Laboratorio di Programmazione a.a. 2015-16

Problema 12

4 Aprile 2016

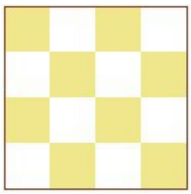
Descrizione – Parte I

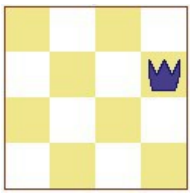
Il protocollo del dato astratto *"scacchiera"* è definito dalle procedure introdotte in classe: `empty-board`, `size`, `queens-on`, `under-attack?`, `add-queen` e `arrangement`. Il file `queens.scm` associato a questo problema contiene le soluzioni già sviluppate e suggerisce un'impostazione di massima per realizzare il protocollo, dove una scacchiera n x n su cui sono collocate q regine è rappresentata da (una lista di) 7 elementi: n; q; quattro liste di righe, di colonne, di diagonali discendenti e di diagonali ascendenti che sono sotto scacco da parte di una regina; infine, l'ultimo elemento è una descrizione della configurazione. Righe e colonne possono essere codificate dai corrispondenti indici; le diagonali discendenti dalla differenza degli indici che è invariante; quelle ascendenti dalla somma degli indici.

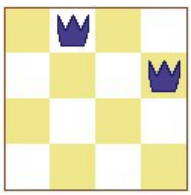
Completa le definizioni delle procedure che realizzano il protocollo in base alle indicazioni fornite sopra e sperimenta il programma che determina il *numero* di soluzioni del rompicapo delle n regine per n = 1, 2, 3, 4, 5, 6, 7, 8. A tal fine la componente "descrizione" è irrilevante e può essere utilizzato il medesimo valore (p.es. una stringa prefissata) per qualsiasi scacchiera.

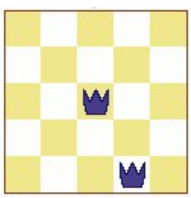
Descrizione – Parte II

A questo punto si vuole intervenire sulla rappresentazione della scacchiera in modo tale da poter visualizzare attraverso un'*immagine* le disposizioni di regine sulla scacchiera. Più precisamente, la procedura `arrangement` deve restituire un'immagine della configurazione, come illustrato dagli esempi che seguono:

```
(arrangement
  (empty-board 4)
  )
```
→ 

```
(arrangement
  (add-queen (empty-board 4) 2 4)
  )
```
→ 

```
(arrangement
  (add-queen (add-queen (empty-board 4) 2 4) 1 2)
  )
```
→ 

```
(arrangement
  (add-queen (add-queen (empty-board 5) 3 3) 5 4)
  )
```
→ 

Apporta le integrazioni appropriate a partire dalla realizzazione di cui sopra, avendo cura di non modificare il protocollo concordato (e, di conseguenza, di poter lasciare inalterato il programma che risolve il rompicapo sulla base di quel protocollo).

Per sviluppare il programma con il supporto del *teachpack* `drawings.ss` (lo stesso già utilizzato per affrontare precedenti problemi), aggiungi il comando Scheme contenuto nel file `queens_images.rkt`, anch'esso associato a questo foglio. Il teachpack `drawings.ss` mette a disposizione due procedure il cui valore è un'*immagine* della scacchiera con le eventuali regine collocate su di essa (immagine analoga a quelle degli esempi riportati sopra):

- `(chessboard-image` <*n*>`)` — immagine di una scacchiera vuota di dimensione *n* x *n*
- `(add-queen-image` <*i*> <*j*> <*n*> <*img*>`)` — immagine che si ottiene aggiungendo l'icona di una regina nella posizione di coordinate *i*, *j* all'immagine *img* che raffigura una scacchiera *n* x *n* con eventuali altre regine (*n* deve essere consistente con *img*)

Per capire meglio il ruolo degli strumenti introdotti conviene sperimentare la valutazione di qualche semplice espressione basata sulle due procedure introdotte sopra.

Sperimenta infine il programma che determina la lista delle soluzioni per *n* = 1, 2, 3, 4, 5, 6.