Constraint Programming & Planning

Agostino Dovier

Università di Udine
Dipartimento di Matematica e Informatica

Udine, Ottobre 2012

Constraint Solving

Constraint Solver

Un risolutore di vincoli è una procedura che trasforma un CSP $\mathcal{P}$ in uno equisoddisfacibile (o equivalente). Si dice:

COMPLETO se, dato $\mathcal{P}$, lo trasforma in un CSP o in una disgiunzione finita di CSP ad esso equisoddisfacibile, e tale che da ciascuno dei disgiunti sia immediato trarre ogni sua soluzione; se $\mathcal{P}$ è inconsistente, viene restituito `fail`.

INCOMPLETO se non è completo. Intuitivamente, dato $\mathcal{P}$, lo trasforma in un CSP più semplice ma non ancora abbastanza semplice.

Constraint Solving

Proof rules

Un constraint solver è basato sull'applicazione ripetuta di regole di dimostrazione

$$\frac{\varphi}{\psi}$$

ove φ e ψ sono dei CSP. In ψ possono occorrere anche variabili non presenti in φ. Una regola mantiene l'equivalenza (è *equivalence preserving*) se $Sol(\varphi) = Sol(\psi)|_{FV(\varphi)}$, ovvero φ e ψ sono equisoddisfacibili.

Constraint Solving

Domain Reduction Rules

Si vuole restringere i domini sfruttando le informazioni presenti nei vincoli. Le regole hanno la forma:

$$\frac{\varphi = \langle \mathcal{C}; \mathcal{D}_\in \rangle}{\psi = \langle \mathcal{C}'; \mathcal{D}'_\in \rangle}$$

- $\mathcal{D}_\in = X_1 \in \mathcal{D}_1, \ldots, X_k \in \mathcal{D}_k$,
- $\mathcal{D}'_\in = X_1 \in \mathcal{D}'_1, \ldots, X_k \in \mathcal{D}'_k$,
- per ogni $i = 1, \ldots, k$ vale che $\mathcal{D}'_i \subseteq \mathcal{D}_i$, e
- $\mathcal{C}'$ è la restrizione di $\mathcal{C}$ ai nuovi domini delle variabili.

Alcuni vincoli in $\mathcal{C}'$ possono diventare ridondanti e dunque rimossi. Se tutti i vincoli si possono rimuovere, $\mathcal{C}' = \texttt{true}$ ($\mathcal{D}'_\in$ rappresenta tutte le soluzioni).
Se invece un $\mathcal{D}'_i = \emptyset$ allora $\psi = \texttt{fail}$.

Constraint Solving

Domain reduction rules: esempio

$$\frac{\langle X < Y; X\ in\ 5..15, Y\ in\ 0..10\rangle}{\langle X < Y; X\ in\ 5..9, Y\ in\ 6..10\rangle}$$

Abbiamo che:

$$\begin{aligned}\mathcal{C} = \mathcal{C}' = \ \{ \quad & (5,6),(5,7),(5,8),(5,9),(5,10),\\ & (6,7),(6,8),(6,9),(6,10),\\ & (7,8),(7,9),(7,10),\\ & (8,9),(8,10),\\ & (9,10)\}\end{aligned}$$

Constraint Solving

Transformation Rules

I vincoli sono trasformati/semplificati. È possibile che la semplificazione introduca nuove variabili a cui va assegnato un dominio. Può essere introdotto il vincolo `fail`.

$$\frac{\varphi = \langle \mathcal{C}; \mathcal{D}_\in \rangle}{\psi = \langle \mathcal{C}'; \mathcal{D}'_\in \rangle}$$

- $\mathcal{D}_\in = X_1 \in \mathcal{D}_1, \ldots, X_k \in \mathcal{D}_k$,
- $\mathcal{D}'_\in = X_1 \in \mathcal{D}_1, \ldots, X_k \in \mathcal{D}_k$, più eventualmente nuovi domini non vuoti per nuove variabili (i domini delle variabili già presenti non si riducono)

Constraint Solving

Transformation Rules: esempio

$$\frac{\langle e_1 \neq e_2; \mathcal{D}_{\in} \rangle}{\langle X = e_1, X \neq e_2; \mathcal{D}_{\in}, X \; in \; \mathbb{Z} \rangle}$$

Viene introdotta una nuova variabile, il cui dominio iniziale è il più grande possibile (assumiamo di lavorare con numeri interi). L'introduzione di una nuova variabile rende necessario ricorrere al concetto di equisoddisfacibilità tra CSP.

Constraint Solving

Introduction Rules

In alcuni casi risulta necessario o utile introdurre nuovi vincoli (tipicamente implicati dai vincoli già presenti). Le regole hanno la forma:

$$\frac{\varphi = \langle \mathcal{C}; \mathcal{D}_{\in} \rangle}{\psi = \langle \mathcal{C}, C; \mathcal{D}_{\in} \rangle}$$

dove C è un nuovo constraint.

Consideriamo questa regola con domini sui reali.

$$\frac{\langle X^2 - 2XY + Y^2 < 5; X > 0, Y > 0\rangle}{\langle X^2 - 2XY + Y^2 < 5, X - Y < 3; X > 0, Y > 0\rangle}$$

Il nuovo vincolo non aggiunge informazione essendo implicato dal primo CSP. Tuttavia introdurre vincoli semplici, se pur ridondanti, può rendere più efficiente il processo di inferenza di un constraint solver.

- L'applicazione ripetuta delle regole dei tre tipi visti viene detta fase di propagazione di vincoli.
- Una *derivazione* è una sequenza di applicazioni delle regole di dimostrazione, previa opportuna rinomina delle variabili presenti nelle regole.
- Una derivazione finita è:
  - di *fallimento* se l'ultimo CSP è `fail`,
  - *stabile* se non è di fallimento e si raggiunge un CSP per cui nessuna regola è applicabile,
  - di *successo* se è stabile e l'ultimo CSP è in forma risolta (ovvero è consistente e da esso è immediato individuare una soluzione).

La fase di constraint propagation deve essere computazionalmente efficiente (P).

NODE CONSISTENCY

Un CSP $\mathcal{P}$ è *node consistent* se per ogni variabile X in $\mathcal{P}$, ogni vincolo unario relativo a X coincide con il dominio di X.

$$\langle X_1 \geq 0, X_2 \geq 0; X_1 \ in \ \mathbb{N}, X_2 \ in \ \mathbb{N} \rangle$$

è node consistent

$$\langle X_1 \geq 5, X_2 \geq 1; X_1 \ in \ \mathbb{N}, X_2 \ in \ \mathbb{N} \rangle$$

non lo è (è comunque consistente)

$$\langle X_1 \neq X_2, X_1 = 0, X_2 = 0; X_1 \ in \ 0..0, X_2 \ in \ 0..0 \rangle$$

è node consistent, ma non è consistente

Constraint Propagation

Node Consistency

La node consistency si ottiene applicando ripetutamente la seguente regola di *domain reduction*:

$$\frac{\langle C; X \ in \ D_X \rangle}{\langle C; X \ in \ D_X \cap C \rangle}$$

dove C è un vincolo unario sulla variabile X.

Complessità? $O(cd)$

La node consistency si ottiene applicando ripetutamente la seguente regola di *domain reduction*:

$$\frac{\langle C; X \ in \ D_X \rangle}{\langle C; X \ in \ D_X \cap C \rangle}$$

dove C è un vincolo unario sulla variabile X.

Complessità? $O(cd)$

Constraint Propagation

Node Consistency

La node consistency si ottiene applicando ripetutamente la seguente regola di *domain reduction*:

$$\frac{\langle C; X \ in \ D_X \rangle}{\langle C; X \ in \ D_X \cap C \rangle}$$

dove C è un vincolo unario sulla variabile X.

Complessità? $O(cd)$

ARC CONSISTENCY

Un constraint binario C sulle variabili X ed Y aventi rispettivi domini D_X e D_Y è *arc consistent* se:

1. $(\forall a \in D_X)(\exists b \in D_Y)\,((a,b) \in C)$, e
2. $(\forall b \in D_Y)(\exists a \in D_X)\,((a,b) \in C)$.

Un CSP $\langle \mathcal{C}; \mathcal{D}_{\in} \rangle$ è arc consistent se lo sono tutti i constraint binari presenti in $\mathcal{C}$.

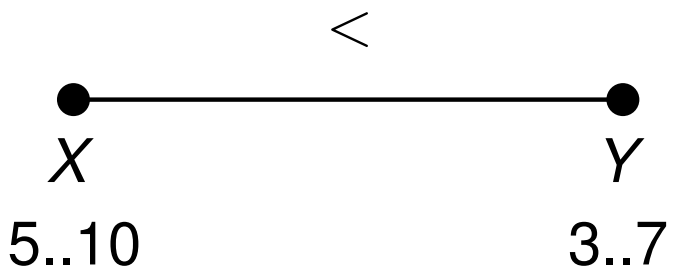
Le nozioni di arco e nodo vanno fatte risalire al grafo che possiamo costruire mettendo come nodi le variabili di un CSP (etichettati dai loro domini) e come archi i vincoli binari tra queste.

CONSTRAINT PROPAGATION

ARC CONSISTENCY: ESEMPIO

$$\langle X < Y; X\ in\ 5..10, Y\ in\ 3..7 \rangle$$

si può rappresentare con il grafo:



Non è arc consistent ma è consistente.

$$\langle X \neq Y, Y \neq Z, X \neq Z; X\ in\ 0..1, Y\ in\ 0..1, Z\ in\ 0..1 \rangle$$

è arc consistent, ma non consistente.

L'arc consistency si ottiene applicando ripetutamente due regole di *domain reduction*):

$$(AC1) \qquad \frac{\langle C; X \ in \ D_X, Y \ in \ D_Y \rangle}{\langle C; X \ in \ D'_X, Y \ in \ D_Y \rangle}$$

$$(AC2) \qquad \frac{\langle C; X \ in \ D_X, Y \ in \ D_Y \rangle}{\langle C; X \ in \ D_X, Y \ in \ D'_Y \rangle}$$

dove C è un vincolo binario sulle variabili X ed Y, mentre:

- $D'_X = \{a \in D_X \ : \ (\exists b \in D_Y)((a, b) \in C)\}$
- $D'_Y = \{b \in D_Y \ : \ (\exists a \in D_X)((a, b) \in C)\}$

Complessità? $O(ncd^3)$

Constraint Propagation

Arc Consistency

L'arc consistency si ottiene applicando ripetutamente due regole di *domain reduction*):

$$(AC1) \qquad \frac{\langle C; X \ in \ D_X, Y \ in \ D_Y \rangle}{\langle C; X \ in \ D'_X, Y \ in \ D_Y \rangle}$$

$$(AC2) \qquad \frac{\langle C; X \ in \ D_X, Y \ in \ D_Y \rangle}{\langle C; X \ in \ D_X, Y \ in \ D'_Y \rangle}$$

dove C è un vincolo binario sulle variabili X ed Y, mentre:

- $D'_X = \{a \in D_X \ : \ (\exists b \in D_Y)((a, b) \in C)\}$
- $D'_Y = \{b \in D_Y \ : \ (\exists a \in D_X)((a, b) \in C)\}$

Complessità? $O(ncd^3)$

L'arc consistency si ottiene applicando ripetutamente due regole di *domain reduction*):

$$(AC1) \qquad \frac{\langle C; X \ in \ D_X, Y \ in \ D_Y \rangle}{\langle C; X \ in \ D'_X, Y \ in \ D_Y \rangle}$$

$$(AC2) \qquad \frac{\langle C; X \ in \ D_X, Y \ in \ D_Y \rangle}{\langle C; X \ in \ D_X, Y \ in \ D'_Y \rangle}$$

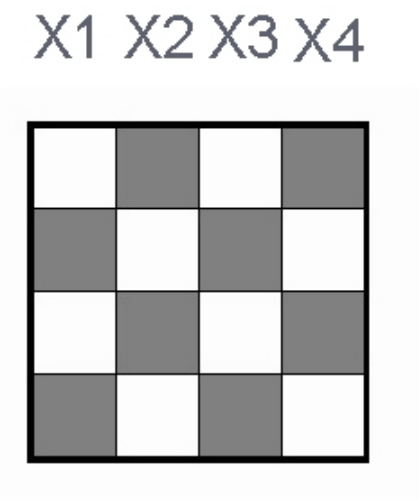
dove C è un vincolo binario sulle variabili X ed Y, mentre:

- $D'_X = \{a \in D_X \ : \ (\exists b \in D_Y)((a, b) \in C)\}$
- $D'_Y = \{b \in D_Y \ : \ (\exists a \in D_X)((a, b) \in C)\}$

Complessità? $O(ncd^3)$

Constraint Propagation

Arc Consistency, $X_1 = 1$



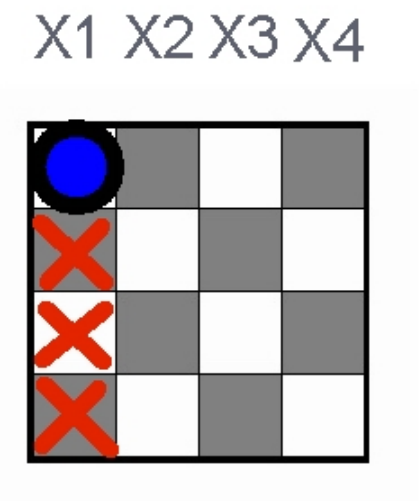
$$X_1, \ldots, X_4$$

$$D(X_i) = \{1, \ldots, 4\} \text{ per } i = 1..4$$

$$X_i \neq X_j, |X_i - X_j| \neq (j - i) \text{ per } i < j$$

Constraint Propagation

Arc Consistency, $X_1 = 1$



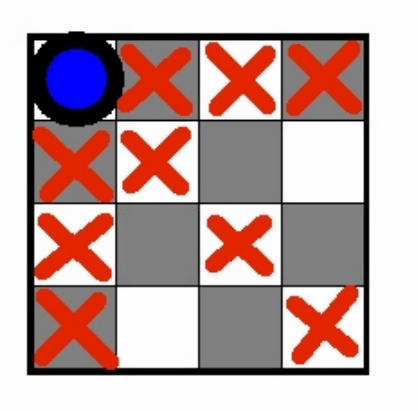
$$X_1, \ldots, X_4$$

$$D(X_1) = \{1\}, D(X_i) = \{1, \ldots, 4\} \text{ per } i = 2..4$$

$$X_i \neq X_j, |X_i - X_j| \neq (j - i) \text{ per } i < j$$

Constraint Propagation

Arc Consistency, $X_1 = 1$



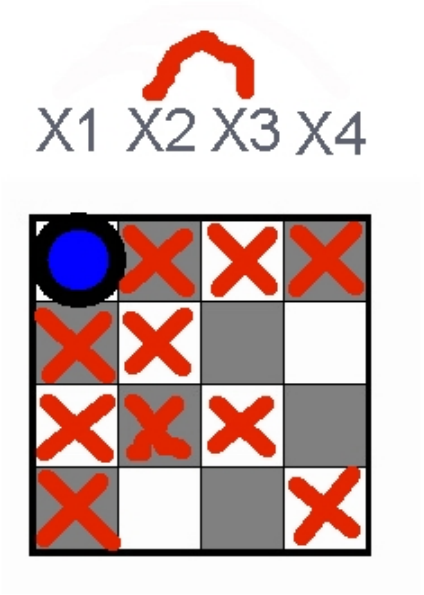
$$X_1, \ldots, X_4$$

$$D(X_1) = \{1\}, D(X_2) = \{3, 4\}, D(X_3) = \{2, 4\}, D(X_4) = \{2, 3\}$$

$$X_i \neq X_j, |X_i - X_j| \neq (j - i) \text{ per } i < j$$

Constraint Propagation

Arc Consistency, $X_1 = 1$



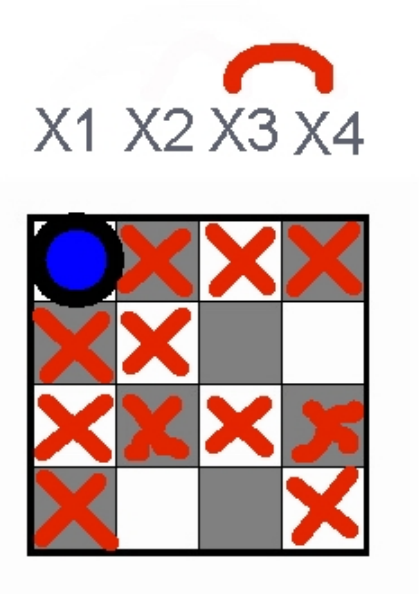
$$X_1, \ldots, X_4$$

$$D(X_1) = \{1\}, D(X_2) = \{4\}, D(X_3) = \{2, 4\}, D(X_4) = \{2, 3\}$$

$$X_i \neq X_j, |X_i - X_j| \neq (j - i) \text{ per } i < j$$

Constraint Propagation

Arc Consistency, $X_1 = 1$



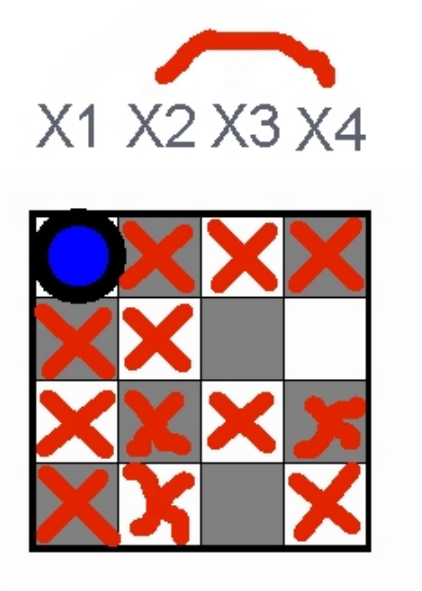
$$X_1, \ldots, X_4$$

$$D(X_1) = \{1\}, D(X_2) = \{4\}, D(X_3) = \{2, 4\}, D(X_4) = \{2\}$$

$$X_i \neq X_j, |X_i - X_j| \neq (j - i) \text{ per } i < j$$

Constraint Propagation

Arc Consistency, $X_1 = 1$



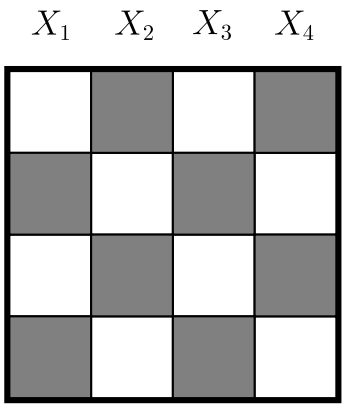
$$X_1, \ldots, X_4$$

$$D(X_1) = \{1\}, D(X_2) = \emptyset, D(X_3) = \{2, 4\}, D(X_4) = \{2, 3\}$$

$$X_i \neq X_j, |X_i - X_j| \neq (j - i) \text{ per } i < j$$

Constraint Propagation

Arc Consistency: $X_1 = 2$



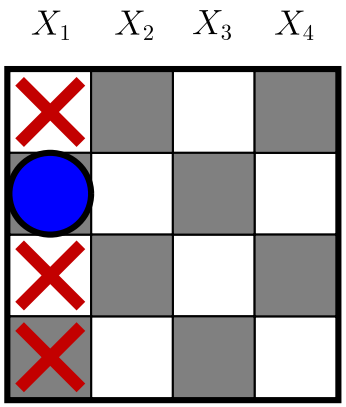
$$X_1, \ldots, X_4$$

$$D(X_i) = \{1, \ldots, 4\} \text{ per } i = 1..4$$

$$X_i \neq X_j, |X_i - X_j| \neq (j - i) \text{ per } i < j$$

Constraint Propagation

Arc Consistency: $X_1 = 2$



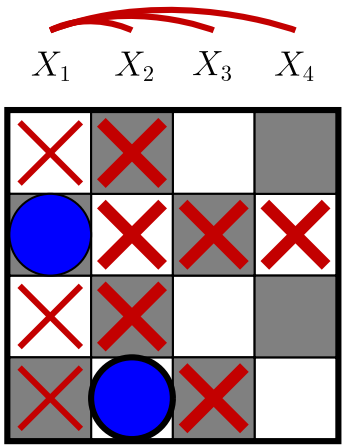
$$X_1, \ldots, X_4$$

$$D(X_1) = \{2\}, D(X_i) = \{1, \ldots, 4\} \text{ per } i = 2..4$$

$$X_i \neq X_j, |X_i - X_j| \neq (j - i) \text{ per } i < j$$

Constraint Propagation

Arc Consistency: $X_1 = 2$



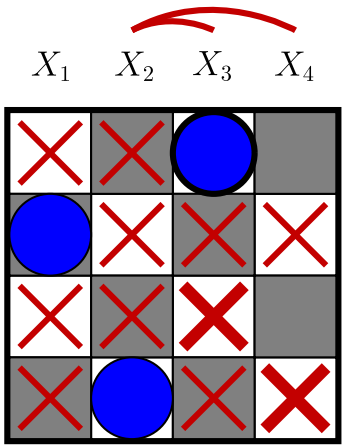
$$X_1, \ldots, X_4$$

$$D(X_1) = \{2\}, D(X_2) = \{4\}, D(X_3) = \{1, 3\}, D(X_4) = \{1, 3, 4\}$$

$$X_i \neq X_j, |X_i - X_j| \neq (j - i) \text{ per } i < j$$

Constraint Propagation

Arc Consistency: $X_1 = 2$



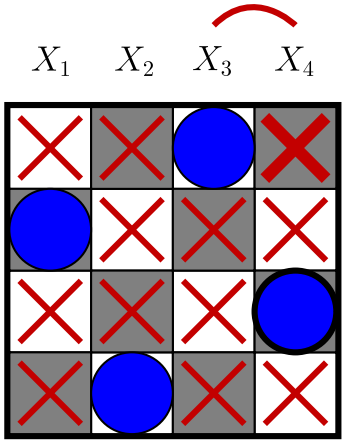
$$X_1, \ldots, X_4$$

$$D(X_1) = \{2\}, D(X_2) = \{4\}, D(X_3) = \{1\}, D(X_4) = \{3, 4\}$$

$$X_i \neq X_j, |X_i - X_j| \neq (j - i) \text{ per } i < j$$

Constraint Propagation

Arc Consistency: $X_1 = 2$



$$X_1, \ldots, X_4$$

$$D(X_1) = \{2\}, D(X_2) = \{4\}, D(X_3) = \{1\}, D(X_4) = \{3\}$$

$$X_i \neq X_j, |X_i - X_j| \neq (j - i) \text{ per } i < j$$

Abbiamo visto

- v variabili, c vincoli, d dominio massimo.
- Node Consistency $O(cd)$
- Arc Consistency $O(vcd^3)$

BOUNDS CONSISTENCY

Un constraint binario C sulle variabili X ed Y aventi rispettivi domini D_X e D_Y è *bounds consistent* se:

1. $(\exists b \in \min(D_Y)..\max(D_Y))\,((\min(D_X), b) \in C)$, e
   $(\exists b \in \min(D_Y)..\max(D_Y))\,((\max(D_X), b) \in C)$,
2. $(\exists a \in \min(D_X)..\max(D_X))\,((a, \min(D_Y)) \in C)$, e
   $(\exists a \in \min(D_X)..\max(D_X))\,((a, \max(D_Y)) \in C)$,

Un CSP $\langle \mathcal{C}; \mathcal{D}_{\in} \rangle$ è bounds consistent se lo sono tutti i constraint binari presenti in $\mathcal{C}$.

Il seguente CSP è consistent, bounds consistent, ma non arc consistent:

$$\langle 2X = Y; X\ in\ \{0, 1, 2\}, Y\ in\ \{0, 4\}\rangle$$

Applicando (AC1) si otterrebbe infatti:

$$\langle 2X = Y; X\ in\ \{0, 2\}, Y\ in\ \{0, 4\}\rangle$$

Le due regole per la bounds consistency sono:

$$(BC1) \qquad \frac{\langle C; X \ in \ D_X, Y \ in \ D_Y \rangle}{\langle C; X \ in \ D'_X, Y \ in \ D_Y \rangle}$$

$$(BC2) \qquad \frac{\langle C; X \ in \ D_X, Y \ in \ D_Y \rangle}{\langle C; X \ in \ D_X, Y \ in \ D'_Y \rangle}$$

- $D'_X = D_X \cap (\min(\{a \in D_X : (\exists b \in \min(D_Y)..\max(D_Y))((a,b) \in C)\})..\max(\{a \in D_X : (\exists b \in \min(D_Y)..\max(D_Y))((a,b) \in C)\}))$
- $D'_Y = D_Y \cap (\min(\{b \in D_Y : (\exists a \in \min(D_X)..\max(D_X))((a,b) \in C)\})..\max(\{b \in D_Y : (\exists a \in \min(D_X)..\max(D_X))((a,b) \in C)\}))$

Le regole per BC sembrano complesse. In realtà si possono implementare in tempo pressocché costante.

$$\frac{\langle X < Y; X\ in\ 5..20, Y\ in\ 1..10\rangle}{\langle X < Y; X\ in\ 5..9, Y\ in\ 1..10\rangle}$$

$$\frac{\langle X < Y; X\ in\ 5..9, Y\ in\ 1..10\rangle}{\langle X < Y; X\ in\ 5..9, Y\ in\ 6..9\rangle}$$

Guardo solo i bounds dei rispettivi domini.

Complessità? $O(vcd)$

Constraint Propagation

Bounds Consistency: esempio

Le regole per BC sembrano complesse. In realtà si possono implementare in tempo pressocché costante.

$$\frac{\langle X < Y; X\ in\ 5..20, Y\ in\ 1..10\rangle}{\langle X < Y; X\ in\ 5..9, Y\ in\ 1..10\rangle}$$

$$\frac{\langle X < Y; X\ in\ 5..9, Y\ in\ 1..10\rangle}{\langle X < Y; X\ in\ 5..9, Y\ in\ 6..9\rangle}$$

Guardo solo i bounds dei rispettivi domini.

Complessità? $O(vcd)$

Constraint Propagation

Bounds Consistency: esempio

Le regole per BC sembrano complesse. In realtà si possono implementare in tempo pressocché costante.

$$\frac{\langle X < Y; X\ in\ 5..20, Y\ in\ 1..10\rangle}{\langle X < Y; X\ in\ 5..9, Y\ in\ 1..10\rangle}$$

$$\frac{\langle X < Y; X\ in\ 5..9, Y\ in\ 1..10\rangle}{\langle X < Y; X\ in\ 5..9, Y\ in\ 6..9\rangle}$$

Guardo solo i bounds dei rispettivi domini.

Complessità? $O(vcd)$

Constraint Propagation

Directional arc/bounds consistency

- La richiesta e' di solo uno dei due punti
- C'e' implicito un ordine tra le variabili
- Vengono di fatto implementate nelle procedure di punto fisso, raddoppiando i vincoli iniziali ottenendo procedure piu' efficienti in pratica.

Si tratta di una generalizzazione dell'arc consistency a vincoli non binari.

HAC

Un vincolo n-ario C sulle variabili $X_1, \ldots, X_n$ è *hyper arc consistent* se per ogni $i = 1, \ldots, n$ vale che:

- $(\forall a_i \in D_i)(\exists a_1 \in D_1) \cdots (\exists a_{i-1} \in D_{i-1})(\exists a_{i+1} \in D_{i+1}) \cdots (\exists a_n \in D_n)(\langle a_1, \ldots a_n \rangle \in C)$

Un CSP è *hyper arc consistent* se lo sono tutti i suoi vincoli.

CONSTRAINT PROPAGATION

HYPER ARC CONSISTENCY

$$\langle 2X + 3Y < Z; X \ in \ 1..10, Y \ in \ 1..10, Z \ in \ 1..10 \rangle$$

Qualunque coppia di valori per X ed Y garantisce almeno la somma 5. Dunque, essendoci 1 in D_Z il CSP non è hyper arc consistent. Anche il 3 in D_Y è un valore da togliere. Infatti, con tale valore l'espressione a sinistra arriva a 9, a cui va sommato almeno 2 come contributo di X. Ció supera il valore massimo a sinistra (9).
Il CSP equivalente e hyper arc consistent è:

$$\langle 2X + 3Y < Z; X \ in \ 1..3, Y \ in \ 1..2, Z \ in \ 6..10 \rangle$$

Constraint Propagation

Hyper arc consistency

Per ottenere questa proprietà vanno applicate le regole ($HACi$) di seguito descritte, ove $i = 1, \ldots, n$:

$$\frac{\langle C; X_1 \ in \ D_1, \ldots, X_{i-1} \ in \ D_{i-1}, X_i \ in \ D_i, X_{i+1} \ in \ D_{i+1}, \ldots, X_n \ in \ D_n \rangle}{\langle C; X_1 \ in \ D_1, \ldots, X_{i-1} \ in \ D_{i-1}, X_i \ in \ D'_i, X_{i+1} \ in \ D_{i+1}, \ldots, X_n \ in \ D_n \rangle}$$

ove C è un vincolo su $a_1, \ldots, a_n$ e

$$\begin{aligned} D'_i \quad = \quad & \{a_i \in D_i \ : \ (\exists a_1 \in D_1) \cdots (\exists a_{i-1} \in D_{i-1}) \\ & (\exists a_{i+1} \in D_{i+1}) \cdots (\exists a_n \in D_n)(\langle a_1, \ldots \ a_n \rangle \in C)\} \end{aligned}$$

Per ogni vincolo n-ario vi saranno dunque n di queste regole proiettive.

Complessità? $O(vcnd^{n+1})$

Constraint Propagation

Hyper arc consistency

Per ottenere questa proprietà vanno applicate le regole ($HACi$) di seguito descritte, ove $i = 1, \ldots, n$:

$$\frac{\langle C; X_1 \ in \ D_1, \ldots, X_{i-1} \ in \ D_{i-1}, X_i \ in \ D_i, X_{i+1} \ in \ D_{i+1}, \ldots, X_n \ in \ D_n \rangle}{\langle C; X_1 \ in \ D_1, \ldots, X_{i-1} \ in \ D_{i-1}, X_i \ in \ D'_i, X_{i+1} \ in \ D_{i+1}, \ldots, X_n \ in \ D_n \rangle}$$

ove C è un vincolo su $a_1, \ldots, a_n$ e

$$\begin{aligned} D'_i \quad = \quad & \{a_i \in D_i \ : \ (\exists a_1 \in D_1) \cdots (\exists a_{i-1} \in D_{i-1}) \\ & (\exists a_{i+1} \in D_{i+1}) \cdots (\exists a_n \in D_n)(\langle a_1, \ldots \ a_n \rangle \in C)\} \end{aligned}$$

Per ogni vincolo n-ario vi saranno dunque n di queste regole proiettive.

Complessità? $O(vcnd^{n+1})$

Constraint Propagation

Hyper arc consistency

Per ottenere questa proprietà vanno applicate le regole ($HACi$) di seguito descritte, ove $i = 1, \ldots, n$:

$$\frac{\langle C; X_1 \ in \ D_1, \ldots, X_{i-1} \ in \ D_{i-1}, X_i \ in \ D_i, X_{i+1} \ in \ D_{i+1}, \ldots, X_n \ in \ D_n \rangle}{\langle C; X_1 \ in \ D_1, \ldots, X_{i-1} \ in \ D_{i-1}, X_i \ in \ D'_i, X_{i+1} \ in \ D_{i+1}, \ldots, X_n \ in \ D_n \rangle}$$

ove C è un vincolo su $a_1, \ldots, a_n$ e

$$\begin{aligned} D'_i \quad = \quad & \{a_i \in D_i \ : \ (\exists a_1 \in D_1) \cdots (\exists a_{i-1} \in D_{i-1}) \\ & (\exists a_{i+1} \in D_{i+1}) \cdots (\exists a_n \in D_n)(\langle a_1, \ldots \ a_n \rangle \in C)\} \end{aligned}$$

Per ogni vincolo n-ario vi saranno dunque n di queste regole proiettive.

Complessità? $O(vcnd^{n+1})$

Constraint Propagation

Riassunto e altre regole

v vars, c vincoli, d max dom, n arità vincoli.

- Node Consistency $O(cd)$
- Arc Consistency $O(vcd^3)$
- Bounds Consistency $O(vcd)$
- Hyper Arc Consistency $O(vcnd^{n+1})$
- Hyper Bounds Consistency $O(vcdn^2)$
- Directional HAC, HBC
- Path Consistency
- k-consistency

Constraint Propagation

Riassunto e altre regole

v vars, c vincoli, d max dom, n arità vincoli.

- Node Consistency $O(cd)$
- Arc Consistency $O(vcd^3)$
- Bounds Consistency $O(vcd)$
- Hyper Arc Consistency $O(vcnd^{n+1})$
- Hyper Bounds Consistency $O(vcdn^2)$
- Directional HAC, HBC
- Path Consistency
- k-consistency

Constraint Propagation

Riassunto e altre regole

v vars, c vincoli, d max dom, n arità vincoli.

- Node Consistency $O(cd)$
- Arc Consistency $O(vcd^3)$
- Bounds Consistency $O(vcd)$
- Hyper Arc Consistency $O(vcnd^{n+1})$
- Hyper Bounds Consistency $O(vcdn^2)$
- Directional HAC, HBC
- Path Consistency
- k-consistency

Constraint Propagation

Riassunto e altre regole

v vars, c vincoli, d max dom, n arità vincoli.

- Node Consistency $O(cd)$
- Arc Consistency $O(vcd^3)$
- Bounds Consistency $O(vcd)$
- Hyper Arc Consistency $O(vcnd^{n+1})$
- Hyper Bounds Consistency $O(vcdn^2)$
- Directional HAC, HBC
- Path Consistency
- k-consistency

Constraint Propagation

Riassunto e altre regole

v vars, c vincoli, d max dom, n arità vincoli.

- Node Consistency $O(cd)$
- Arc Consistency $O(vcd^3)$
- Bounds Consistency $O(vcd)$
- Hyper Arc Consistency $O(vcnd^{n+1})$
- Hyper Bounds Consistency $O(vcdn^2)$
- Directional HAC, HBC
- Path Consistency
- k-consistency

CONSTRAINT PROPAGATION

PATH CONSISTENCY

- Un CSP $\mathcal{P}$ è *normalizzato* se per ogni coppia non ordinata di variabili X, Y esiste al più un vincolo binario su X, Y in $\mathcal{P}$.
- Se $\mathcal{P}$ è *normalizzato* indichiamo con $C(X, Y)$ tale vincolo se esiste. Se non c'è, possiamo definire $C(X, Y) = D_X \times D_Y$.
- Un CSP $\mathcal{P}$ è *standard* se per ogni coppia non ordinata di variabili X, Y esiste esattamente un vincolo binario su X, Y ($C(X, Y)$) in $\mathcal{P}$.

Constraint Propagation

Path Consistency: esempio

$$\langle \underbrace{X+Y<5}_{C(X,Y)}, \underbrace{X+Y\leq Z}_{C(X,Y,Z)}, \underbrace{X<Z}_{C(X,Z)}, \underbrace{2X\neq 3Z}_{C(X,Z)}, \underbrace{X+Y\neq Z}_{C(X,Y,Z)}; \mathcal{D}_{\in}\rangle$$

- Non è normalizzato in quanto ci sono due vincoli per la coppia X, Z. Possiamo (se la sintassi lo permette) rimpiazzare due vincoli C_1 e C_2 con l'unico vincolo $C_1 \wedge C_2$.
- Se aggiungiamo il vincolo $D_Y \times D_Z$ (chiamiamolo `all`) è standard.

$$\langle \underbrace{X+Y<5}_{C(X,Y)}, \underbrace{(X<Z \wedge 2X\neq 3Z)}_{C(X,Z)}, \underbrace{\texttt{all}(Y,Z)}_{C(Y,Z)}, \underbrace{X+Y<Z}_{C(X,Y,Z)}; \mathcal{D}_{\in}\rangle$$

Si può dimostrare che per ogni CSP ne esiste uno standard equivalente. In questa fase di normalizzazione, si può perdere l'arc consistency (e si può pure accorgersi dell'eventuale inconsistenza).

$$\langle X + Y = 0, X - Y = 0; X\ in\ \{-1, 1\}, Y\ in\ \{-1, 1\}\rangle$$

è arc consistent ma non standard.

$$\langle (X + Y = 0 \wedge X - Y = 0); X\ in\ \{-1, 1\}, Y\ in\ \{-1, 1\}\rangle$$

è standard ma non arc consistent.

Constraint Propagation

Path Consistency

- Data R binaria $R^T = \{(b, a) : (a, b) \in R\}$
- Date R e S binarie:

$$RS = \{(a, b) : \exists c\,((a, c) \in R, (c, b) \in S)\}$$

Path consistency

Un CSP standard è *path consistent* se per ogni tripla di variabili X, Y, Z vale che

$$C(X, Z) \subseteq C(X, Y)C(Y, Z)$$

ovvero, se $(a, b) \in C(X, Z)$ esiste $c \in D_Y$ tale che $(a, c) \in C(X, Y)$ e $(c, b) \in C(Y, Z)$.

E' implicitamente richiesto anche:

$$C(X, Y) \subseteq C(X, Z)C(Z, Y) \text{ e } C(Y, Z) \subseteq C(Y, X)C(X, Z)$$

Constraint Propagation

Path Consistency: esempio

- Si consideri il solito:

$$\langle X \neq Y, Y \neq Z, X \neq Z; X\ in\ 0..1, Y\ in\ 0..1, Z\ in\ 0..1 \rangle$$

- Abbiamo che $C(X,Z) = C(X,Y) = C(Y,Z) = \{(0,1),(1,0)\}$.
- Osserviamo che $C(X,Y)C(Y,Z) = \{(0,0),(1,1)\}$
- e dunque

$$C(X,Z) = \{(0,1),(1,0)\} \not\subseteq \{(0,0),(1,1)\} = C(X,Y)C(Y,Z)$$

- ovvero non è path consistent.

Constraint Propagation

Path Consistency: proof rules

$$(PC1) \qquad \frac{\langle C(X,Y), C(X,Z), C(Y,Z); \mathcal{D}_{\in}\rangle}{\langle C'(X,Y), C(X,Z), C(Y,Z); \mathcal{D}_{\in}\rangle}$$

$$(PC2) \qquad \frac{\langle C(X,Y), C(X,Z), C(Y,Z); \mathcal{D}_{\in}\rangle}{\langle C(X,Y), C'(X,Z), C(Y,Z); \mathcal{D}_{\in}\rangle}$$

$$(PC3) \qquad \frac{\langle C(X,Y), C(X,Z), C(Y,Z); \mathcal{D}_{\in}\rangle}{\langle C'(X,Y), C(X,Z), C'(Y,Z); \mathcal{D}_{\in}\rangle}$$

ove:

- $C'(X,Y) = C(X,Y) \cap (C(X,Z)C(Y,Z)^T)$
- $C'(X,Z) = C(X,Z) \cap (C(X,Y)C(Y,Z))$
- $C'(Y,Z) = C(Y,Z) \cap (C(X,Y)^T C(X,Z))$

Complessità? $O(vd\, v(v-1)(v-2)\, d^4) = O(v^4 d^5)$

Constraint Propagation

Path Consistency: proof rules

$$(PC1) \qquad \frac{\langle C(X,Y), C(X,Z), C(Y,Z); \mathcal{D}_{\in}\rangle}{\langle C'(X,Y), C(X,Z), C(Y,Z); \mathcal{D}_{\in}\rangle}$$

$$(PC2) \qquad \frac{\langle C(X,Y), C(X,Z), C(Y,Z); \mathcal{D}_{\in}\rangle}{\langle C(X,Y), C'(X,Z), C(Y,Z); \mathcal{D}_{\in}\rangle}$$

$$(PC3) \qquad \frac{\langle C(X,Y), C(X,Z), C(Y,Z); \mathcal{D}_{\in}\rangle}{\langle C'(X,Y), C(X,Z), C'(Y,Z); \mathcal{D}_{\in}\rangle}$$

ove:

- $C'(X,Y) = C(X,Y) \cap (C(X,Z)C(Y,Z)^T)$
- $C'(X,Z) = C(X,Z) \cap (C(X,Y)C(Y,Z))$
- $C'(Y,Z) = C(Y,Z) \cap (C(X,Y)^T C(X,Z))$

Complessità? $O(vd\, v(v-1)(v-2)\, d^4) = O(v^4 d^5)$

Constraint Propagation

Path Consistency: proof rules

$$(PC1) \qquad \frac{\langle C(X,Y), C(X,Z), C(Y,Z); \mathcal{D}_{\in}\rangle}{\langle C'(X,Y), C(X,Z), C(Y,Z); \mathcal{D}_{\in}\rangle}$$

$$(PC2) \qquad \frac{\langle C(X,Y), C(X,Z), C(Y,Z); \mathcal{D}_{\in}\rangle}{\langle C(X,Y), C'(X,Z), C(Y,Z); \mathcal{D}_{\in}\rangle}$$

$$(PC3) \qquad \frac{\langle C(X,Y), C(X,Z), C(Y,Z); \mathcal{D}_{\in}\rangle}{\langle C'(X,Y), C(X,Z), C'(Y,Z); \mathcal{D}_{\in}\rangle}$$

ove:

- $C'(X,Y) = C(X,Y) \cap (C(X,Z)C(Y,Z)^T)$
- $C'(X,Z) = C(X,Z) \cap (C(X,Y)C(Y,Z))$
- $C'(Y,Z) = C(Y,Z) \cap (C(X,Y)^T C(X,Z))$

Complessità? $O(vd\, v(v-1)(v-2)\, d^4) = O(v^4 d^5)$

Constraint Propagation

k-consistency

- E' una generalizzazione delle precedenti nozioni.
- Dato un CSP $\mathcal{P}$ sulle variabili $\mathcal{V}$.
- Un *assegnamento* $I = \{(X_1, d_1), \ldots, (X_k, d_k)\}$ (in breve $[X_1/d_1, \ldots, X_k/d_k]$) è una funzione di alcune (o tutte) variabili $X_1, \ldots, X_k$ di $\mathcal{V}$ su elementi dei rispettivi domini.
- Diremo che il dominio di I, $\mathrm{dom}(I) = \{X_1, \ldots, X_k\}$ ($|\mathrm{dom}(I)| = k$).
- Dato un vincolo C, se $FV(C) \subseteq \{X_1, \ldots, X_k\}$ allora diciamo che I soddisfa C se la restrizione di I alle variabili di C è una soluzione di C.
- Per un CSP $\mathcal{P}$, un *assegnamento* I con dominio $\{X_1, \ldots, X_k\}$, è k-consistente se soddisfa tutti i vincoli di $\mathcal{P}$ definiti su sottoinsiemi di $\{X_1, \ldots, X_k\}$.

Si consideri il CSP:

$$\langle X \neq Y, Y < Z; X\ in\ 0..1, Y \in in 0..1, \in Z\ in\ 0..1 \rangle$$

L'assegnamento $I = [X/0, Y/1]$ è 2-consistente (l'unico vincolo da considerarsi è il primo)
Osserviamo che non esiste nessuna estensione di I che sia soluzione del CSP (che però è consistente)

- Se $dom(I) \supseteq FV(\mathcal{P})$ e I è k-consistente, allora I è una soluzione di $\mathcal{P}$.
- Un CSP $\mathcal{P}$ è:
  - 1-consistente se è node consistent.
  - k-consistente ($k > 1$) se per ogni assegnamento I che sia $k-1$ consistente, e per ogni variabile $X \notin \text{dom}(I)$, esiste un valore in D_X tale che l'assegnamento risultante è k-consistente.

Constraint Propagation

k-consistency

- Per definizione, si ha che $\mathcal{P}$ è 1-consistente sse è node consistent.
- Supponiamo $\mathcal{P}$ sia 2-consistente. Allora per ogni vincolo binario $C(X, Y)$ e per ogni assegnamento di una sola delle due variabili, poniamo Y, esiste un valore nel dominio di X che rende vero C. Ma ciò equivale a dire che è arc-consistent.
- Supponiamo $\mathcal{P}$ sia 3-consistente. Consideriamo i vincoli $C(X, Z), C(X, Y), C(Y, Z)$. Per mostrare che $\mathcal{P}$ è path-consistent dobbiamo mostrare che $C(X, Z) \subseteq C(X, Y)C(Y, Z)$. Prendiamo un assegnamento I che sia 2-consistente per $\mathcal{P}$ e supponiamo che I contenga $X/a, Z/b$. Essendo 3-consistente, abbiamo che esiste un valore per Y che soddisfi $C(a, Y)$ e $C(Y, b)$.

Constraint Propagation

Vincoli globali

- In Constraint Programming vincoli che riguardano gruppi di variabili (in pratica, tutti i vincoli non unari nè binari) sono detti vincoli globali.
- Solitamente vincoli globali possono essere implementati come congiunzioni di vincoli binari, ma in tal caso la propagazione che si ottiene è spesso molto povera.
- Vincoli globali di uso comune sono studiati indipendentemente. Una buona propagazione di questi è un punto cruciale per la velocizzazione dei constraint solvers.
- Ne vedremo alcuni nel resto del corso, in particolare il vincolo di `alldifferent`.

Constraint Solving

- La ricerca delle soluzioni avviene alternando fasi di scelta non-deterministica a fasi di propagazione deterministica.
- Si sceglie (in base a qualche regola) una variabile e si riduce il suo dominio (scelta di un valore, dimezzamento del domino, etc.)
- Oppure si sceglie un vincolo e lo si *suddivide*
- Si applica propagazione e si continua.
- Se un dominio diventa vuoto, si ferma quel ramo.
- Se non ci sono piu' variabili, si fornisce la soluzione.
- Grandezza e forma dell'albero dipendono dai criteri di scelta.

Constraint Solving

Domain Splitting Rules

1. (domain) labeling:

$$\frac{X \in \{a_1, \ldots, a_k\}}{X \in \{a_1\} | \cdots | X \in \{a_k\}}$$

2. (domain) enumeration:

$$\frac{X \in \mathcal{D}}{X \in \{a\} | X \in \mathcal{D} \setminus \{a\}}$$

   ove $a \in \mathcal{D}$

3. (domain) bisection:

$$\frac{X \in \mathcal{D}}{X \in \min(\mathcal{D})..a | X \in b..\max(\mathcal{D})}$$

   ove $a, b \in \mathcal{D}$, e b è l'elemento immediatamente più grande di a in $\mathcal{D}$. Se $\mathcal{D}$ è un intervallo $x..y$ si prenderanno $a = \lfloor (x+y)/2 \rfloor$ e $b = a + 1$.

Constraint Solving

Constraint Splitting rules (esempi)

1. implicazione:

$$\frac{(C_1 \rightarrow C_2)}{\neg C_1 | C_2}$$

2. valore assoluto:

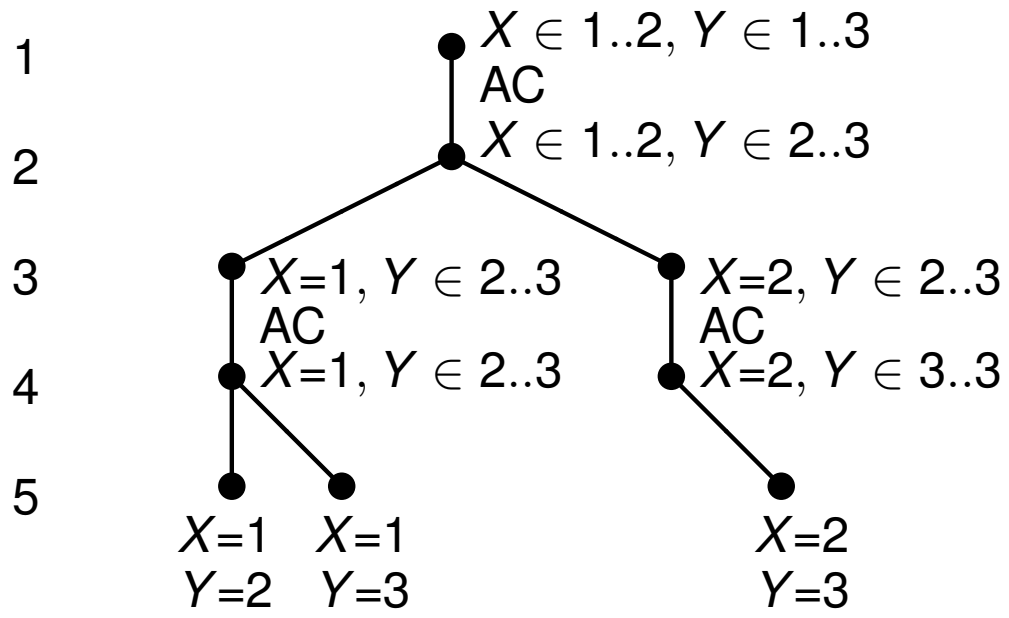
$$\frac{|e| = X}{X = e | X = -e}$$

3. diseguaglianza:

$$\frac{e_1 \neq e_2}{e_1 < e_2 | e_2 < e_1}$$

Constraint Solving

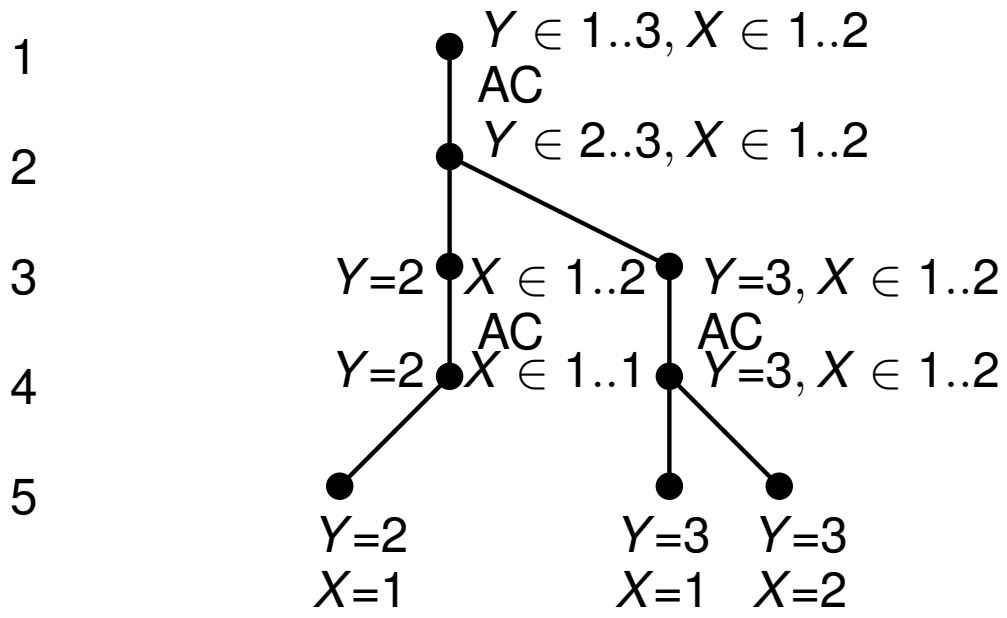
prop-labeling-tree



prop-labeling-tree per
$\mathcal{P} = \langle X < Y; X \in \{1,2\}, Y \in \{1,2,3\}\rangle$.

Constraint Solving

prop-labeling-tree



prop-labeling-tree per
$\mathcal{P} = \langle X < Y; X \in \{1, 2\}, Y \in \{1, 2, 3\}\rangle$.

Constraint Solving

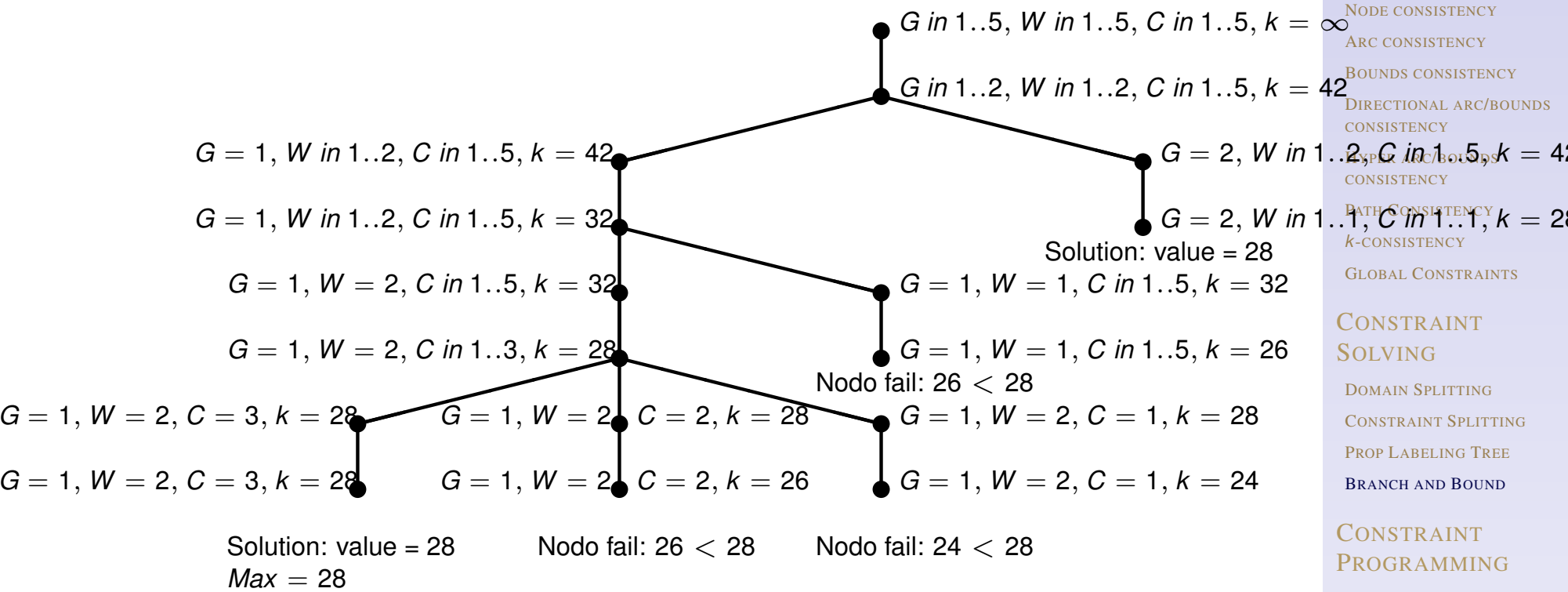
prop-labeling-tree per COP

- La costruzione e la visita del prop-labeling-tree viene effettuata dal predicato built-in `labeling`.
- Ogni constraint solver mette a disposizione molti parametri per il labeling
- Scelta della variabile (leftmost, ff, etc)
- Scelta del valore nel dominio (min, max, med, etc)
- Altri parametri (ricerca approssimata, timeout etc)
- Vedremo gli esempi del manuale di SICStus 4

Constraint Solving

Branch and Bound per COP

$\mathcal{C} = 17G + 10W + 4C < 50, f(W, G, C) = 10G + 6W + 2C$



Constraint Programming

- Si deve disporre di un linguaggio di programmazione con vincoli in diversi domini (interi, Booleani, reali, insiemi, etc)
- Provvisto di un risolutore di vincoli con una serie di parametri per guidare la fase di labeling
- Provvisto anche di costrutti standard dei linguaggi di programmazione
- Allo stato attuale, si trovano (anche liberi)
  - Linguaggi logici (Constraint Logic Programming—CLP)
  - Linguaggi ad-hoc per il modeling (OPL, Comet, Minizinc, etc)
  - Vincoli in linguaggi piu' tradizionali (p.es., Gecode in C++, JChoco in Java)
- In questo corso ci focalizzeremo sul CLP