

Vincoli

'Programming with constraints' Capitolo 1

1



Vincoli

- ◆ Cosa sono i vincoli?
- ◆ Modellare i problemi reali
- ◆ Risolvere un problema con vincoli
- ◆ Vincoli sugli alberi
- ◆ Altre classi di vincoli
- ◆ Proprieta' della soluzione di problemi con vincoli

2



Vincoli

Variabile: un place holder per valori

$X, Y, Z, L_3, U_{21}, List$

Simboli di funzione: mapping di valori a valori

$+, -, \times, \div, \sin, \cos, ||$

Simboli di relazione: relazioni tra valori

$=, \leq, \neq$

3



Vincoli

Vincolo primitivo: relazione con argomenti

$$X \geq 4$$

$$X + 2Y = 9$$

Vincolo: congiunzione di vincoli primitivi

$$X \leq 3 \wedge X = Y \wedge Y \geq 4$$

4



Soddisfacibilita'

Valutazione: un assegnamento di valori a variabili $\theta = \{X \mapsto 3, Y \mapsto 4, Z \mapsto 2\}$

$$\theta(X + 2Y) = (3 + 2 \times 4) = 11$$

Soluzione: valutazione che soddisfa i vincoli

$$\theta(X \geq 3 \wedge Y = X + 1)$$

$$= (3 \geq 3 \wedge 4 = 3 + 1) = true$$

5



Soddisfacibilita'

Soddisfacibile: vincolo che ha una soluzione

Non soddisfacibile: vincolo che non ha una soluzione

$$X \leq 3 \wedge Y = X + 1$$

soddisfacibile

$$X \leq 3 \wedge Y = X + 1 \wedge Y \geq 6$$

Non

soddisfacibile

6



Vincoli e modello a stati

Stato: variabili con valori da un dominio

Test per il goal: vincoli che specificano le combinazioni permesse di valori per delle variabili

7



Esempio

N-regine:

- Variabili Q_1, \dots, Q_n
- Domini $D_i = \{1, \dots, n\}$
- Vincoli di non attacco

8



Esempio

Cripto-aritmetica:

- Es.: SEND+MORE=MONEY
- Variabili S,E,N,D,M,O,R,Y
- Domini $D_i = \{0, 1, \dots, 9\}$
- Vincoli:
- $M \neq 0, S \neq 0$, somma, all-different

9



Esempio

Colorazione di mappe:

- In modo che stati adiacenti non abbiano lo stesso colore
- Variabili stati S_1, \dots, S_n
- Domini: insieme dei colori
- Vincoli: all-different

10



Esempi reali

- Problemi di assegnamento:
 - Es.: chi insegna quale classe
- Problemi di orario: quale classe e' in quale orario?
- Configurazioni hardware
- Spreadsheets
- Scheduling di trasporti
- Scheduling di attivita'

11



Albero di ricerca

- Nodo: rappresenta uno stato (variabili con domini di lavori)
- Goal: assegnamento di valori a tutte le variabili tale che i vincoli siano soddisfatti
- Stato iniziale: nessuna variabile istanziata
- Funzione successore: istanziamento di una variabile
- Lo stesso per tutti i problemi con vincoli

12



Ricerca o no?

- Strategie di ricerca viste prima: tutte applicabili ai problemi con vincoli
- Euristiche specifiche per i problemi con vincoli (per scelta della variabile da istanziare e scelta del valore per la variabile, cioè la scelta del nodo da espandere)
- Classi di problemi con vincoli che non hanno bisogno di ricerca → risolvibili polinomialmente

13



Vincoli come sintassi

- ◆ I vincoli sono stringhe di simboli
- ◆ L'ordine non importa
- ◆ Ma alcuni algoritmi dipendono dall'ordine

$$X = 0 \wedge Y = 1 \wedge Z = 2 \neq Y = 1 \wedge Z = 2 \wedge X = 0$$

14



Vincoli equivalenti

Due vincoli diversi possono rappresentare la stessa informazione

$$X > 0 \leftrightarrow 0 < X$$

$$X = 1 \wedge Y = 2 \leftrightarrow Y = 2 \wedge X = 1$$

$$X = Y + 1 \wedge Y \geq 2 \leftrightarrow X = Y + 1 \wedge X \geq 3$$

Due vincoli sono **equivalenti** se hanno lo stesso insieme di soluzioni

15



Modellare con i vincoli

- ◆ I vincoli descrivono il comportamento idealizzato di oggetti nel mondo reale

$$V1 = I1 \times R1$$

$$V2 = I2 \times R2$$

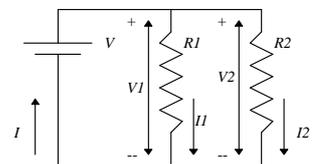
$$V - V1 = 0$$

$$V - V2 = 0$$

$$V1 - V2 = 0$$

$$I - I1 - I2 = 0$$

$$-I + I1 + I2 = 0$$

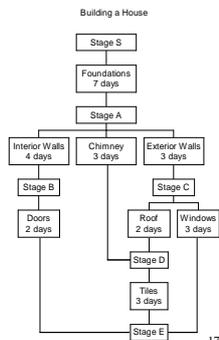


16



Modellare con i vincoli

- start $T_S \geq 0$
- foundations $T_A \geq T_S + 7$
- interior walls $T_B \geq T_A + 4$
- exterior walls $T_C \geq T_A + 3$
- chimney $T_D \geq T_A + 3$
- roof $T_D \geq T_C + 2$
- doors $T_E \geq T_B + 2$
- tiles $T_E \geq T_D + 3$
- windows $T_E \geq T_C + 3$



17



Soddisfazione di vincoli

- ◆ Dato un vincolo C, due domande
 - ◆ **soddisfazione**: ha una soluzione?
 - ◆ **soluzione**: trovare una soluzione, se ce n'è una
- ◆ La prima domanda è più di base
- ◆ Un **risolutore di vincoli** risponde al problema della soddisfazione

18



Soddisfazione di vincoli

- ◆ Come rispondiamo a questa domanda?
- ◆ Approccio piu' semplice: tentare tutte le valutazioni

| $X > Y$ | $X > Y$ |
|--------------------------------------|--------------------------------------|
| $\{X \mapsto 1, Y \mapsto 1\}$ false | $\{X \mapsto 1, Y \mapsto 1\}$ false |
| $\{X \mapsto 1, Y \mapsto 2\}$ false | $\{X \mapsto 2, Y \mapsto 1\}$ true |
| $\{X \mapsto 1, Y \mapsto 3\}$ false | $\{X \mapsto 2, Y \mapsto 2\}$ false |
| • | $\{X \mapsto 3, Y \mapsto 1\}$ true |
| • | $\{X \mapsto 3, Y \mapsto 2\}$ true |
| • | • |

19



Soddisfazione di vincoli

- ◆ Il metodo enumerativo non funziona con i numeri reali
- ◆ Una versione piu' intelligente sara' usata per vincoli a dominio finito
- ◆ Come risolviamo i vincoli sui reali?
- ◆ Ricordiamo l'eliminazione di Gauss-Jordan dalle scuole medie

20



Eliminazione di Gauss-Jordan

- ◆ Scegliamo un equazione c da C
- ◆ Riscriviamo c nella forma $x = e$
- ◆ Rimpiazziamo x , ovunque appaia in C , con e
- ◆ Continuiamo finche'
 - ◆ Tutte le equazioni sono nella forma $x = e$
 - ◆ O una equazione e' equivalente a $d = 0$ ($d \neq 0$)
- ◆ Ritorniamo *true* nel primo caso, altrimenti *false*

21



Gauss-Jordan: Esempio 1

$$\begin{array}{l} 1 + X = 2Y + Z \wedge \quad 1 + X = 2Y + Z \\ Z - X = 3 \wedge \\ X + Y = 5 + Z \end{array}$$

Rimpiazza X con $2Y + Z - 1$

$$\begin{array}{l} X = 2Y + Z - 1 \wedge \\ Z - 2Y - Z + 1 = 3 \wedge \quad -2Y = 2 \\ 2Y + Z - 1 + Y = 5 + Z \end{array}$$

Rimpiazza Y con -1

$$\begin{array}{l} X = -2 + Z - 1 \wedge \\ Y = -1 \wedge \\ -2 + Z - 1 - 1 = 5 + Z \quad -4 = 5 \end{array}$$

Ritorna *false*

22



Gauss-Jordan: Esempio 2

$$\begin{array}{l} 1 + X = 2Y + Z \wedge \quad 1 + X = 2Y + Z \\ Z - X = 3 \end{array}$$

Rimpiazza X con $2Y + Z - 1$

$$\begin{array}{l} X = 2Y + Z - 1 \wedge \\ Z - 2Y - Z + 1 = 3 \quad -2Y = 2 \end{array}$$

Rimpiazza Y con -1

$$\begin{array}{l} X = Z - 3 \wedge \\ Y = -1 \end{array}$$

Forma risolta: vincoli in questa forma sono soddisfacenti

23



Forma risolta

- ◆ **Variabile non parametrica:** appare sulla sinistra di un'equazione
- ◆ **Variabile parametrica:** appare sulla destra di alcune equazioni
- ◆ **Soluzione:** scegli i valori dei parametri e determina i non-parametri

$$\begin{array}{l} X = Z - 3 \wedge \longrightarrow Z = 4 \longrightarrow X = 4 - 3 = 1 \\ Y = -1 \end{array}$$

24



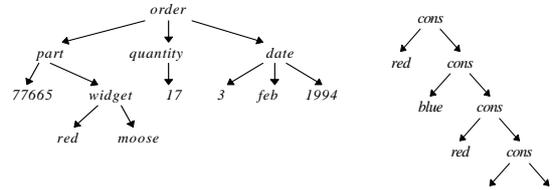
Vincoli sugli alberi

- ◆ Rappresentano dati strutturati
- ◆ **Costruttori sugli alberi:** stringa di caratteri
 - ◆ *cons, node, null, widget, f*
- ◆ **Costante:** costruttore o numero
- ◆ **Albero:**
 - ◆ Una costante e' un *albero*
 - ◆ Un costruttore con una lista di > 0 alberi e' un *albero*
 - ◆ Disegnato con il costruttore sopra i figli

25



Esempi



$order(part(77665, widget(red, moose)),$
 $quantity(17), date(3, feb, 1994))$

$cons(red, cons(blue, cons$
 $(red, cons(...))))$

26



Vincoli sugli alberi

- ◆ **Altezza di un albero:**
 - ◆ Una costante ha altezza 1
 - ◆ Un albero con figli t_1, \dots, t_n ha altezza uno piu' della massima altezza degli alberi t_1, \dots, t_n
- ◆ **Albero finito:** ha altezza finita
- ◆ Esempi: altezza 4 e altezza

27

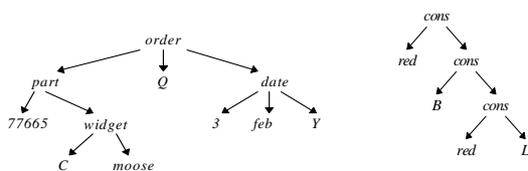


Termini

- ◆ Un *termine* e' un albero con variabili che sostituiscono i sottoalberi
- ◆ **Termine:**
 - ◆ Una costante e' un *termine*
 - ◆ Una variabile e' un *termine*
 - ◆ Un costruttore con una lista di > 0 termini e' un *termine*
 - ◆ Disegnato con il costruttore sopra i figli
- ◆ **Equazione di termini:** $s = t$ (s, t termini) ²⁸



Esempi di termini



$order(part(77665, widget(C, moose)),$
 $Q, date(3, feb, Y))$

$cons(red, cons(B, cons(r$
 $ed, L)))$

29



Risolutore per vincoli sugli alberi

- ◆ Assegnare alberi a variabili in modo che i due termini siano uguali
- ◆ *Es.:* $cons(R, cons(B, nil)) = cons(red, L)$
 $\{R \mapsto red, L \mapsto cons(blue, nil), B \mapsto blue\}$
- ◆ Simile al metodo di Gauss-Jordan
- ◆ Inizia con un insieme di equazioni di termini C e un insieme vuoto di equazioni di termini S
- ◆ Continua finche' C e' vuoto o ritorna *false*

30



Soluzione di vincoli sugli alberi

- ◆ unify(C)
 - ◆ Rimuove l'equazione c da C
 - ◆ case $x=x$: non fa niente
 - ◆ case $f(s1, \dots, sn)=g(t1, \dots, tn)$: **return false**
 - ◆ case $f(s1, \dots, sn)=f(t1, \dots, tn)$:
 - ◆ aggiunge $s1=t1, \dots, sn=tn$ a C
 - ◆ case $t=x$ (x variabile): aggiunge $x=t$ a C
 - ◆ case $x=t$ (x variabile): aggiunge $x=t$ a S
 - ◆ sostituisce t per x ovunque in C e S

31



Esempio

| | |
|---|---|
| C | S |
| $cons(Y, nil) = cons(X, Z) \wedge Y = cons(a, T)$ | <i>true</i> |
| $Y = X \wedge nil = Z \wedge Y = cons(a, T)$ | <i>true</i> |
| $nil = Z \wedge X = cons(a, T)$ | $Y = X$ |
| $Z = nil \wedge X = cons(a, T)$ | $Y = X$ |
| $X = cons(a, T)$ | $Y = X \wedge Z = nil$ |
| <i>true</i> | $Y = cons(a, T) \wedge Z = nil \wedge X = cons(a, T)$ |

Come Gauss-Jordan, le variabili sono parametri o non-parametri. Una soluzione e' ottenuta settando I parametri (T) ad un qualsiasi valore.

$$\{T \mapsto nil, X \mapsto cons(a, nil), Y \mapsto cons(a, nil), Z \mapsto nil\}$$

32



Un caso particolare

- ◆ C'è una soluzione per $X = f(X)$?
- ◆ NO!
 - ◆ Se l'altezza di X nella soluzione e' n
 - ◆ allora $f(X)$ ha altezza $n+1$
- ◆ **Occurs check:**
 - ◆ Prima di sostituire t per x
 - ◆ Controlla che x non appare in t

33



Altri domini di vincoli

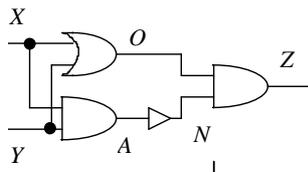
- ◆ Ce ne sono molti:
 - ◆ Vincoli booleani
 - ◆ Vincoli di sequenze
 - ◆ Mondo dei blocchi
- ◆ Molti altri, di solito collegati a una qualche struttura matematica

34



Vincoli booleani

Usati per modellare circuiti, ...



Circuito per l'or esclusivo

$$\begin{aligned} O &\leftrightarrow (X \vee Y) \wedge \\ A &\leftrightarrow (X \& Y) \wedge \\ N &\leftrightarrow \neg A \wedge \\ Z &\leftrightarrow (O \& N) \end{aligned}$$

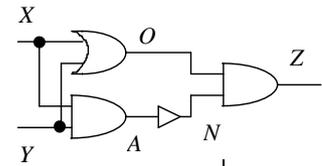
Vincolo booleano che descrive il circuito

35



Vincoli booleani

$$\begin{aligned} \neg FO &\leftrightarrow (O \leftrightarrow (X \vee Y)) \wedge \\ \neg FA &\leftrightarrow (A \leftrightarrow (X \& Y)) \wedge \\ \neg FN &\leftrightarrow (N \leftrightarrow \neg A) \wedge \\ \neg FG &\leftrightarrow (Z \leftrightarrow (N \& O)) \end{aligned}$$



Vincolo che modella il circuito con variabili per i guasti
 $\neg(FO \& FA) \wedge \neg(FO \& FN) \wedge \neg(FO \& FG) \wedge$
 $\neg(FA \& FN) \wedge \neg(FA \& FG) \wedge \neg(FN \& FG)$

Vincolo che modella che solo un gate e' guasto

Comportamento osservato: $\{X \mapsto 0, Y \mapsto 0, Z \mapsto 1\}$

Soluzione: $\{FO \mapsto 1, FA \mapsto 0, FN \mapsto 0, FG \mapsto 0,$
 $X \mapsto 0, Y \mapsto 0, O \mapsto 1, A \mapsto 0, N \mapsto 1, Z \mapsto 1\}$

36



Risolutore per vincoli booleani

Sia m il numero di vincoli primitivi in C

$n := \left\lceil \frac{\ln(\epsilon)}{\ln(1 - \frac{1}{m})} \right\rceil$ *epsilon e' tra 0 e 1 e determina il grado di incompletezza*

for $i := 1$ to n **do**

genera una valutazione random sulle variabili in C

if la valutazione soddisfa C **then return true** **endif**

endfor

return unknown

37



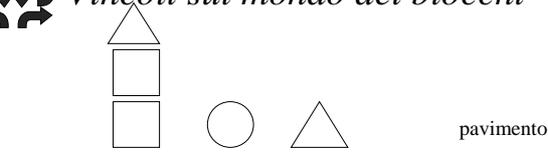
Vincoli booleani

- ◆ Il risolutore booleano puo' ritornare *unknown*
- ◆ E' **incompleto** (non risponde a tutte le domande)
- ◆ E' polinomiale in tempo, mentre un risolutore completo e' esponenziale (a meno che $P = NP$)

38



Vincoli sul mondo dei blocchi



I vincoli non sono solo matematici

Gli oggetti possono essere sul pavimento o su un altro oggetto. La fisica dice quali posizioni sono stabili. I vincoli primitivi sono, per es., $red(X)$, $on(X,Y)$, $not_sphere(Y)$.

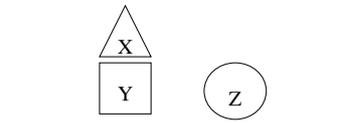
39



Vincoli sul mondo dei blocchi

Una soluzione e' una figura con una annotazione che dice quale variabile e' quale blocco

$yellow(Y) \wedge$
 $red(X) \wedge$
 $on(X,Y) \wedge$
 $floor(Z) \wedge$
 $red(Z)$



40



Definizione di risolutore

- ◆ Un **risolutore di vincoli** e' una funzione $solv$ che prende un vincolo C e ritorna *true*, *false* o *unknown* in dipendenza dal fatto se il vincolo e' soddisfacibile
 - ◆ se $solv(C) = true$ allora C e' soddisfacibile
 - ◆ se $solv(C) = false$ allora C e' non soddisfacibile

41



Proprieta' dei risolutori

- ◆ Vogliamo risolutori che hanno certe proprieta'
- ◆ **well-behaved:**
 - ◆ **basati su insiemi:** la risposta dipende solo dall'insieme dei vincoli primitivi
 - ◆ **monotonici:** se il risolutore fallisce per $C1$ allora fallisce anche per $C1 \wedge C2$
 - ◆ **indipendenti dal nome delle variabili:** il risolutore da' la stessa risposta indipendentemente dal nome delle variabili

$solv(X > Y \wedge Y > Z) = solv(T > U_1 \wedge U_1 > Z)$

42



Proprieta' dei risolutori

- ◆ La proprieta' piu' stringente che possiamo chiedere
- ◆ **completezza:** A un risolutore e' completo se ritorna sempre la risposta *true* o *false* (mai *unknown*)

43



Sommario sui vincoli

- ◆ I vincoli sono usati per modellare il comportamento del mondo reale
- ◆ Un risolutore di vincoli determina se un vincolo ha una soluzione
- ◆ Aritmetica reale e vincoli sugli alberi
- ◆ Proprieta' dei risolutori

44