

Relazione Progetto AI

Corso Vignoli

July 2019

1 Introduzione

Il mio progetto ha come scopo la risoluzione di un problema CSP: data una variante del classico cruciverba numerico, si doveva posizionare i numeri sulla scacchiera in modo tale che il massimo dei numeri (in orizzontale o verticale) producesse la definizione nell'esempio sottostante (figura 1).

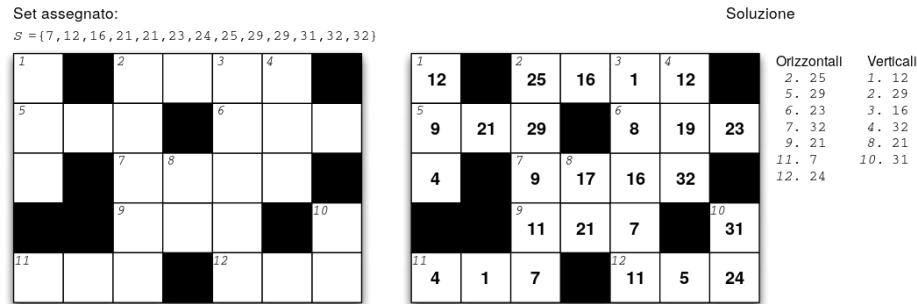


Figure 1: Esempio di risoluzione

2 Implementazione

Come ambiente di sviluppo, ho utilizzato **MiniZinc IDE**, con il solver **Gecode 6.1.1 Built-in**, quindi bisogna accertarsi di avere l'ultima versione scaricata di MiniZinc.

Il mio programma, per funzionare, utilizza 2 file:

- **Numerical-Crossword-Data.dzn** : Sono presenti tutti i dati di cui il programma ha bisogno in ingresso (ad esempio, la rappresentazione del cruciverba, il vettore soluzione, le dimensioni dei segmenti, ecc...). Ce ne sono 3, ognuno un esempio diverso di valori in ingresso al problema.
- **Numerical-Crossword.mzn** : E' presente il codice che processa tutti i vari input presi dal file .dzn e, in caso di problema soddisfacibile, viene mostrato il cruciverba completato con il vettore soluzione sottostante; se il problema fosse insoddisfacibile, il programma mostra in uscita la stringa "====UNSATISFIABLE====".

2.1 Numerical-Crossword-Data.dzn

Vediamo adesso, nel dettaglio, come e' composto ogni file di questo tipo:

- **board** : Viene mostrata la griglia di partenza come un array monodimensionale di interi: gli zeri rappresentano le caselle nere (dove non si puo' scrivere i numeri); mentre i trattini bassi rappresentano le caselle bianche dove verranno inseriti i numeri.
- **segments** : Sono rappresentati i vari segmenti contigui di numeri, sia orizzontali che verticali (nel codice sono commentati con 'h' se il determinato segmento e' orizzontale, 'v' se e' verticale). Lo ho rappresentato come un array bidimensionale di interi.
- **rows/columns** : Numero di righe e di colonne del cruciverba.
- **r/s** : Sono i 2 numeri entro i quali devono essere compresi tutti i numeri che andranno inseriti nel cruciverba.
- **max-length-seg** : Lunghezza massima dei segmenti all' interno del cruciverba.
- **solutions-vector** : Vettore soluzione del problema.
- **sol-card** : Numero di elementi presenti nel vettore soluzione.

2.2 Numerical-Crossword.mzn

Descriviamo adesso il codice vero e proprio che mi ha permesso di risolvere questo problema.

Dopo aver incluso il file .dzn ed aver istanziato gli array, ho creato questi 2 vincoli di cui adesso ne mostrero' il funzionamento:

```
constraint forall(i in 1..sol_card, j in 1..max_length_seg)(if segments[i,j]!=0 then
board[segments[i, j]] <= solutions_vector[i] /\ board[segments[i, j]]>=1 endif);
```

L'array segments e' ordinato in modo tale che ad ogni segmento (prima orizzontale poi il corrispondente verticale se e' presente) corrisponde il numero relativo nel vettore solutions-vector (ad es, al primo segmento 1h di segments corrisponde il primo numero del vettore soluzione; al secondo, il secondo; ecc...).

Viene fatto un doppio ciclo, sia su riga che colonna dell' array segments : se segments[i,j] e' diverso da 0 (ovvero non e' su una casella nera), allora il numero presente sul cruciverba deve essere minore o uguale del numero soluzione corrispondente del vettore, e deve essere maggiore o uguale di 1.

```
constraint forall( [board[segments[i,j]]==solutions_vector[i] /\
board[segments[i,j+1]]==solutions_vector[i] | i in 1..sol_card,
j in 1..max_length_seg-1 where segments[i,j]!=0 /\ segments[i,j+1]!=0]);
```

Con questo ultimo vincolo, scorro per righe il cruciverba e faccio un or logico tra due elementi contigui del segmento, dicendo che uno dei due quindi deve essere uguale al corrispondente numero del vettore soluzione (controllando chiaramente che entrambi i numeri che io vado a controllare siano diversi da zero).

Viene poi mostrato in output, in caso di problema soddisfacibile, il cruciverba completato ed il suo relativo vettore soluzione.

3 Riproduzione dei risultati

Come solver, ho utilizzato **Gecode 6.1.1 Built-in**, quindi bisogna accertarsi di avere l'ultima versione scaricata di MiniZinc.

Per risolvere un determinato problema, basta solo cambiare, all' interno del file .mzn, l'inclusione del file .dzn contenente il problema corrispondente (prima riga di codice).