Metodi per risolvere i **Sudoku**

Introduzione

In Italia il **Sudoku** (che può essere tradotto come “numeri unici”) è stato senz’altro il gioco rivelazione dell’estate 2005. Sotto gli ombrelloni era facilissimo trovare qualche amico o qualche parente alle prese con la diabolica griglia 9x9, tutti armati di matita e gomma, o penna, e tanta pazienza. Non sono mancati di certo i giornali, sia a tiratura nazionale sia locale, che periodicamente foraggiavano i villeggianti proponendo un vasto campionario di Sudoku con differenti livelli di difficoltà.
Queste pagine sono il risultato della curiosità personale nel capire quali arcani logico-matematici ci fossero dietro il diabolico puzzle e come mettere insieme alcune tecniche risolutive realizzabili anche senza l’ausilio di un computer. Ovviamente nulla vieta di implementare le regole descritte nel seguito in un programma per computer ed infatti, durante la breve ricerca bibliografica fatta per raccogliere del materiale sull’argomento, mi è capitato spesso di trovare siti che propongono software per la soluzione di Sudoku che implementano quasi tutte le tecniche descritte. Buona parte di questi programmi sono inclusi nei siti riportati in bibliografia o si possono recuperare attraverso i link indicati in questi.

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

$\Rightarrow$

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

Un esempio di problema Sudoku e la sua soluzione.

Prima di iniziare premetto una breve digressione di carattere storico. Il Sudoku mantiene una lontana parentela con i più antichi, e più semplici, quadrati latini (molte varianti di questo tipo di quadrati vengono genericamente chiamati *quadrati magici*). Un **quadrato latino** di ordine n è una griglia quadrata di nxn celle nella quale compaiono n **simboli** diversi e che soddisfa le seguenti condizioni:
1) in ogni cella della griglia compare un simbolo;
2) ciascun simbolo deve comparire in ogni riga e in ogni colonna.

	1	2	3
1	1	2	3
2	3	1	2
3	2	3	1

Esempio di quadrato latino che utilizza i simboli dell’insieme S = {1, 2, 3}.

Un **quadrato greco-latino** è ottenibile come sovrapposizione di due particolari tipi di quadrati latini, ciascuno dei quali utilizza un diverso insieme di simboli, ed è tale che:

1) ciascuna coppia di simboli, ottenuta dall’unione dei due simboli con la stessa cella nei quadrati latini, compare una sola volta nel quadrato greco-latino.

Per esempio, se consideriamo due quadrati latini che utilizzano rispettivamente l’insieme di simboli S1= {1, 2, 3} ed S2 = {a, b, c}, il quadrato greco-latino sarà riempito con le coppie di simboli: a1, a2, a3, b1, b2, b3, c1, c2, c3. All’interno di una cella, l’ordine con cui compaiono i due simboli è ininfluente (1a = a1).

	1	2	3	4	5	6	7	8	9
1	1	2	3						
2	3	1	2	3					
3	2	3	1	2	3				
4				a	b	c			
5			a	b	c	a			
6		a	b	c	a	b			
7							1a	2b	3c
8							3b	1c	2a
9							2c	3a	1b

Come ottenere, su una tavola Sudoku, due quadrati latini di ordine 3 ed il relativo quadrato greco-latino.

Uno tra i più grandi matematici della storia, **L. Eulero**, nel XVIII secolo studiò a fondo questo tipo di quadrati magici e, tra l'altro, propose anche un nuovo tipo di quadrato magico di 9x9 celle, da riempire in modo che ciascun simbolo fosse presente una sola volta su ciascuna riga e ciascuna colonna (rispetto al Sudoku manca il vincolo sui blocchi) e che alcuni ritengono sia il progenitore dell'attuale Sudoku. Per onor di cronaca, bisogna anche dire che altri studiosi fanno risalire il luogo di nascita del Sudoku in zone molto lontane dalla nostra Europa, in terre orientali e in epoche antecedenti il XVIII secolo.
Andando ancora più indietro nel tempo, troviamo il seguente graffito di epoca Romana (I sec. d.c.) presenta una certa somiglianza con i quadrati latini: è un quadrato ed è scritto in latino, ma non è un quadrato latino.

A =

S	A	T	O	R
A	R	E	P	O
T	E	N	E	T
O	P	E	R	A
R	O	T	A	S

La frase latina, che può essere letta dall'alto verso il basso e viceversa, da sinistra a destra e viceversa è la seguente: "SATOR AREPO TENET OPERA ROTAS", che potrebbe essere tradotta ne "Il seminatore, col suo aratro, tiene con cura le ruote". Si può verificare facilmente come anche le celle di questo quadrato godono di una forma di simmetria rispetto al centro. In particolare, essendo A una matrice quadrata di ordine n=5, abbiamo che $a(i,j) = a(n+1-i, n+1-j)$.

Di una proprietà simile gode anche **Lo Shu**, forse **il primo quadrato magico della storia**, ritrovato in Cina ed inciso sul dorso di una carcassa di tartaruga. Oltre alla particolare simmetria rispetto al centro ($a(i,j) = n+1- a(n+1-i, n+1-j)$), questo quadrato 3x3 gode anche della seguente proprietà: ogni numero compreso tra 1 e 9 compare una sola volta in tutta la matrice e la somma delle celle in qualunque riga o diagonale è pari a 15.

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

Lo Shu cinese, di circa tremila anni.

Un buon riferimento, in italiano, per approfondire questi argomenti è fornito dal sito **Polymath** del Politecnico di Torino [4], che oltre ad una panoramica sull'argomento e molti riferimenti bibliografici, contiene anche un'intervista a **Wayne Gould**, che ha importato i Sudoku in europa curandone la pubblicazione sulla rivista Times. Anche il sito di **Wikipedia** [5] propone una pagina dedicata al Sudoku con molti riferimenti utili, così come la versione on-line del quotidiano "la Repubblica" [3].

Un po' di teoria …

In questa sezione vengono fornite alcune definizioni, estratte un po' dalla classica teoria delle matrici ed un po' dal linguaggio usato dai "sudokisti", che rappresentano i concetti di base utilizzati poi nella descrizione dei metodi risolutivi. Le definizioni, anche se abbastanza sintetiche, non sono date in una forma rigorosissima, tipica di un testo di matematica, cercando un compromesso tra rigore e leggibilità che vada a leggero vantaggio (almeno spero) di quest'ultima.

Definizione 1.

Una **Tavola** Sudoku di ordine n consiste in una **matrice** (o griglia) di $n^2 \times n^2$ celle (o variabili), inizialmente vuote. Ogni Tavola è scomponibile in n^2 **blocchi**, o in n^2 **righe**, o ancora in n^2 **colonne**, tutti di n^2 celle. Alle colonne, alle righe ed ai blocchi di una Tavola, che sono particolari sottomatrici, daremo il nome di **sottomatrici di base**. Ciascuna cella è univocamente determinata dal suo numero di riga e colonna e può contenere un solo simbolo. L'insieme S dei simboli da utilizzare per riempire le celle è dato dai numeri Naturali dall'1 al n^2. Quando una cella viene riempita con un simbolo, in modo definitivo, diciamo la **cella** è stata **assegnata** a quel simbolo. Una Tavola Sudoku è **completa** se a tutte le celle sono stati assegnati dei simboli, viceversa una Tavola Sudoku è **parziale** se le celle risultano parzialmente assegnate.

Definizione 2

Una **soluzione** Sudoku, ovvero un Sudoku risolto, consiste in una Tavola Sudoku che soddisfa due condizioni:
Condizione 1) la Tavola è completa;
Condizione 2) ogni simbolo appartenente ad S compare nelle celle (assegnate) di ciascuna sottomatrice di base nella Tavola.

Il Sudoku più diffuso è quello con n=3, quindi con 9x9=81 celle, e in questo caso abbiamo S = {1,2,3,4,5,6,7,8,9}.

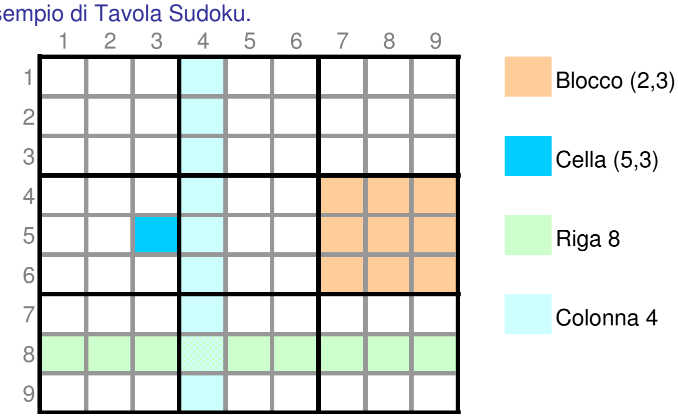
Può capitare di leggere che il Sudoku sia un gioco elaborato con un certo numero di ingredienti riconducibili esclusivamente alla Logica. La Matematica non c'entra. I sostenitori di questa tesi portano come esempio il fatto che i Sudoku potrebbero essere realizzati tranquillamente utilizzando simboli diversi dai nove numeri Naturali dall'1 al 9. Simboli come cerchi, triangoli, e così via, possono sostituire i numeri lasciando inalterate le proprietà dei Sudoku. Vero. Le due condizioni appena descritte valgono per qualunque insieme di simboli.

Osserviamo, però, che alcuni metodi risolutivi sfruttano delle proprietà matematiche derivanti proprio dall'utilizzo di simboli numerici (per es. la somma delle celle di ciascuna riga e colonna deve essere sempre pari a 45). Queste proprietà affiancano e potenziano le caratteristiche dei metodi risolutivi che sfruttano esclusivamente proprietà logiche. Si vedano [1] [2] per degli esempi di approcci risolutivi improntati ai metodi matematici classici della Ricerca Operativa (dove imperversano termini anglosassoni come: serach tree, branch and bound, bipartite matching, network flows, etc.).
D'altro canto è anche vero che questi metodi sono per lo più adatti ad essere tradotti in programmi per computer, per cui in queste pagine, pensate per chi è armato solo di carta e penna, abbiamo potuto riportare solo una parte dei metodi proposti nei documenti appena citati.

Definizione 3

Un **numero Sudoku** rappresenta un numero assegnato ad una cella in una soluzione, cioè il numero giusto al posto giusto. Durante la soluzione di un Sudoku possiamo avere delle celle per le quali esistono più numeri assegnabili che non violano nessun vincolo della Tavola parzialmente assegnata. Questi numeri sono detti **numeri ammissibili** o **candidati** per la cella.

La figura seguente mostra un esempio di Tavola Sudoku.



La Condizione 2 è fondamentale per costruire alcuni metodi risolutivi. Dal fatto che il numero di celle in ciascuna sottomatrice di base è pari al numero di simboli in S, si può facilmente verificare che la Condizione 2 è equivalente alla seguente:

Proposizione 1

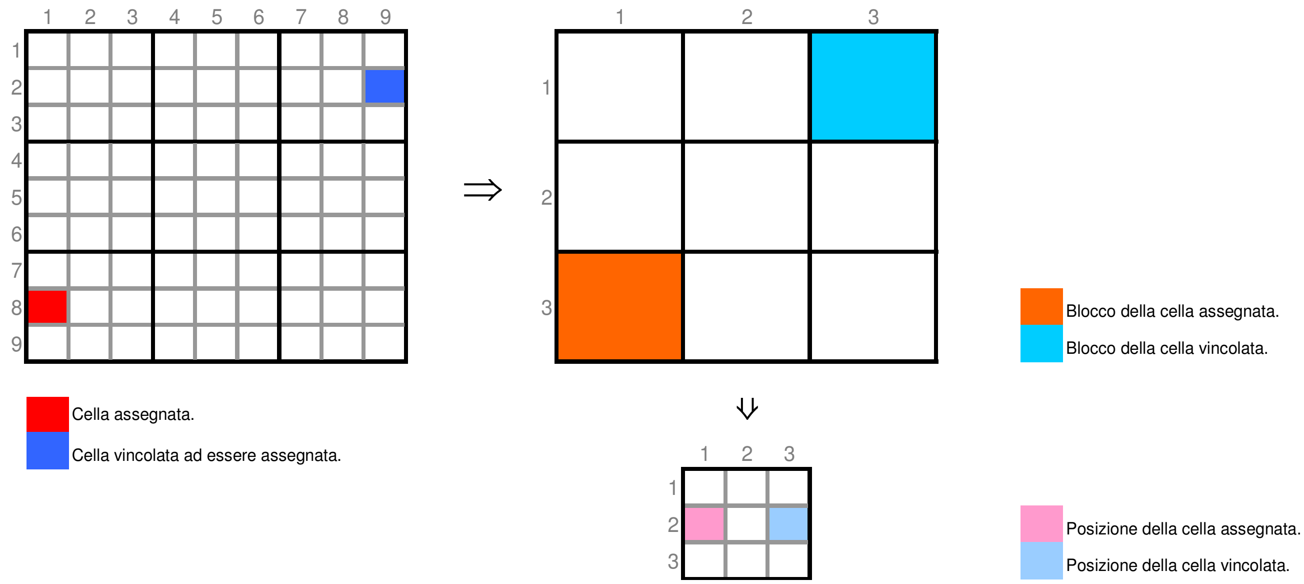
In una Tavola Sudoku, ciascun simbolo dovrà apparire **una sola volta** in ciascuna sottomatrice di base di una soluzione.

Possiamo scrivere una semplice dimostrazione di questa proprietà utilizzando il principio d'induzione. Verifichiamo che per m=1 (una cella e un simbolo) la proprietà è banalmente vera. Supponiamo che sia vera per un m>1 e verifichiamo se è vera per m+1. Risolviamo il problema per un sottoinsieme di m simboli ed m celle (è possibile per ipotesi) e poi assegniamo l'unico simbolo rimasto all'unica cella libera. La proprietà è di conseguenza vera per ogni m>0 e in particolare per $m=n^2$, con n l'ordine del nostro Sudoku.

Definizione 4

Un **problema Sudoku** consiste in una Tavola Sudoku parziale, dove le celle assegnate verificano la seguente proprietà di **simmetria rispetto al centro**: se la cella di posto (i,j) risulta assegnata, allora è assegnata anche la cella di posto (n^2+1-i, n^2+1-j), per i,j = 1,..,n^2. Il giocatore deve provare a completare la Tavola rispettando la Condizione 2, ovvero deve trovare una soluzione che contiene l'assegnazione parziale iniziale. In un problema Sudoku la soluzione dovrebbe essere unica.

La proprietà di simmetria rispetto al centro è evidente sia a livello di blocchi che di celle coinvolte, come mostrato nella figura seguente.



Dato un problema Sudoku, l'unicità della soluzione non è in generale assicurata a priori. In realtà potrebbero esistere diversi modi di completare la Tavola a partire da una assegnazione parziale data. La probabilità che questo sia vero è in qualche modo inversamente proporzionale al numero di celle assegnate nella configurazione di partenza.
Per esempio, in un problema "difficile", con meno di 20 celle assegnate, è più facile trovare soluzioni multiple piuttosto che in un problema "facile" con più di 30 celle assegnate. Coloro che propongono Sudoku si accertano dell'unicità della soluzione con l'uso di programmi ad-hoc che esplorano tutte le possibili soluzioni di un dato problema.

… e adesso la pratica.

La metodologia che seguiremo è tra le più diffuse. Partendo da un problema Sudoku, si applicano ripetutamente alcune tecniche di riduzione dell'insieme dei numeri ammissibili in ciascuna cella non assegnata. Per questo motivo queste tecniche sono spesso chiamate metodi o **regole di riduzione**.
Le regole di riduzione non fanno altro che verificare delle condizioni necessarie per una soluzione, non sufficienti. E' per questo motivo che l'applicazione di una o più regole non garantisce di ottenere una soluzione. Una **condizione necessaria** è del tipo "se x è una soluzione, allora è vera la condizione y"; viceversa una condizione sufficiente sarebbe del tipo "se è vera la condizione w, allora x è una soluzione". Se riuscissimo a trovare una condizione w sufficiente, che si rivelasse vera, riusciremmo a risolvere il Sudoku in un solo passo.

La natura del procedimento è di tipo iterativo e, generalmente, permette di arrivare alla soluzione del Sudoku. Dato un problema Sudoku, il procedimento proposto può essere schematizzato come segue:

Fase iniziale:
Passo 1: in ogni cella libera della tavola scrivi tutti gli n simboli di S;
Passo 2: per ogni cella non assegnata ottieni la lista dei simboli ammissibili, cancellando da ciascuna cella i simboli che, se assegnati, porterebbero a violare la Condizione 2 (Proposizione 1);
Passo 3: ***risolto***=FALSO;

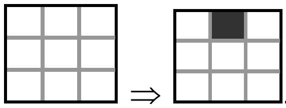
Fase iterativa:
Passo 4: *finchè* ***risolto***=FALSO *ripeti*:
applica Regola 1, Regola 2,…, Regola6;
Passo 5: **FINE**.

Abbiamo supposto che ad ogni applicazione di una Regola, se ci rendiamo conto di aver ottenuto una soluzione, la variabile ***risolto*** assumerà il valore VERO e quindi raggiungeremo il Passo 5, che equivale a terminare la procedura.
In realtà non è detto che l'insieme delle regole di riduzione proposte sia sufficiente a risolvere ogni Sudoku. Per poter risolvere qualunque problema Sudoku occorre una procedura più generale di quella appena descritta, che utilizzi alcuni concetti di Ricerca Operativa, in particolare quello di esplorazione (o visita) esaustiva di un **albero decisionale** (albero di ricerca delle soluzioni o **search tree** [1] [2]). Torneremo più avanti sull'argomento, fornendo una procedura per risolvere qualsiasi problema Sudoku!

Il Passo 1 ci porterà a trasformare una cella vuota in qualcosa di simile a questo:

1	2	3
4	5	6
7	8	9

.
Un metodo alternativo consiste, per esempio, nel suddividere la cella in n sottocelle e fare corrispondere a ciascuna di queste un simbolo. Per cancellare un simbolo, per esempio il numero 2, si procederà annerendo il quadratino corrispondente:



. Quest'ultima notazione posizionale, di tipo binario, è utilizzata spesso nei programmi risolutivi per computer. La particolare codifica si presta bene alle operazioni di cancellazione di un simbolo all'interno di una cella che risulta presente anche in un'altra, attraverso l'uso dei cosiddetti operatori logici bit-a-bit (un **bit** corrisponde ad una sottocella), in particolare attraverso l'uso dell'operatore **XOR** o OR esclusivo.

Adesso passiamo in rassegna le regole di riduzione utilizzabili al Passo 4 della procedura. Ognuna delle regole sfrutta in modo peculiare la struttura della Tavola, con righe, colonne e blocchi e, soprattutto, le relazioni che la Condizione 2 (ovvero la Proposizione 1) impone tra queste sottomatrici di base. Vedremo quindi delle regole che prenderanno in esame singole sottomatrici di base o una coppia riga-colonna, o ancora una colonna e un blocco, e così via.
Per indicare l'insieme delle sottomatrici di base coinvolte utilizzeremo i connettivi logici "∨" ("o" logico) ed "∧" ("e" logico), per cui la scrittura [Blocco ∧ (Colonna ∨ Riga)] sarà relativa ad una regola che coinvolge un Blocco e una Colonna oppure un Blocco e una Riga. Le regole sono elencate per gruppi, in funzione delle sottomatrici coinvolte.

Regola 1: [Blocco∨Riga∨Colonna]

Individuiamo sulla Tavola un sottoinsieme di k celle ($k<n^2$) appartenenti ad una stessa sottomatrice di base, per le quali l'unione dei loro candidati ha cardinalità k (contiene esattamente k numeri). Avendo individuato le k celle con la proprietà descritta, possiamo procedere alla cancellazione delle occorrenze di questi k candidati da tutte le rimanenti celle della sottomatrice di base.

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

⇒

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

Nell'esempio in figura è possibile individuare 3 celle della colonna 2, la cui unione dei candidati è l'insieme {1,2,3} e dunque tali numeri possono essere esclusi dalle rimanenti celle della colonna.
Un **caso particolare della Regola 1** si ottiene ponendo k=1. E' proprio questa la situazione in cui ci troviamo al Passo 2 quando ci viene fornito un problema Sudoku e procediamo al primo tipo di riduzione, eliminando tutte le occorrenze dei numeri assegnati da righe, colonne e blocchi.

Regola 2: [Blocco∨Riga∨Colonna]

Se in una sottomatrice di base k ($k<n^2$) numeri compaiono solo tra i candidati di k celle della sottomatrice e se l'unione degli insiemi ammissibili delle k celle ha cardinalità maggiore di k, allora possiamo cancellare dalle celle individuate tutti i numeri diversi da quei k.

Nell'esempio seguente, i numeri 1 e 2 compaiono tra i candidati di esattamente due tra le celle del blocco 1. Possiamo quindi cancellare da queste celle tutti i candidati diversi dall'1 e dal 2.

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

⇒

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

Regola 3: [Blocco∧(Riga∨Colonna)]

Se k numeri (k<n+1) compaiono su una sola riga (o colonna) all'interno di un blocco, allora non potranno apparire in nessuna altra cella della matrice appartenente alla stessa riga (o colonna) ed esterna al blocco.

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

⇒

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

Regola 4: [Blocco∧(Riga∨Colonna)]

Se in una riga (colonna) esistono k (k<n+1) numeri che compaiono solo nell'intersezione della riga (colonna) con un unico blocco, allora quei numeri non possono apparire nelle due rimanenti righe del blocco in questione.

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

⇒

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

Regola 5: [Blocco∧(Riga∨Colonna)]

Consideriamo l'intersezione di un blocco con una riga (colonna). Notiamo che sia l'unione dei candidati per le celle della riga che per le celle del blocco non appartenenti all'intersezione, devono avere lo stesso sottoinsieme di simboli S'. Possiamo quindi cancellare tutte le occorrenze di candidati della riga esterni all'intersezione che non compaiono anche tra i candidati del blocco esterno all'intersezione.

Nell'esempio seguente abbiamo individuato l'intersezione della prima riga con il primo blocco ed S'={2,4,5,6,7,8}.

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

$\Rightarrow$

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

Regola 6: [Riga∨Colonna]

Per applicare questa regola occorre individuare k ($k<n^2$) righe (o colonne) nelle quali un numero candidato compare solo in k posizioni, le stesse per tutte le k righe (o colonne). Si individua così un quadrato costituito di k^2 celle, dato dall'intersezione delle righe con le colonne in cui compare il candidato, che conterrà k^2 occorrenze del numero prescelto. Ancora una volta, questo significa che le occorrenze del candidato saranno disposte sulle colonne (righe) selezionate e quindi possono essere cancellate, su ogni colonna (riga) della Tavola appartenente anche alla matrice di k^2 celle, nelle righe non incluse in questa sottomatrice.

Nell'esempio abbiamo indicato con "x" un qualunque gruppo di candidati (escluso il numero 2) ragionevolmente diverso da cella a cella. Poichè il numero 2 non compare in nessuna altra cella delle tre righe, si procede con la cancellazione del candidato dalle altre celle delle colonne in cui è presente.

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

$\Rightarrow$

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

Ok, abbiamo finito di elencare le regole di riduzione. Resta da modificare la procedura risolutiva descritta in precedenza, affinché sia in grado di risolvere qualunque Sudoku. Per fare questo dobbiamo cambiare leggermente la struttura della procedura:

Fase iniziale:
Passo 1: in ogni cella libera della tavola scrivi tutti gli n simboli di S;
Passo 2: per ogni cella non assegnata ottieni la lista dei simboli ammissibili, cancellando da ciascuna cella i simboli che, se assegnati, porterebbero a violare la Condizione 2 (Proposizione 1);
Passo 3: ***risolto***=FALSO; ***numero_riduzioni***=1;

Fase iterativa:
Passo 4: *finchè* (***risolto***=FALSO ∧ ***numero_riduzioni***>0), *ripeti*:
applica Regola 1, Regola 2,…, Regola6;
Passo 5: *se* ***risolto***=VERO *allora* **FINE**.
Passo 6: *se* non esistono celle vuote, *allora*:
6.1: scegli un candidato da una cella contenente solo numeri non marchiati (devono essere più di uno) e marchialo come numero Sudoku (per es. cerchialo);
6.2: copia la configurazione attuale su un'altra Tavola e gioca su questa;
altrimenti:
6.3: cancella, dalla Tavola da cui hai ottenuto la copia su cui giochi, l'ultimo numero marchiato come numero Sudoku e torna a giocare su questa Tavola (cancella la copia);
6.4: se nella cella dove hai cancellato il numero è rimasto un solo candidato, marchialo come numero Sudoku;
Passo 7: cancella, nelle celle non assegnate, tutti i simboli che violano la Condizione 2 (Proposizione 1);
Passo 8: ***numero_riduzioni***=1 e vai al Passo 4;

In effetti le cose si sono complicate un tantino, soprattutto nella Fase iterativa, dove è stata innestata una nuova parte finale. Fino al Passo 5, le due procedure sono sostanzialmente identiche.
In questa versione della procedura, abbiamo implicitamente supposto che alla fine dell'applicazione di tutte le regole, si aggiorni il contatore del numero di riduzioni applicate (***numero_riduzioni***), e che si verifichi se il Sudoku è risolto o meno assegnando alla variabile ***risolto*** rispettivamente il valore di VERO oppure FALSO. In effetti, al Passo 8, il contatore viene forzato ad 1 anche se non sono state realmente effettuate riduzioni al passo precedente, ma la forzatura è necessaria per poter verificare la condizione al Passo 4 e ricominciare una nuova fase iterativa.
Inoltre è evidente come adesso la procedura può portarci a giocare su diverse copie di Tavole Sudoku. Questo modo di operare (anche se costringe ad usare più carta!) garantisce una esplorazione completa delle possibili soluzioni intermedie della Tavola, finché non si perviene alla soluzione del Sudoku [1] [2].
Non vi resta che mettere in pratica le tecniche imparate, sul primo malcapitato Sudoku che vi viene a tiro. Buon divertimento.

Bibliografia

[1] W.J. van Hoeve (2001), *The Alldifferent Constraint: A survey*. Proceedings of the sixth annual workshop of the ERCIM Working Group on Constraints.

[2] H. Simonis (2005), *Sudoku as a constraint problem*. Proceedings of the eleventh international conference on Principles and Practice of Constraint Programming.

[3] Sito de "La Repubblica": http://www.repubblica.it

[4] Pagina di Polymath, Politecnico di Torino, dedicata al Sudoku: http://www2.polito.it/didattica/polymath/htmlS/probegio/GAMEMATH/Sudoku/Sudoku.html

[5] Sito di Wikipedia, in lingua italiana, dedicato al Sudoku: http://it.wikipedia.org/wiki/Sudoku