

Cruciverba Numerico

Elaborato per L'Esame di Intelligenza Artificiale

di Paolo Innocenti (5607538)

In questo esercizio si costruisce un modello Minizinc per risolvere il problema di soddisfacimento vincoli di un Cruciverba Numerico.

1 Sinossi

Si consideri la seguente variante del classico cruciverba, dove le caselle bianche dello schema $n \times m$ devono essere riempite con numeri compresi tra due interi assegnati $r > 0$ e in modo che la somma dei numeri (in orizzontale o verticale) produca la definizione. L'esempio risolto riportato in figura (qui $r = 1$ e $s = 9$) dovrebbe essere autoesplicativo.

1	5	2	4		3	6	2	4	1	3	
5	3	5	4	2				7			
6	3	1					7	5	4	8	2
	2		9	3	10	5	5			4	
		11	5	1	2	4	3				

Definizioni

Orizzontali	Verticali
1. 9	1. 13
3. 12	2. 10
5. 14	3. 8
6. 4	4. 12
7. 11	7. 14
9. 13	8. 6
11. 15	9. 4
	10. 7

2 Modello Minizinc

Il problema è stato modellato come un CSP.

2.1 Parametri

I parametri passati tramite file .dzn sono specifici per ogni istanza e si compongono della struttura del cruciverba e delle definizioni. Nello specifico, relativamente alla struttura troviamo il numero di righe **nr** e di colonne **nc** e un array **nr*nc** composto di 1 e di 0 in cui l'1 rappresenta una cella da riempire mentre lo 0 rappresenta una cella non riempibile. Relativamente alle definizioni troviamo due attributi **ndefh** e **ndefv**, che indicano il numero di definizioni in orizzontale e verticale, due attributi **r** e **s**, rappresentanti gli estremi dell'insieme dei numeri naturali da cui scegliere per riempire il cruciverba, e due array **defh** e **defv** contenenti le definizioni orizzontali e verticali del cruciverba. In tali array ogni elemento esprime la cella di origine della definizione, la somma totale delle celle che compongono la definizione e il numero di celle che la compongono.

2.2 Variabili

Il modello si serve di un array **nr*nc, solution**, di interi in cui va ad inserire, in fase di risoluzione, i vari valori che soddisfano i vincoli determinati tra i valori compresi tra **r** ed **s**, definiti nel dataset. In tale array si suppone associato, ad ogni casella, un identificatore compreso tra 1 e $n*m$, usato poi negli array delle definizioni per indicare la casella dalla quale, iniziata la somma sulla lunghezza espressa sempre nell'array delle definizioni, si deve ottenere il valore indicato.

2.3 Vincoli

Il modello si basa su due vincoli. Il primo visita tutte le definizioni orizzontali presenti nell'array **defH** e si occupa di verificare che la somma delle caselle all'interno della riga, partendo da **defH[i,1]** fino a **defH[i,1] + defH[i,3] - 1**, corrisponda a **defH[i,2]**, ovvero che la somma in orizzontale delle caselle, dalla casella di origine fino a una casella barrata o fino alla fine della riga, risulti effettivamente uguale alla definizione. Il secondo effettua lo stesso lavoro ma in verticale, andando quindi a scorrere attraverso le righe, invece che attraverso le colonne, mantenendosi sempre sulla stessa colonna.

In questi due vincoli il ruolo importante è svolto dal riconoscimento della riga e della colonna attraverso l'identificatore ad essa associato. Si è infatti supposto l'incremento dell'identificatore sulle righe come mostrato nell'esempio 1.1.

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35

Schematizzazione identificatori caselle cruciverba

Esempio 1.1

In tale configurazione, la colonna viene ricavata come il resto della divisione tra l'identificatore della cella e il numero di colonne, essendo la colonna il luogo degli identificatori aventi resto della divisione con la variabile **nc** uguale. La riga viene invece ricavata calcolando la divisione tra l'identificatore e il numero di colonne e analizzando poi il resto di tale divisione. In caso di resto nullo, la riga è uguale alla divisione tra l'identificatore e il numero di colonne, altrimenti tale valore viene incrementato di 1.

2.4 Strategia di Ricerca

Per quanto riguarda la strategia di ricerca si è scelto di utilizzare il semplice solve satisfy in quanto si è ritenuto non esistere un criterio per definire una soluzione migliore di un'altra. Infatti, a differenza del cruciverba tradizionale, è facile nel cruciverba numerico avere più soluzioni per un set di definizioni dato.

3 Test

I test sono stati effettuati su 4 diversi input, di cui 3 soddisfacibili e uno no. I test sono stati effettuati su una macchina con processore i5 con frequenza di clock 1.8 GHz, sistema operativo Microsoft Windows 10 a 64bit e 8GB di RAM.

	Set1	Set2	Set3	Set4
Righe(nr)	5	4	4	4
Colonne(nc)	7	4	6	6
Def. Orizzontali(ndefh)	7	4	5	5
Def. Verticali(ndefv)	8	4	4	4
Inizio dominio(r)	1	1	2	5
Fine dominio(s)	9	9	19	9

Risultati Test

	Set1	Set2	Set3	Set4
Tempo(ms)	138	134	162	141