	
9		
9	**9**	**9**

The 2-, 3-, and 8-patterns seem the best. I'd opt for the 8-pattern, in that it seems to free up more of the other digits.

6-box	**5-box**	**3-box**	**2-box**
4-,5-pattern	2-,3-,8-pattern	9-pattern	1-,6-pattern
too extensive	just right	too simple	too simple

Example No. 5.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	6	35	8	579	4579	4579	1	347	2
row2	1	4	7	3	6	2	9	58	58
row3	239	235	29	8	4579	1	46	3467	367
row4	4	28	1	256	3	56	7	9	568
row5	78	9	3	1567	14578	479	4568	2	568
row6	278	6	5	279	4789	479	3	48	1
row7	2789	1	269	4	579	3	568	5678	56789
row8	379	37	4	5679	2	8	56	1	35679
row9	5	378	69	1679	179	679	2	3678	4

1	**1**	**1**
1		**1**
1		**1**

	2	**2**
		2
	2	**2**

	3	
3	**3**	**3**
	3	

4		
4		
4	**4**	**4**

5		
5		

6	**6**	
6		

7		
		7

8	**8**	
	8	

		9
9		**9**

Here, the 2-, 3-, and 4–patterns are best. I'd opt for the 2-pattern, but all three look pretty good.

7-box	**6-box**	**4-box**	**2-box**
5-,7-pattern	6-,8-,9-pattern	2-,3-,4-pattern	1-pattern
too extensive	**too extensive**	best choice	too simple

Example No. 6.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	4	1	3578	578	357	6	789	2	389
row2	9	238	23578	578	2357	4	1678	378	1368
row3	378	238	6	1	9	237	4	5	38
row4	13	39	139	56	256	8	25	4	7
row5	78	5	4	9	1237	1237	128	6	1238
row6	2	6	78	4	1357	13	1589	389	1389
row7	16	4	129	67	8	5	3	79	269
row8	5	289	289	3	167	17	26789	789	4
row9	368	7	38	2	4	9	68	1	5

1	**1**	
		1

		2
2		
	2	

	3	**3**

4	**4**	**4**
4	**4**	**4**
4	**4**	**4**

		5
5		
5	**5**	**5**

6	**6**	
6		**6**

		7
7		

	8	
	8	

9	**9**	
	9	
	9	

The 6- and 9-box patterns look best. I'd opt for the 6-pattern.

7-box	**6-box**	**5-box**	**4-box**
3-,7-,8-pattern	1-,2-pattern	6-,9-pattern	5-pattern
too extensive	**too extensive**	best choice	too simple

Partial subpatterns

There are some situations in which all subpatterns but one fail, yet that last subpattern, although clearly the correct subpattern, does not completely solve the sudoku. This is a situation where another subpattern must be used to finish it.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	467	467	3	8	47	5	1	2	9
row2	47	2	8	1	347	9	37	6	5
row3	9	5	1	37	2	6	8	4	37
row4	27	79	4	29	6	1	5	3	8
row5	136	36	5	29	8	347	47	19	247
row6	13	8	29	5	347	347	6	19	247
row7	348	34	6	37	9	378	2	5	1
row8	2348	1	29	6	5	38	349	7	34
row9	5	39	7	4	1	2	39	8	6

1	1	1
	1	
1	1	1

2	2	2
	2	2

3		
		3

		4
4		
	4	

5	5	5
5	5	5
5	5	5

	6	6
	6	6
6	6	6

7		7

8	8	8
8	8	8
		8

9	9	9
	9	

The 2-pattern and the 9-pattern seem to be the best choices. We'll choose the 2-pattern.

7-box	**6-box**	**5-box**	**4-box**	**2-box**
3-,7-pattern	4-pattern	9-pattern	2-pattern	1-,6-,8-box
too complex	too complex	just about right	just about right	too simple

This is the 2-pattern

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

We'll choose the box order D→E→F→G. Subpattern-1 is quickly completed. Back at the starting box, we have no pending situation, so we are allowed to change the box order to D→F→E→G, and subpattern-2 is equally simple.

A	B	C
D	E	F
G	H	I

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1								**x**	
row2		**x**							
row3					**x**				
row4	2_1			2_2					
row5				2_1					2_2
row6			2_2						2_1
row7							**x**		
row8	2_2		2_1						
row9						**x**			

Subpattern-1 fails, proving that subpattern-2 is the correct subpattern, but subpattern-2, although correct, does not lead to the solution. We are stuck with the following situation. What do we do next? The answer - we try another subpattern. However, the sudoku has changed. Look at the box pattern below.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	467 **6**	467 **7**	3	8	47 **4**	5	1	2	9
row2	47 **4**	2	8	1	347 **37**	9	37	6	5
row3	9	5	1	37	2	6	8	4	37
row4	27 **7**	79 **9**	4	29 **2**	6	1	5	3	8
row5	136 **3**	36 **6**	5	29 **9**	8	347 **47**	47	19 **1**	247 **2**
row6	13 **1**	8	29 **2**	5	347 **37**	347	6	19 **9**	247 **47**
row7	348 **8**	34 **4**	6	37	9	378 **37**	2	5	1
row8	2348 **2**	1	29 **9**	6	5	38 **8**	349 34	7	34
row9	5	39 **3**	7	4	1	2	39 **9**	8	6

1	**1**	**1**
1	**1**	**1**
1	**1**	**1**

2	**2**	**2**
2	**2**	**2**
2	**2**	**2**

3		
3		**3**
3		

		4
4		
4	**4**	

5	**5**	**5**
5	**5**	**5**
5	**5**	**5**

6	**6**	**6**
6	**6**	**6**
6	**6**	**6**

7		
7		
7		**7**

8	**8**	**8**
8	**8**	**8**
8	**8**	**8**

9	**9**	**9**
9	**9**	**9**
9	**9**	**9**

The 3, 4 and 7 patterns are all equal with 5 boxes, so we'll arbitrarily choose the new 3-pattern.

Before proceeding with the next subpattern analysis, we finalize the updates, by removing all the candidates in solved squares, and replace the original candidate with the temporary update candidates, since the temporary update is the correct one. The new 3-pattern is shown at the bottom of the page.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	6	7	3	8	4	5	1	2	9
row2	4	2	8	1	37	9	37	6	5
row3	9	5	1	37	2	6	8	4	37
row4	7	9	4	2	6	1	5	3	8
row5	3	6	5	9	8	47	47	1	2
row6	1	8	2	5	37	347	6	9	47
row7	8	4	6	37	9	37	2	5	1
row8	2	1	9	6	5	8	34	7	34
row9	5	3	7	4	1	2	9	8	6

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

We'll choose the box order I→C→B→E→H, with I as the starting box. Subpattern-1 is quickly completed. Back at the starting box I, we have no pending situation, so we may change the box order to I→C→B→H→E, and subpattern-2 is equally simple.

	B	C
	E	
	H	I

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1			x						
row2					3_1		3_2		
row3				3_2					3_1
row4								x	
row5	x								
row6					3_2	3_1			
row7				3_1		3_2			
row8							3_1		3_2
row9		x							

The 3-subpattern-2 leads quickly to the solution:

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

Chapter 5. Pattern-directed updating.

Using two sudoku forms:

The use of subpatterns should not, as a general rule, be a method used entirely by itself. It is one of four "single-candidate" methods, all of which relate to the pattern of candidates. The first three, in the list below, are important is that, if they decrease the number of candidates, they enhance the effectiveness of solution by subpattern.

It is important, therefore, to preserve one's comprehension of this pattern of candidates as it undergoes reduction. As I have urged before, the only way to accomplish this is to use two sudoku forms, side by side, with the sudoku to be solved on the right, and a blank form on the left for analyzing each digit pattern in its turn. At the conclusion of studying a particular digit pattern, and before begiining an analysis of the next, this form is erased. The study of a digit pattern is much more convenient and effective when carried out in this manner, compared to the visual difficulties encountered in keeping track of all the occurrences of a candidate when it is surrounded by all the other candidates.

These single-candidate methods, in the order of their usage, are:

1. Row, column, and box claims.
2. N-wings.
3. Polarity chains.
4. Subpattern examination.

It is wise to examine every digit pattern for these four situations. Every n-wing found has the possibility of further reduction, and reduced patterns fare better as useful subpatterns.

Example No. 1.

The following sudoku has been through all the initial stages including annotation and some related updates.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	4	7	2	16	5	16	8	9	3
row2	35	59	6	239	4	8	57	1	27
row3	135	8	1359	239	7	23	6	25	4
row4	8	169	179	4	26	17	1379	23	5
row5	1567	3	157	8	26	9	17	4	27
row6	2	149	1479	17	3	5	179	8	6
row7	9	45	3457	2357	8	2347	35	6	1
row8	13567	2	1357	3567	9	367	4	35	8
row9	356	456	8	356	1	346	2	7	9

		1
	1	**1**

2		
2		
2		**2**

		3
3	**3**	

4	**4**	**4**
	4	**4**
		4

	5	
	5	**5**

6		**6**
		6
		6

7	**7**	
		7

8	**8**	**8**
8	**8**	**8**
8	**8**	**8**

		9
	9	
9	**9**	**9**

Even though the 1-pattern doesn't seem as if it is the best choice, we shall nonetheless begin with it. See the next page for the analysis.

6-box	**5-box**	**4-box**	**3-box**
1-,3-,5-,7-pattern	2-,6-pattern	9-pattern	4-pattern
too complicated?	just about right	might work	too simple

Example No. 1, continued.

This is the 1-pattern. Notice the two 1's in squares (42) and (62). They are the ***only*** 1-candidates in column 2. Therefore they own column 2. Therefore they own box D. Therefore we can eliminate all the other 1-candidates in box D.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	X			1		1			
row2								X	
row3	1		1						
row4		1	~~1~~			1	~~1~~		
row5	~~1~~		~~1~~				X		
row6		1	~~1~~	1			~~1~~		
row7									
row8	1		1						
row9					X				

Example No. 1, continued.

Next, eliminating the two 1-candidates in row 5 makes the 1-candidate in square (57) unique. It therefore becomes the solved value for its square, and we replace it by an X. It immediately eliminates the 1-candidates from squares (47) and (67). These changes simplify the 1-pattern. See below.After making the eliminations, the picture of 1-candidates looks like this, and it is clear that there are two separate 1-patterns: The first one, shown in white squares, is in boxes A and G. The second one, in dark squares, is in boxes B, D and E. The reason these two patterns are separate is that their members cannot affect one another. (Although both patterns have 2 candidates per box, the second one gets a third more coverage, allowing more freeing of digits through interaction with other squares. We shall, nevertheless, go with the first pattern.)

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1				**1**		**1**			
row2								X	
row3	1_1		1_2						
row4		**1**				**1**			
row5							X		
row6		**1**		**1**					
row7									X
row8	1_2		1_1						
row9					X				

We must next test these two subpatterns of the first 1-pattern, to discover which is the correct one. Trying out subpattern-1 of the first pattern, we insert solved 1's in squares (31) and (83), But the temporary update doesn't

Example No. 1, continued.

seem to lead anywhere, so we erase the temporary updates and try out subpattern-2, with 1's in squares (33) & (81).

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	**4**	**7**	**2**	16	**5**	16	**8**	**9**	**3**
row2	35	59	**6**	239	**4**	**8**	57	**1**	27
row3	135	**8**	**1**	239	**7**	23	**6**	25	**4**
row4	**8**	169	179	**4**	26	17	1379	23	**5**
row5	1567	**3**	157	**8**	26	**9**	17	**4**	27
row6	**2**	149	1479	17	**3**	**5**	179	**8**	**6**
row7	**9**	45	3457	2357	**8**	2347	35	**6**	**1**
row8	**1**	**2**	1357	3567	**9**	367	**4**	35	**8**
row9	356	456	**8**	356	**1**	346	**2**	**7**	**9**

The 1-subpattern-2 solves the sudoku:

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

Example No. 1, continued.

Let us now test the 2-pattern:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1			X						
row2				2_1					2_{23}
row3				2_2		2_3		2_1	
row4					2_1			2_{23}	
row5					2_{23}				2_1
row6	X								
row7				2_3		2_{12}			
row8		X							
row9							X		

The 2-subpattern-1 solves the sudoku:

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

The 6-pattern also works, but is more complex.

Example No. 2.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	16	16	5	4	2378	27	9	78	2378
row2	28	7	9	6	2358	1	235	4	2358
row3	3	28	4	28	579	579	6	57	1
row4	1258	9	28	7	12	3	125	6	4
row5	125678	12368	23678	12	4	69	1257	15789	25789
row6	12678	4	2678	5	69	8	127	3	279
row7	9	238	2378	1238	123578	257	4	157	6
row8	678	5	3678	9	13678	4	137	2	37
row9	4	236	1	23	23567	2567	8	579	3579

This is the 1-pattern. Notice the 2-wing in squares (54), (58), (74) & (78). This eliminates the 1's in squares (51), (52), (57) & (75).

	1	2	3	4	5	6	7	8	9
1	**1**	**1**							
2						**x**			
3									**x**
4	**1**				**1**		**1**		
5	~~**1**~~	~~**1**~~		**1**			~~**1**~~	**1**	
6	**1**						**1**		
7				**1**	~~**1**~~			**1**	
8					**1**		**1**		
9			**x**						

Example No. 2, continued.

The elimination of the 1-candidates creates a unique candidate in square (12), which eliminates the 1-candidate in square (11), promoting the 1-candidate to a solved value, which then gets replaced with an X. The pattern analysis is shown at the bottom of the page.

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

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1		X							
row2						X			
row3									
row4	1_1				1_2		1_3		
row5				1_{13}				1_2	
row6	1_{23}						1_1		
row7				1_2				1_{13}	
row8					1_{13}		1_2		
row9	1_1	1_{12}							

Example No. 2, continued.

Subpattern-1 solves the sudoku:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	16 **6**	16 **1**	**5**	4	2378 **3**	27 **7**	9	78 **8**	2378 **2**
row2	28 **2**	7	9	6	2358 **8**	1	235 **3**	4	2358 **5**
row3	3	28 **8**	4	28 **2**	579 **9**	579 **5**	6	57 **7**	1
row4	1258 **1**	9	28 **8**	7	12 **2**	3	125 **5**	6	4
row5	125678 **5**	12368 **6**	23678 **3**	12 **1**	4	69 **9**	1257 **2**	15789 **9**	25789 **8**
row6	12678 **7**	4	2678 **2**	5	69 **6**	8	127 **1**	3	279 **7**
row7	9	238 **3**	2378 **7**	1238 **8**	123578 **5**	257 **2**	4	157 **1**	6
row8	678 **8**	5	3678 **6**	9	13678 **1**	4	137 **7**	2	37 **3**
row9	4	236 **2**	1	23 **3**	23567 **7**	2567 **6**	8	579 **5**	3579 **9**

Example No. 3.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	9	134	146	3678	5	368	368	17	2
row2	67	8	2	1	37	4	5	9	36
row3	167	28	5	23678	9	2368	4	17	368
row4	138	6	189	2379	137	5	289	4	789
row5	2	49	7	469	8	69	1	3	5
row6	1348	5	189	379	1347	12369	289	6	789
row7	1468	19	3	89	2	189	7	5	469
row8	146	2	169	5	134	7	369	8	3469
row9	5	7	489	3489	6	389	39	2	1

	1	
		1
		1

2		2
2		
2	2	2

		3
3		

	4	4
		4

5	5	5
5	5	5
5	5	5

6		6
	6	

7		
	7	

8		
	8	
		8

9	9	9

Patterns 1, 2 & 3 have no row, column or box claims, nor any n-wings, so their subpatterns are too extensive to lead to an easy solution. The 4-pattern, however, has a 4-wing in squares (12), (13), (52), (54), (93) and (94),

7-box	**6-box**	**3-box**
3-, 7- pattern	1-,4-,6-,8-,9-pattern	2-pattern
too complex	???	too simple

The 3-wing in squares (12), (13), (52), (54), (93) and (94) (shaded squares), eliminates the 4-candidates from squares (64), (72) & (74)
(vertical stripes):

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1		4	4	•					
row2						X			
row3							X		
row4								X	
row5		4	•	4					
row6	4		4	4	4				
row7	4	4		4					4
row8	4		4		4				4
row9		•	4	4		4			

Note that basis rule about n-wings, that they may have as little as 2 candidates per row or column, yet still be accepted as legitimate.
The squares with a dot at their centers are the ones not possessing a 2, yet still accepted as part of the n-wing.

Example No. 3, continued.

We next investigate the polarity of the candidates:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1		4+	4-						
row2									
row3									
row4									
row5		4-		4+					
row6	4		~~4~~		4-				
row7	4								4
row8	4		~~4~~		4+				4
row9			4+	4-					

The 4's in squares (83) & (63) can "see" both the 4+ in square (93) and the 4- in square (13), so they may both be eliminated.

Example No. 3, continued.

Analyzing the resultant 4-pattern, we have the following results:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1		4_1	4_{23}						
row2									
row3									
row4									
row5		4_{23}		4_1					
row6	4_1				4_{23}				
row7	4_2								4_{13}
row8	4_3				4_1				4_2
row9			4_1	4_{23}					

The 4-subpattern-2 solves the sudoku.

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

Example No. 4

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	57	457	8	6	25	9	1	24	3
row2	1	3	2	4	8	7	5	6	9
row3	9	45	6	35	1235	12	8	24	7
row4	358	589	7	2	59	4	6	1389	158
row5	35	2	1	58	569	68	39	7	4
row6	6	589	4	1	7	3	2	89	58
row7	4	178	5	78	16	168	39	39	2
row8	78	6	9	378	123	128	4	5	18
row9	2	18	3	9	4	5	7	18	6

1		1
1	1	

2		
2	2	2
2		2

3		3
	3	
3		

	4	
4	4	4
4	4	4

		5
5	5	5

6	6	6
6		6
6		6

	7	7
7	7	7
		7

8	8	8

9	9	9
9	9	

Just look at box F and the updates provided by the insertion of a 1,3 or 5 digit in any square and the ensuing updates in box I. The entire middle of the sudoku in boxes D, E & F is immediately solved. The effects of a 2, 4, 6 or 7 in most of the boxes look meager. I would choose the 1-pattern, just because 1 is first in the 5-box patterns. It is quite likely that several patterns would work, but the 1, 3 & 5 look to be the easiest.

6-box	**5-box**	**4-box**	**3-box**	**2-box**
8-pattern	1-,3-,5-pattern	9-pattern	2- ,7-pattern	4-, 6-pattern
too complicated	just right	too simple?	too simple	too simple

Example No. 4, continued.

This is the 1-pattern:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1							X		
row2	X								
row3					1_{24}	1_{13}			
row4								1_{12}	1_{34}
row5			X						
row6				X					
row7		1_{34}			1_{1}	1_{2}			
row8					1_{3}	1_{4}			1_{12}
row9		1_{12}						1_{34}	

The 1-subpattern-3 solves the sudoku.

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

Example No. 5

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	3	4	1	9	57	2	8	57	6
row2	7	25	8	3	15	6	9	125	4
row3	6	259	259	157	4	8	17	1257	3
row4	2	3	49	6	19	7	14	8	5
row5	5	69	7	18	189	4	3	16	2
row6	8	1	46	2	3	5	47	67	9
row7	1	267	26	4	267	3	5	9	8
row8	9	258	3	58	258	1	6	4	7
row9	4	5678	56	578	5678	9	2	3	1

1		
1		
1	**1**	**1**

	2	
2	**2**	**2**
		2

3	**3**	**3**
3	**3**	**3**
3	**3**	**3**

4	**4**	**4**
	4	
4	**4**	**4**

5	**5**	**5**
		5

6	**6**	**6**
	6	
		6

7		
7	**7**	
		7

8	**8**	**8**
8		**8**

	9	**9**
		9
9	**9**	**9**

The 5-pattern doesn't seem as effective as the 7-pattern. I'd opt for the 7-pattern. Nevertheless, I'll start out with the 1-pattern. There is no n-wing, but the patttern looks promising, so I'll check out the 1-subpatterns.

5-box	**4-box**	**3-box**	**2-box**
5-,7-pattern	1-2-,6-,8-pattern	9-pattern	4-pattern
just right	too simple	too simple	too simple

This is the analysis of the 1-pattern:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1			X						
row2					1_1			1_2	
row3				1_2			1_K	1_1	
row4					1_K		1_{12}		
row5				1_1	1_2			1_K	
row6		X							
row7	X								
row8						X			
row9									X

The killers makes the 1-candidate in square (47) unique, thus promoting it to a solved value. The rest of the update stems from this one square, leading to the solution, shown below.

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

Example No. 5, continued.

Just for fun, we'll try out the 7-pattern.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1								7_{14}	
row2	X								
row3				7_{14}			7_{2}	7_{3}	
row4						X			
row5			X						
row6							7_{134}	7_{2}	
row7		7_{234}			7_{1}				
row8									X
row9		7_{1}		7_{23}	7_{4}				

When we try these subpatterns, 7-subpattern-1 fails, and 7-subpattern-2 is the solution, which is, of course, the same solution as on the preceding page.

Example No. 6

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	6	9	5	7	3	4	1	8	2
row2	17	2	17	569	589	58	4	3	69
row3	8	4	3	1	69	2	5	69	7
row4	2	138	167	56	157	9	378	4	3568
row5	1379	138	1679	2	4	1357	3789	579	35689
row6	379	5	4	8	67	37	2	679	1
row7	4	13	2	59	78	6	3789	1579	3589
row8	1359	6	19	4	2	78	3789	1579	3589
row9	59	7	8	3	159	15	6	2	4

	1	**1**
		1

2	**2**	**2**
2	**2**	**2**
2	**2**	**2**

3		**3**
	3	

4	**4**	**4**
4	**4**	**4**
4	**4**	**4**

5		**5**
5		

6		
6	**6**	**6**

	7	**7**

8		**8**
	8	
8		

9		
	9	

Here, the 6-pattern and the 8-pattern look like the best choices. I'd opt for the 6-pattern, since it seems to free up more of the other digits.

7-box	**6-box**	**5-box**
7-,9-pattern	1-,3-,5-pattern	6-,8-pattern
far too complicated	too complicated	just right

Example No. 6, continued.

The 1-, 3-& 5-patterns are too complicated (the 5-pattern has ten subpatterns), so we'll try the 6-pattern.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	X								
row2				6_{12}					6_{3}
row3					6_{3}			6_{12}	
row4			6_{2}	6_{3}					6_{1}
row5			6_{13}						6_{2}
row6					6_{12}			6_{3}	
row7						X			
row8		X							
row9							X		

6-subpattern-1 fails, 6-subpattern-2 jams up, and 6-subpattern-3 succeeds:

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

Example No. 6, continued

Just as well we aren't dependent on the 8-pattern for a solution. I wouldn't be too keen on having to test out this set of subpatterns. I could have guessed, looking at the two boxes, F & I. with 4 patttern members each, generating multiple pending conditions:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1								X	
row2					8_{1357}	8_{2468}			
row3	X								
row4		8_{5678}					8_{12}		8_{34}
row5		8_{1234}					8_{56}		8_{78}
row6				X					
row7					8_{2468}		8_{37}		8_{15}
row8						8_{1357}	8_{48}		8_{26}
row9			X						

(Note that the analysis of this subpattern can be simplified by using just boxes F & I alone, since box D is a derivative of box F, H a derivative of box I, and B a derivative of box H. See the section on derivative boxes on page 57.)

Example No. 7.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	9	2	6	3	1	58	4	7	58
row2	3	7	48	46	25	258	19	56	1589
row3	1	48	5	46	7	9	3	26	28
row4	45	1	234	8	2349	6	7	25	2359
row5	58	9	238	7	23	1	6	4	235
row6	6	34	7	5	2349	234	19	8	1239
row7	7	345	1	2	345	345	8	9	6
row8	48	3458	348	9	6	345	2	1	7
row9	2	6	9	1	8	7	5	3	4

1	**1**	
1	**1**	
1	**1**	**1**

2		
2	**2**	**2**

3	**3**	**3**
		3

		4
		4
		4

5		
	5	
		5

6		
6	**6**	**6**
6	**6**	**6**

7	**7**	**7**
7	**7**	**7**
7	**7**	**7**

	8	**8**
	8	**8**

9	**9**	
9		
9	**9**	**9**

The 2-, 3-, and 8-patterns seem the best. I'd opt for the 8-pattern, in that it seems to free up more of the other digits.

6-box	**5-box**	**3-box**	**2-box**
4-,5-pattern	2-,-,8-pattern	9-pattern	1-,6-pattern
too extensive	just right	too simple	too simple

Example No. 7, continued.

The 2-pattern is too extensive, with up to 8 subpatterns:

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

The 3-pattern is little better, promising at least 8 subpatterns.

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

continued on next page

Example No. 7, continued.

The 8-pattern is a slightly better, with only 5 subpatterns.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1						8_{345}			8_{12}
row2			8_{5}			8_{12}			8_{34}
row3		8_{1234}							8_{5}
row4				X					
row5	8_{245}		8_{13}						
row6								X	
row7							X		
row8	8_{13}	8_{5}	8_{24}						
row9					X				

The 8-subpattern-1 solves the sudoku:

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

Problem No. 8.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	6	35	8	579	4579	4579	1	347	2
row2	1	4	7	3	6	2	9	58	58
row3	239	235	29	8	4579	1	46	3467	367
row4	4	28	1	256	3	56	7	9	568
row5	78	9	3	1567	14578	47	4568	2	568
row6	278	6	5	279	4789	479	3	48	1
row7	2789	1	269	4	579	3	568	5678	56789
row8	379	37	4	5679	2	8	56	1	35679
row9	5	378	69	1679	179	679	2	3678	4

1	1	1
1		1
1		1

	2	2
		2
	2	2

	3	
3	3	3
	3	

4		
4		
4	4	4

5		
5		

6	6	
6		

7		
		7

8	8	
	8	

		9
9		9

Here, the 2-, 3-, and 4–patterns are best. I'd opt for the 2-pattern, although all three look pretty good.

7-box	**6-box**	**4-box**	**2-box**
5-,7-pattern	6-8-9-pattern	2-,3-,4-pattern	1-pattern
too extensive	too extensive	best choice	too simple

continued on the next page

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1									X
row2						X			
row3	2_1	2_2	2_3						
row4		2_{13}		2_2					
row5								X	
row6	2_2			2_{13}					
row7	2_3		2_{12}						
row8					X				
row9							X		

The 2-subpattern-3 solves the sudoku:

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

Problem No. 9.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	4	1	3578	578	357	6	789	2	389
row2	9	238	23578	578	2357	4	1678	378	1368
row3	378	238	6	1	9	237	4	5	38
row4	13	39	139	56	256	8	25	4	7
row5	78	5	4	9	1237	1237	128	6	1238
row6	2	6	78	4	1357	13	1589	389	1389
row7	16	4	129	67	8	5	3	79	269
row8	5	289	289	3	167	17	26789	789	4
row9	368	7	38	2	4	9	68	1	5

1	**1**	
		1

		2
2		
	2	

	3	**3**

4	**4**	**4**
4	**4**	**4**
4	**4**	**4**

		5
5		
5	**5**	**5**

6	**6**	
6		**6**

		7
7		

	8	
	8	

9	**9**	
	9	
	9	

The 6- and 9-patterns look best. I'd opt for the 6-pattern.

7-box	6-box	5-box	4-box
3-,7-,8-pattern	1-,2-pattern	6-,9-pattern	5-pattern
too extensive	**too extensive**	best choice	too simple

continued on the next page

Problem No. 9, continued.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1						X			
row2							6_{1}		6_{23}
row3			X						
row4				6_{13}	6_{2}				
row5								X	
row6		X							
row7	6_{3}			6_{2}					6_{1}
row8					6_{13}		6_{2}		
row9	6_{12}						6_{3}		

The 6-subpattern-3 solves the sudoku.

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

Chapter 6. Other directions.

Killer Candidates:

Killer candidates were first noticed for their properties in the study of patterns. They occurred as candidates not included in any subpattern. On examination, they all proved to have an interesting property – if promoted to established values, they created contradictions in their sudokus – rows, columns, or boxes lacking the associated candidate, or logical contradictions in which two would-be established values for that candidate vie for supremacy in the same row, column, or block.

Subsequent tests were made on the candidates eliminated as a result of n-wings and polarity structures, and all of those tested turned out also to be killer candidates. These anecdotal comments, however, do not establish the general statement that all such candidates are killers (although I believe that that to be the case).

The test to determine if a candidate is a killer candidate is a simple one, and is shown on page 115.

At the conclusion of this process, if there are any rows, columns, or blocks without any remaining candidates, or if two would-be established values end up in the same row, column, or block, then the original candidate being tested is a killer candidate, and should be eliminated.

If this test fails, every row, column, and block contains exactly one established value for the candidate in question. (These include the original established values which were not part of the candidate pattern.) This subpattern of established values is also the solution to the subpattern analysis, and the test itself is in reality the first step in the analysis of subpatterns using the "neighborhood elimination method."

Below is the 2-pattern for a sudoku in the process of being solved. The reason this subpattern was chosen was that, despite there being only 3 solved squares for the 2-candidates, and therefore as many as 6 blocks in the 2-pattern, the pattern itself was, except for two blocks, a simple bipattern, and in those two blocks consisted of only 3 candidates each:

Below that are the results of the 2-subpattern analysis. Note that squares (81) & (89) are part of no subpattern. Each was tested and found to be a killer candidate. The proofs that they are follow the 2-subpattern analysis.

The 2-pattern

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

Results of analysis of 2-pattern, showing the four subpatterns:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1									
row2						$2_{\mathbf{12}}$			$2_{\mathbf{34}}$
row3				$2_{\mathbf{34}}$				$2_{\mathbf{12}}$	
row4									
row5	$2_{\mathbf{24}}$		$2_{\mathbf{13}}$						
row6									
row7	$2_{\mathbf{13}}$		$2_{\mathbf{24}}$						
row8	$2_{\mathbf{K}}$			$2_{\mathbf{12}}$				$2_{\mathbf{34}}$	$2_{\mathbf{K}}$
row9		$2_{\mathbf{12}}$				$2_{\mathbf{34}}$			$2_{\mathbf{12}}$

Proof that the 2-candidate in square (81) is a killer:

Original 2-pattern with underlined killer in square (81); Established 2's in boxes [11], [22] & [23] not shown.

	1	2	3	4	5	6	7	8	9
1									
2						2			2
3				2				2	
4									
5	2		2						
6									
7	2		2						
8	$\underline{\mathbf{2}}$			2				2	2
9						2			2

Neighborhood of (81) in large bold:

	1	2	3	4	5	6	7	8	9
1									
2						2			2
3				2				2	
4									
5	**2**		2						
6									
7	**2**		**2**						
8	$\underline{\mathbf{2}}$			**2**				**2**	**2**
9						2			2

After elimination of all candidates in neighborhood of (91), establishing 2's in squares (96) & (99).

	1	2	3	4	5	6	7	8	9
1									
2						2			2
3				2				2	
4									
5			2						
6									
7									
8	$\underline{\mathbf{2}}$								
9						**2**			**2**

After elimination of 2-candidates in neighborhoods of established 2's in squares (96) & (99)

	1	2	3	4	5	6	7	8	9
1									
2									
3				**2**				**2**	
4									
5			2						
6									
7									
8	$\underline{\mathbf{2}}$								
9						**2**			**2**

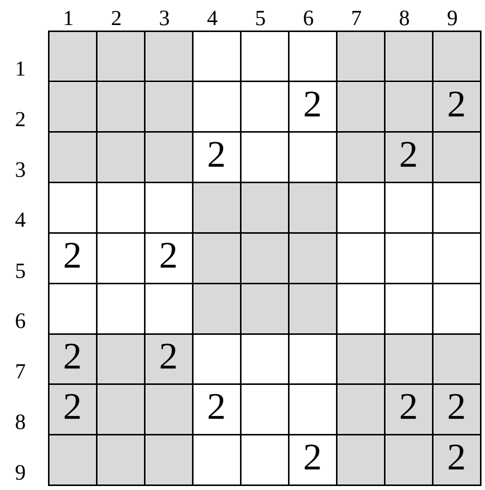
Now we have established values duking it out in both rows 3 & 9, and if either is arbitrarily selected, the elimination of the other leads to a box with no 2-candidate.. Clearly an impossible situation. The assumption of the 2-candidate in square (91) has led to contradictions, proving that it is a killer candidate.

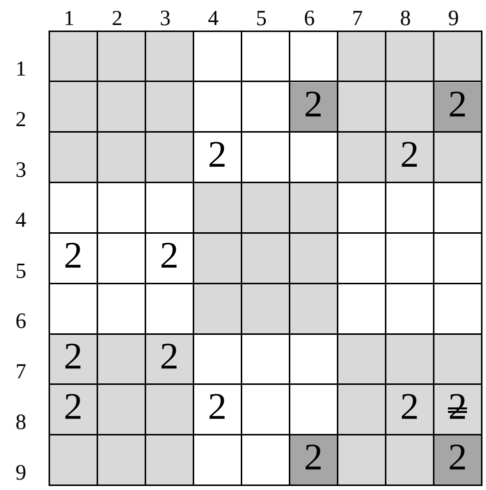
A similar proof shows that the 2-candidate in square (89) is likewise a killer candidate.

It is easy to argue that both squares (81) & (89) would also be killers if either were eliminated first. If one eliminated the other, it would play no part. If both persisted in a partial subpattern, so that both exercised their killing talents, the results would still be contradictory in the sense of incompleteness. It is always the assumption in solving sudokus that there is one and only one solution, so having no solution is a contradiction.

The use of 2-wings to eliminate the same two killer candidates:

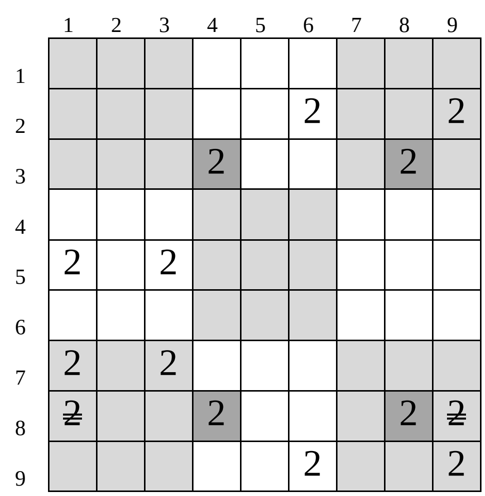
Looking again at the 2-pattern, we notice that the 2-candidates in squares (26), (29), (96), and (99) form a horizontal 2-wing, rows 2 & 9 having 2's only in columns 6 & 9. By the structure theorem, columns 6 & 9 may have 2's only in rows 2 & 9. Therefore the 2-candidate in square (89) must be eliminated.

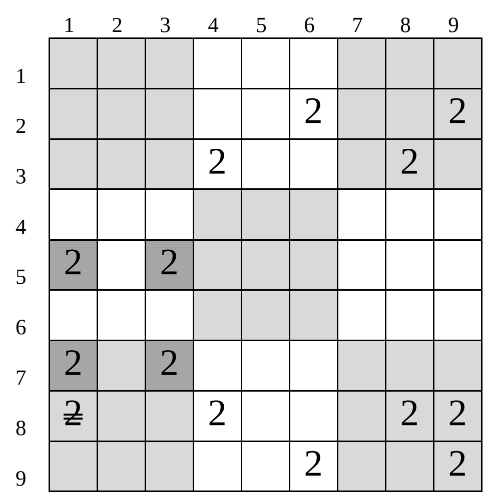




There is a second 2-wing in 2's in squares (34), (38), (84), & (88). It is a vertical 2-wing composed of columns 4 & 8, with 2's only in rows 3 & 8. By the structure theorem, rows 3 & 8 may only have 2's in columns 4 & 8. Therefore the 2's in (81) & (89) must be eliminated:

And a third 2-wing can be found in squares (51), (53), (71) & (73). It is a horizontal 2-wing in rows 5 & 7 with 2's in columns 1 & 3. By the structure theorem, columns 1 & 3 may only have 2's in rows 5 & 7. The 2 in (81) must therefore be eliminated:





Thus there are 3 2-wings in 2-candidates, the first 2-wing eliminating the 2-candidate in square (89), the second 2-wing eliminating the 2-candidates in both squares (81) & (89), and the third 2-wing eliminating only the 2-candidate in square (81).

The use of polarity to eliminate the same two killer candidates:

Chain: (88+)
↕
↔↔ (38-)↔↔
↕ ↕
(34+)↔ (38-)↔ (38-)
↕ ↕ ↕
(84-)↔ (88+)↔ (38-) (shown in the shaded squares below)

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1									
row2						2-			2+
row3				2+				2-	
row4									
row5	2		2						
row6									
row7	2		2						
row8	~~2~~			2-				2+	~~2~~
row9						2+			2-

Square (81) can "see" both (84-) & (88+), so its 2-candidate must be eliminated. The same is true of square (89). Square (89) can also "see" both (29+) & (99-), giving us a second reason to eliminate square (89).

A second, lesser polarity chain exists which eliminates only one of the two killer candidates:

Chain: (51+)↔(53-)↔(73+)↔ (71-) (shown in the shaded squares below)

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1									
row2						2			2
row3				2				2	
row4									
row5	2+		2-						
row6									
row7	2-		2+						
row8	~~2~~			2				2	2
row9						2			2

Square (81) can "see" both (51+) & (71-), so its 2-candidate must be eliminated.

Conclusions:

This example is unusual in that three different methods may be used to eliminate the two 2-candidates in squares (81) & (89): the discovery of the two killer candidates using **subpattern analysis**, the elimination of the same two candidates using **n-wings**, and the elimination using **polarity chains.**

These three methods seldom overlap like this; the simplicity of the pattern is one reason. Only 2-wings may be solved using the polarity principle, and only bipatterns lend themselves to the use of polarity. Polarity can only be used when sufficient paired squares are available, and cannot be used when the number of candidates in all the rows, columns and boxes is three or more. Subpattern analysis may always be used, although it is not practical when the number of subpatterns is too great. Quite often, n-wings eliminate the number of candidates to make subpattern analysis practical.

Chapter 7. Box-line deletions.

Most readers are familiar with the concept of what I call "Box-line" deletions, (where line = row or column). Paul Stephens (see bottom of next page) calls them "Box Claims," "Row Claims," and "Column Claims." The only reason this chapter exists is to explain the term "box-line deletions," so as to be able to include it as an additional single-candidate method.

In the first example, the 4-candidates in squares (31) & (32) are the only ones in row 3, so if either of the 4-candidates in squares (13) or (21) were promoted, they would cause the 4-candidates in squares (31) and (32) to be deleted. But then, row 3 would have no 4-candidate. Therefore the 4-candidates in squares (31) & (32) not only "own" row 3, they also "own" box A, and the 4-candidates in squares (13) & (21) must be removed.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row1	3	7	2458						
row2	468	9	28						
row3	46	45	1	29	269	7	3	25	8

In the second example, the 4-candidate in square (78) is the only 4-candidate in box I. If the 4-candidate in either square (71) or (72) is promoted, the 4-candidate in square (78) would be deleted, leaving box I bereft of 4-candidates. Therefore the 4-candidates in both squares (71) & (72) must be deleted.

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8	col 9
row7	457	245	8	6	257	9	1	24	3
row8							257	6	9
row9							8	257	57

In the third example (on the left, below), if the 4-candiate in either square (51) or (61) were promoted, the 4-candidates in squares (81) & (91) would have to be removed, resulting in no 4-candidate in box G. Therefore the 4-candidates in both squares (51) & (61) must be removed.

In the fourth example (on the right, below), if the 4-candidate in square (27) were promoted, the 4-candidate in square (19) would have to be removed. But this would result in no 4-candidate in column 9. Therefore the 4-candidate in square (27) must be removed.

	col 1	col 2	col 3
row1	8		
row2	256		
row3	3		
row4	7		
row5	249		
row6	49		
row7	1	28	258
row8	45	9	7
row9	46	68	3

	col 7	col 8	col 9
row1	9	25	149
row2	14	26	7
row3	3	256	8
row4			29
row5			6
row6			12
row7			123
row8			139
row9			5

Postscripts: I have no reason to recommend other textbooks on subpattern analysis, as nobody, to my knowledge, has yet written anything closely related to it (although simple subpatterns result from guessing their first members, a cleaning-up procedure promulagated by several of the masters when ordinary methods fail to completely resolve a sudoku).

However I do take the liberty at this point of mentioning Paul Stephens' excellent textbook "Mastering Sudoku week by week (2007), published by Duncan Baird Publishers, London, as Stephens has offered the nomenclature of "box claims," "row claims," and "column claims" for the three aspects of box-line deletions.