

## 1. Ricerca non informata

1. *ricerca per ampiezza* = per righe
2. *ricerca in profondità* = per colonne
  - limitata ad  $L$ : mi fermo a riga  $L$
  - iterativa = ad ogni iterazione riparto da radice e aumento  $L$
3. *ricerca a costo uniforme* (Dijkstra) = aggiorno frontiera espandendo nodo con costo del cammino minore e aggiorno nodi che posso raggiungere da nodo espanso

## 2. Ricerca informata

1. *best-first-greedy* = aggiorno frontiera espandendo nodo con valore minore di  $f$  (funzione euristica di distanza)
2.  $A^*$  = aggiorno frontiera espandendo nodo con valore minore di  $f + g$  (distanza + costo cammino)

## 3. CSP

1. formulare problema = insieme delle variabili  $X$ , insieme dei domini  $D$ , insieme dei vincoli  $C$
2. disegnare grafo vincoli
3. algoritmo AC3 =
  - rendere archi orientati
  - tutti gli archi in Arc
  - estraggo e rimuovo un arco da Arc
  - se devo, modifico dominio nodo da cui parte arco
  - se ho modificato il dominio di nodo  $X$ , aggiungo ad Arc tutti gli archi entranti in  $X$
  - mi fermo quando Arc è vuoto
4. backtracking
  - scelta della variabile
    - MRV: variabile con minor numero di valori legali rimanenti
    - Degree = variabile coinvolta nel maggior numero di vincoli
  - ordine di prova dei valori
    - dato dal testo
    - LCV = valore che lascia maggior numero di scelte possibili per le variabili adiacenti
  - inferenza
    - FC = ogni volta che la variabile  $x$  viene assegnata, si stabilisce la sua consistenza d'arco: per ogni  $y$  collegata a  $x$  si eliminano valori del dominio di  $y$  non legali per  $x$

## 4. Logica proposizionale

1. *formalizzare frasi*
2. conseguenze logiche vere o false con *model checking*
  - modello =  $K$  modello per  $x$  se  $x$  vera in  $K$
  - conseguenza logica =  $K \models \alpha$  se e solo se  $M(k) \subseteq M(\alpha)$
  - quindi calcolo tabelle di verità di  $k$  e di  $\alpha$ , vedo dove sono vere e paragono risultati
3. *soddisfatta* (vera in almeno un'interpretazione) e *tautologia* (vera in tutte le interpretazioni)
4. converire in *CNF* = and di clausole, che sono or di letterali
  1. eliminare  $\leftrightarrow$
  2. eliminare  $\rightarrow$
  3. portare  $\neg$  davanti a singole variabili

4. proprietà distributive di  $\wedge$  e  $\vee$
5. verificare conseguenze logiche con *algoritmo di risoluzione*
  - per dimostrare che  $k \models \alpha$ , dimostra che  $k \wedge \neg\alpha$  è insoddisfacibile (cioè arriviamo ad una clausola vuota):
    - converto in CNF e separo clausole genero nuove clausole con principio di risoluzione:  $\frac{c_1, c_2}{c_1 \vee c_2}$  semplificando (es.  $\cancel{A} \vee \cancel{B} \vee \neg\cancel{A} \vee B \vee E$ )
    - se finisco clausole allora  $k \not\models \alpha$ , se arrivo a clausola vuota allora  $k \models \alpha$
6. *clausole di Horn* (contengono al più un letterale positivo) e *forward chaining* =
  - conto  $[c]$  = per ogni clausola  $c$ , indica quante premesse mancano da valutare  
inferiti  $[s]$  = per ogni simbolo  $s$ , true se lo abbiamo già dedotto. Inizio tutti false  
coda = coda dei simboli che inizialmente contiene quelli noti come veri
  - finché coda non è vuota:
    - estrai un simbolo  $p$  dalla coda
    - se  $p = 1q$  abbiamo già trovato quello che cercavamo
    - se  $p$  non è ancora stato inferito:
      - segna  $p$  come inferito
      - per ogni clausola  $c$  che ha  $p$  tra le premesse: riduci conteggio delle premesse non ancora soddisfatte (conto  $[c]$ ) e se tutte le premesse sono state soddisfatte (conto = 0) aggiungi  $c$  in coda