

# Relazione per l'esame di Intelligenza Artificiale

Gabriele Stefani 6342913

10 Gennaio 2020

## 1 Introduzione

In questo elaborato è stato implementato l'algoritmo **Forward Checking** per risolvere **Constraint Satisfaction Problem** (CSP) binari applicato come inferenza nell'algoritmo **Backtracking Search**. I 3 problemi presi in considerazione sono:

- Map Coloring
- Sudoku
- NQueens

Per ognuno di essi sono stati codificati: le variabili, i domini di ciascuna di esse, le variabili unite da un vincolo (obbligatoriamente binario) e i vincoli.

## 2 Algoritmi

L'algoritmo **Forward Checking** è usato come inferenza, ad ogni assegnazione di un valore ad una variabile riduce i domini delle variabili vicine, in modo da rispettare i vincoli tra esse. In questo modo il numero di conflitti tra variabili adiacenti è sempre 0. Per la ricerca della soluzione invece viene implementato l'algoritmo di **Backtracking Search** che riceve in ingresso un CSP e restituisce un dizionario in cui a ogni variabile è associato un valore che è soluzione del problema (ovvero soddisfa tutti i vincoli). Ad ogni passo viene scelta come variabile a cui assegnare un valore la prima della lista delle variabili non ancora assegnate (la scelta dipende dunque dal modo in cui la lista è creata). Anche l'ordine con cui vengono assegnati i valori del dominio è quello della lista in cui è salvato, e dipende quindi dall'ordine di inserimento all'interno di essa.

### 3 CSP generico

La base del codice, oltre a fornire gli algoritmi sopra citati, definisce anche una classe che descrive un CSP binario generico (chiamata CSP) in base a 4 strutture dati:

- Variabili: lista delle variabili del problema.
- Domini: dizionario che associa ad ogni variabile la lista di valori del dominio
- Vicini: dizionario che associa ad ogni variabile la lista di variabili che hanno un vincolo con essa
- Vincolo: funzione che restituisce vero se il vincolo tra due generiche variabili alle quali è stato assegnato un valore è rispettato e falso altrimenti.

Internamente esiste un altro dizionario che associa ad ogni variabile il dominio che viene ridotto ad ogni passo della **Backtracking Search** dall'inferenza del **Forward Checking**.

### 4 Problemi

#### 4.1 Map Coloring

Il *Map Coloring* è stato codificato come problema di CSP binario utilizzando come variabili la lista degli stati da colorare, come dominio di ogni variabile la lista dei colori utilizzabili, come vicini gli stati confinanti tra loro e come vincolo la diversità tra i 2 valori assegnati a 2 variabili vicine.

#### 4.2 Sudoku

Nel *Sudoku* le variabili sono le celle, i domini di ogni variabile sono i valori da 1 a 9 tranne nei casi in cui a una cella sia già stato assegnato un valore, in tal caso il dominio è il valore stesso. Il vincolo è ancora una volta il vincolo di diversità per cui i vicini di ogni variabile (o cella) sono tutte le celle nello stesso quadrante, nella stessa riga e nella stessa colonna.

#### 4.3 NQueens

*NQueens* è stato codificato pensando che ogni regina deve essere disposta su una colonna diversa, altrimenti le regine si mangerebbero sicuramente tra

loro. Ogni variabile corrisponde quindi ad una colonna e il dominio sono le righe. Usando delle coordinate  $(x, y)$  per indicare una posizione sulla scacchiera, le variabili corrispondono alle  $y$  e i valori del dominio alle  $x$ . Il vincolo sfrutta proprio queste coordinate per capire se due regine si trovino sulla stessa diagonale o sulla stessa riga. Per cui ogni variabile ha come vicini tutte le altre variabili in modo che nessuna regina si trovi sulla stessa riga o sulla stessa diagonale.

## 5 Prestazioni

La risoluzione di *Sudoku* e *NQueens* attraverso l'algoritmo **Forward Checking** non è la più efficiente. La risoluzione di *NQueens* con  $n$  molto grandi può essere molto lunga così come risolvere alcuni sudoku può portare a tempi di attesa notevoli.

## 6 Riferimenti

La maggioranza delle informazioni sono state recuperate da *Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach. 3rd edition. Pearson, 2010.*

Per la codifica di *NQueens* si rimanda questo problema di CSPLib.

Il codice e la codifica dei restanti problemi sono stati presi dal repository `aima-python`.

La scelta di CSP binari deriva invece dalla lettura di questo articolo.