Finite Set Constraints

Constraint Programming (Lecture 8)

Marco Kuhlmann, Guido Tack

Plan for today

- finite set variables
- propagators for constraints on finite sets
- encoding binary relations
- encoding finite trees

Finite-set constraints

Finite-set variables

- let Δ be a finite interval of integers
- finite domain variables take values in Δ
- finite set variables take values in $\wp(\Delta)$
- *domain*: fixed subset of Δ

Basic constraints

- assignments to FD variables are approximated using element constraints:

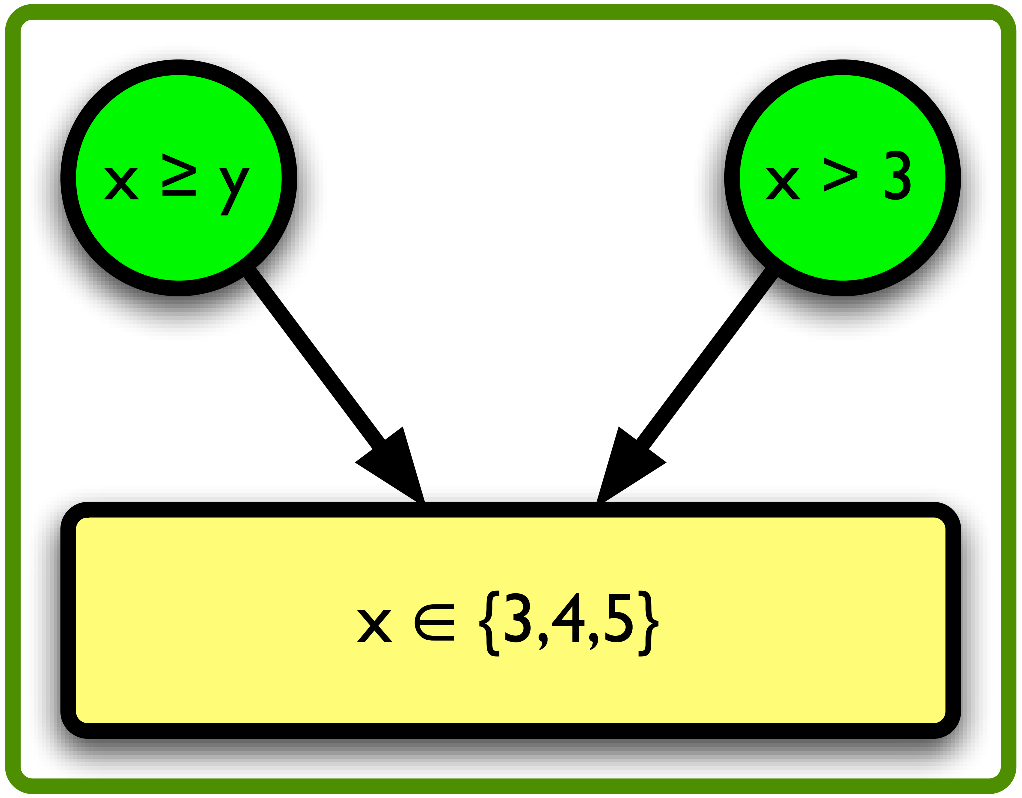
  $I \in D$

- assignments to FS variables are approximated using subset constraints:

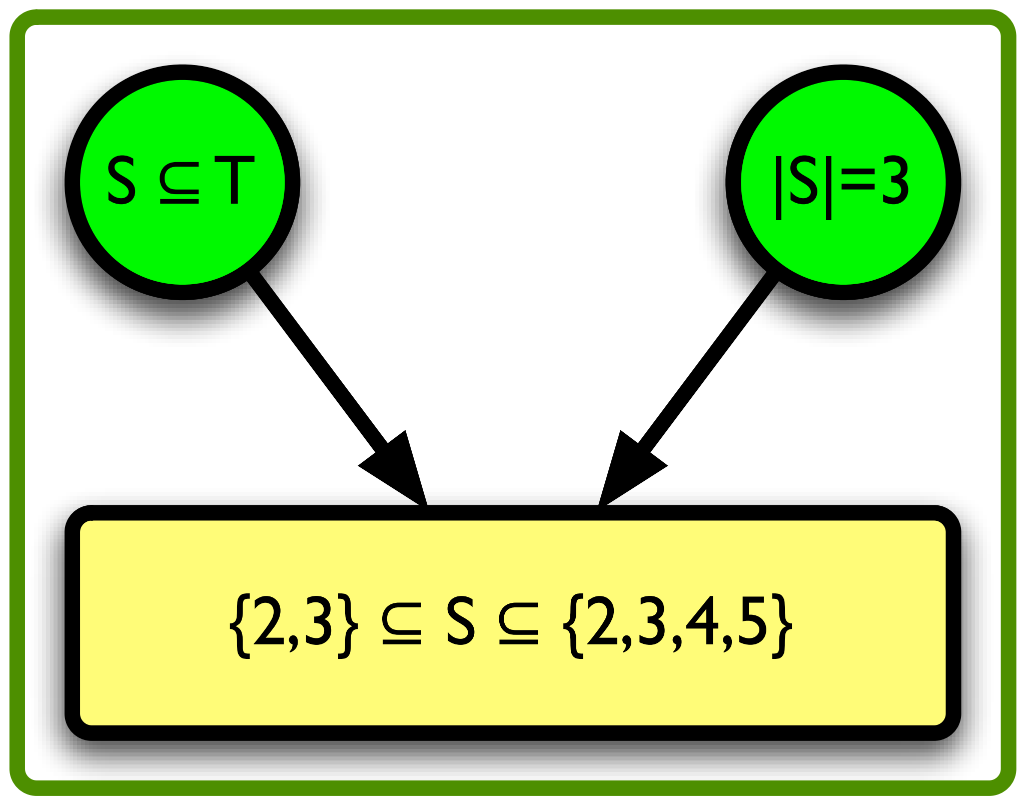
  $D \subseteq S \qquad S \subseteq D$

- together, these form the *basic constraints*

FD constraint store



FS constraint store



Non-basic constraints

just as with FD

- express non-basic constraints in terms of basic constraints, using sets of inference rules
- monotonicity: more specific premises yield more specific conclusions
- express inferences in terms of the currently entailed lower and upper bounds

$$\begin{aligned} \lfloor S \rfloor &= \cup\{\, D \mid D \subseteq S \,\} \\ \lceil S \rceil &= \cap\{\, D \mid S \subseteq D \,\} \end{aligned}$$

Subset constraint

$$\boldsymbol{S_1} \subseteq \boldsymbol{S_2} \quad \equiv \quad \lfloor S_1 \rfloor \subseteq S_2 \quad \wedge \quad S_1 \subseteq \lceil S_2 \rceil$$

basic constraint

Union and intersection

$$\boldsymbol{S_1} \cup \boldsymbol{S_2} = \boldsymbol{S_3} \quad \equiv \quad S_3 \subseteq \lceil S_1 \rceil \cup \lceil S_2 \rceil \ \wedge \ \lfloor S_1 \rfloor \cup \lfloor S_2 \rfloor \subseteq S_3$$

$$\boldsymbol{S_1} \cap \boldsymbol{S_2} = \boldsymbol{S_3} \quad \equiv \quad \lfloor S_1 \rfloor \cap \lfloor S_2 \rfloor \subseteq S_3 \ \wedge \ S_3 \subseteq \lceil S_1 \rceil \cap \lceil S_2 \rceil$$

Cardinality constraint

$|S| = I$

$$\begin{aligned}
\top &\implies |\lfloor S \rfloor| \leq I \\
\top &\implies I \leq |\lceil S \rceil| \\
n \leq I \ \wedge\ |\lceil S \rceil| = n &\implies \lceil S \rceil \subseteq S \\
I \leq n \ \wedge\ |\lfloor S \rfloor| = n &\implies S \subseteq \lfloor S \rfloor
\end{aligned}$$

Binary relations

The plan

- use FS constraints to encode binary relations on a fixed (and finite) universe
- express constraints on binary relations as constraints on FS variables

Encoding

- define the notion of the ‘relational image’

  $$Rx = \{\, y \in \mathcal{C} \mid Rxy \,\}$$

- understand binary relations as total functions from the carrier to subsets of the carrier

  $$f_R = \{\, x \mapsto Rx \mid x \in \mathcal{C} \,\}$$

- represent these functions as vectors on finite set variables

Union and intersection

$$\boldsymbol{R_1} \cup \boldsymbol{R_2} = \boldsymbol{R_3} \quad \equiv \quad R_3 = \langle R_1[1] \cup R_2[1], \ldots, R_1[n] \cup R_2[n] \rangle$$
$$\boldsymbol{R_1} \cap \boldsymbol{R_2} = \boldsymbol{R_3} \quad \equiv \quad R_3 = \langle R_1[1] \cap R_2[1], \ldots, R_1[n] \cap R_2[n] \rangle$$

How would you do composition?

Selection constraints

- generalisation of binary set operations
- participating elements are variable, too
- example: union with selection

$$S = \bigcup_{i \in S'} S_i$$

- propagation in all directions

Union with selection

$$\boldsymbol{S} = \cup\langle \boldsymbol{S_1}, \ldots, \boldsymbol{S_n} \rangle [\boldsymbol{S'}]$$

$$\begin{aligned}
&\implies S' \subseteq [1, n] \\
&\implies S \subseteq \cup\{\, \lceil S_j \rceil \mid j \in \lceil S' \rceil \,\} \\
&\implies \cup\{\, \lfloor S_j \rfloor \mid j \in \lfloor S' \rfloor \,\} \subseteq S \\
\lfloor S_j \rfloor \not\subseteq \lceil S \rceil &\implies j \notin S' \\
\lfloor S \rfloor \setminus \cup\{\, \lceil S_j \rceil \mid j \in \lceil S' \rceil \setminus \{k\} \,\} \neq 0 &\implies k \in S' \wedge \lfloor S \rfloor \setminus \cup\{\, \lceil S_j \rceil \mid j \in \lceil S' \rceil \setminus \{k\} \,\} \subseteq S_k
\end{aligned}$$

Composition

$$R_1 \circ R_2 \quad = \quad \{\, (x, z) \in \mathcal{C}^2 \mid \exists y \in \mathcal{C} :\; R_1 x y \wedge R_2 y z \,\}$$

$$\boldsymbol{R_1 \circ R_2 = R_3} \quad \equiv \quad R_3 = \langle \cup R_2[R_1[1]], \ldots, \cup R_2[R_1[n]] \rangle$$

Intersection with selection

- intersection with selection

$$S = \cap\langle S_1, \ldots, S_n\rangle[S']$$

- defines a new binary relation

$$\{\, (x, z) \in \mathcal{C}^2 \mid \forall y \in \mathcal{C} \colon\ R_1 x y \implies R_2 y z \,\}$$

- but what is it good for?

A weird relation

$$R_1 \bullet R_2 = \{\, (x, z) \in \mathcal{C}^2 \mid \forall y \in \mathcal{C} :\; R_1 x y \implies R_2 y z \,\}$$

- x sees z iff all R1-successors of x R2-see z
- x sees z if it does not have any R1-successors

Putting the weird relation to use

- require that the weird relation contains the identity relation
- the new relation is quite useful:

$$\forall x \in \mathcal{C} : x \in (R_1 \bullet R_2)x \implies R_1 \subseteq R_2^{-1}$$
$$\forall x \in \mathcal{C} : x \in (R_2 \bullet R_1)x \implies R_2 \subseteq R_1^{-1}$$

- the ‘converse constraint’ on relations

Converse

$$\boldsymbol{R_2 = R_1^{-1}}$$

$$i \in \mathcal{C} \implies i \in \cap R_1[R_2[i]]$$

$$i \in \mathcal{C} \implies i \in \cap R_2[R_1[i]]$$

Summing up

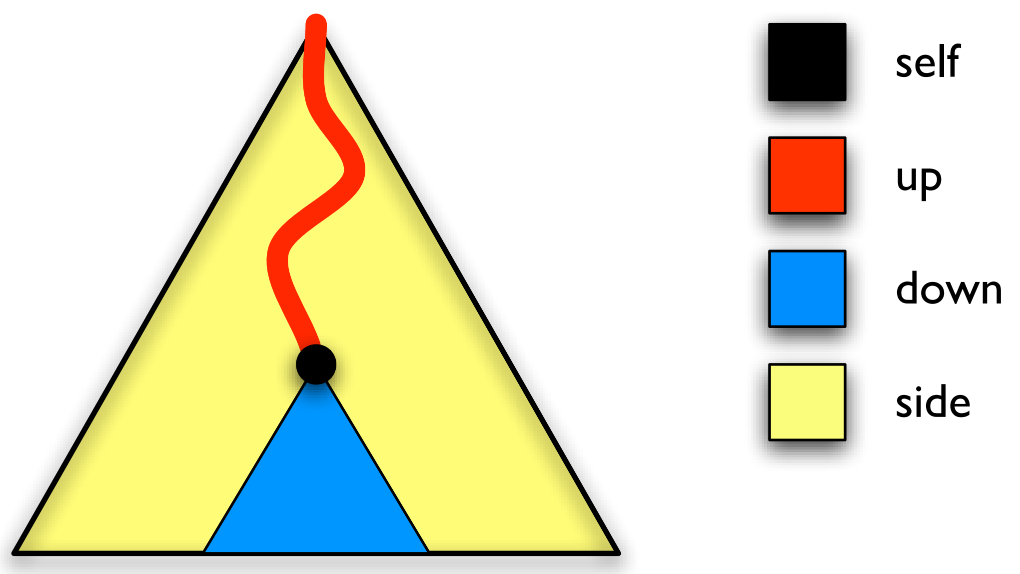
- used vectors of FS variables
  to encode binary relations
- constraints on binary relations
  can be stated as constraints on FS variables
- featured on next assignment

Solving tree descriptions

The plan

- use binary relations and set variables to encode various kinds of trees
- use FS constraints and constraints on binary relations to encode constraints on trees

Tree regions



Rooted tree constraint

$\mathbf{rootedTree}(\boldsymbol{v_1}, \ldots, \boldsymbol{v_n}, \mathcal{R})$

$$\Longrightarrow \quad \mathbf{succ} = \mathbf{pred}^{-1}$$

$$\Longrightarrow \quad \mathbf{succ}_* = \mathrm{Id} \cup \mathbf{succ}_+$$

$$\Longrightarrow \quad \mathbf{succ}_+ = \mathbf{succ} \circ \mathbf{succ}_*$$

$$\Longrightarrow \quad |\mathcal{R}| = 1$$

$$\Longrightarrow \quad \mathcal{V} = \mathcal{R} \uplus \mathbf{succ}(v_1) \uplus \ldots \mathbf{succ}(v_n)$$

$$v \in \mathcal{V} \quad \Longrightarrow \quad v \in \mathcal{R} \iff \mathbf{pred}(x) = \emptyset$$

Tree constraints

$$
\begin{aligned}
\mathbf{root}(v) &\equiv \mathbf{pred}(v) = \emptyset \\
\mathbf{leaf}(v) &\equiv \mathbf{succ}(v) = \emptyset \\
\mathbf{edge}(v_1, v_2) &\equiv v_2 \in \mathbf{succ}(v_1) \\
\mathbf{dominates}(v_1, v_2) &\equiv v_2 \in \mathbf{succ}_*(v_1)
\end{aligned}
$$

Fourth graded assignment

- implement a structure for constraints on binary relations
- implement solvers for rooted trees:
  - unordered trees
  - ordered trees