Department of Mathematics and Computer Science
University of Southern Denmark, Odense

September 20, 2020
Marco Chiarandini

DM877 - Constraint Programming

Exercises, Autumn 2020

Exercise 1 – Modelling

Show that CSP generalizes SAT formulating the following SAT problem as a CSP:

$$(x \vee y \vee \neg z) \wedge (\neg w \vee \neg z) \wedge (w \vee \neg y \vee z)$$

Solution

Variables: $\{w(x_1), x(x_2), y(x_3), z(x_4)\}$
Domains: $D(x_1) = D(x_2) = D(x_3) = D(x_4) = \{false, true\} = \{0, 1\}$
Constraints: $\mathcal{C} = \{C(x_2, x_3, x_4) \equiv x_2 \vee x_3 \vee \neg x_4; C(x_1, x_4) \equiv \neg x_1 \vee \neg x_4; C(x_1, x_2, x_4) \equiv x_1 \vee \neg x_3 \neg x_4\}$
The constraints can be also written as 0–1 linear inequalities of the form $a^T x \geq a_0$. Let $\neg x = \bar{x} = 1 - x$:

$$\begin{aligned} x_2 + x_3 + x_4 &\geq 1 \\ 1 - x_1 + 1 - x_4 &\geq 1 \\ x_1 + 1 - x_3 + x_4 &\geq 1 \end{aligned} \qquad \begin{aligned} x_2 + x_3 + x_4 &\geq 1 \\ x_1 + x_4 &\leq 1 \\ x_1 - x_3 + x_4 &\geq 0 \end{aligned}$$

Exercise 2 – Binary CSP

Show how an arbitrary (non-binary) CSP can be polynomially converted into an equivalent binary CSP.

Solution This can be done in two ways. (see fx [1])

For:

$$\begin{aligned} C_1 &: x_1 + x_2 + x_6 = 1 \\ C_2 &: x_1 - x_3 + x_4 = 1 \\ C_3 &: x_4 + x_5 - x_6 > 0 \\ C_4 &: x_2 + x_5 - x_6 = 0 \end{aligned}$$

See Figs. 1 and 2.

Exercise 3 – Domain-based tightenings

Given two CSP $\mathcal{P}$ and $\mathcal{P}'$, we write $\mathcal{P}' \preceq \mathcal{P}$ iff any instantiation I on $Y \subseteq X_{\mathcal{P}}$ locally inconsistent in $\mathcal{P}$ is locally inconsistent in $\mathcal{P}'$ as well.
Consider the following CSP: $\mathcal{P} = \langle X = \{x, y\}, \mathcal{DE} \equiv \{D(x) = \{1, 2, 3\}, D(y) = \{1, 2, 3\}\}, \mathcal{C}\rangle$.
Construct two domain tightenings $\mathcal{P}_1$ and $\mathcal{P}_2$ of $\mathcal{P}$ (a domain tightening is $\mathcal{P}'$ such that $X_{\mathcal{P}'} = X_{\mathcal{P}}$, $\mathcal{DE}' \subseteq \mathcal{DE}, \mathcal{C}_{\mathcal{P}'} = \mathcal{C}_{\mathcal{P}}$) for which the relation $\preceq$ defined above is in fact a partial order. (Assume $\mathcal{C}$ admits any combination of values as valid.)

Solution A domain-tightening always gives a partial ordering since it is isomorphic with the partial order $\subseteq$ on $\mathcal{DE}$.

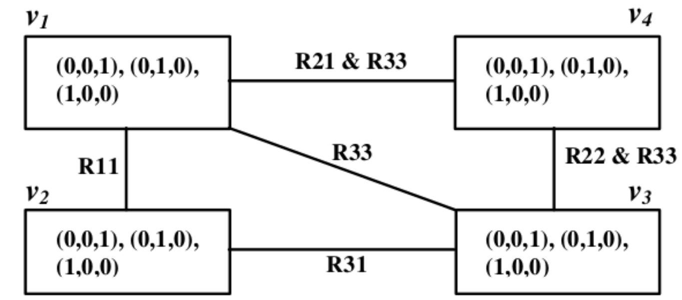


Figure 1: **Dual encoding**

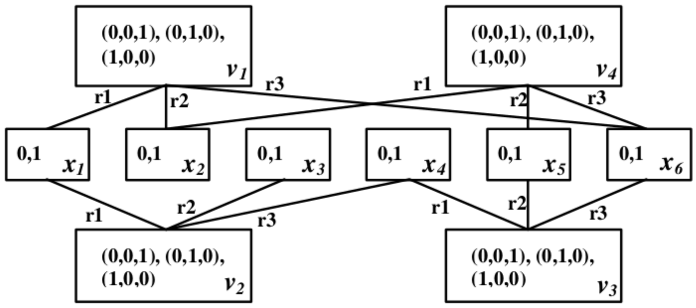


Figure 2: **Hidden variables encoding**

For example:

$$\mathcal{P}_1 = \langle X = \{x, y\}, \mathcal{DE} \equiv \{D(x) = \{1, 2\}, D(y) = \{2, 3\}\}$$
$$\mathcal{P}_2 = \langle X = \{x, y\}, \mathcal{DE} \equiv \{D(x) = \{2\}, D(y) = \{2, 3\}\}$$

Note that domain tightening is a well founded operation, that is, it has a last element (fixed point) because finitely many variables and finitely many values.

Exercise 4 – Local Consistency

Are the two following CSPs arc consistent:

- $\langle \{x = 1, y \in \{0, 1\}, z \in \{0, 1\}\}; x \wedge y = z; \rangle$

  Solution Yes.

- $\langle \{x \in \{0, 1\}, y \in \{0, 1\}, z = 1\}; x \wedge y = z; \rangle$

  Solution No: $x = 0$ and $y = 0$ have no support

Exercise 5 – Local Consistency

Consider the n-queens problem with $n \geq 3$ and its formulation as a binary CSP that uses the least variables (that is, n variables that indicate the position of the queens, say, on the columns). Is the initial status of this CSP problem arc consistent? If not, enforce arc consistency.

Solution the binary CSP that models the n-queens problem is

Variables: $x_1, \ldots, x_n$ with domain $[1, \ldots, n]$ whre x_i represents the row position of the queen placed in the ith column.

- $x_i \neq x_j$ for $i \in [1..n-1], j \in [i+1..n]$
- $x_i - x_j \neq i - j$ for $i \in [1..n-1], j \in [i+1..n]$
- $x_i - x_j \neq j - i$ for $i \in [1..n-1], j \in [i+1..n]$

It is arc consistent. Formally we need to analyse each constraint separately. Consider for instance the constraint $x_i - x_j \neq i - j$ with $1 \leq i < j \leq n$ and take $a \in [1..n]$.Then there exists $b \in [1..n]$ such that $a - b \neq i - j$: just take $b \in [1..n]$ that is different from $a - i + j$.
What about the non-binary formulation?

Exercise 6 – Propagation on paper

Consider an initial domain expression $\{x \in \{0,1,2,3\}, y \in \{0,1,2,3\}\}$ and two constraints $x < y$ and $y < x$. Apply the propagation algorithm Revise2001 from the lecture using pen and paper.

Solution

Not normalized. If we normalize it we discover the problem is inconsitent.
However to apply the Revise2001 we proceed by calculating

$$\texttt{Last}[x, v, y]$$

ie, the smallest support for (x, v) on y...
Note that bound arc consistency could be enforced faster for $>$ with the rules:

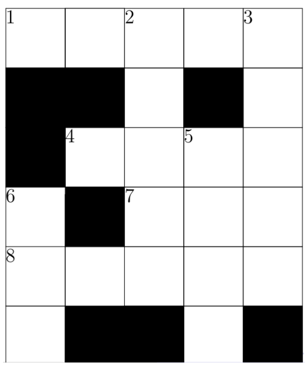
$$D(x) \leftarrow \{n \in D(x) | n < \max[D(y)]\}$$
$$D(y) \leftarrow \{n \in D(y) | n > \min[D(x)]\}$$

Exercise 7 – Directed Arc Consistency

A form of weaker arc consistency is directed arc consistency, which enforces consistency only in one direction. Decide if the following CSP $\langle x \in [2..10], y \in [3..7], x < y\rangle$ is directed arc consistent in the case of linear ordering $y \prec x$ and in the case $x \prec y$.

Exercise 8 – Crossword puzzle

Consider the crossword grid of the figure



and suppose we are to fill it with the words taken from the following list:

- HOSES, LASER, SAILS, SHEET, STEER,
- HEEL, HIKE, KEEL, KNOT, LINE,
- AFT, ALE, EEL, LEE, TIE.

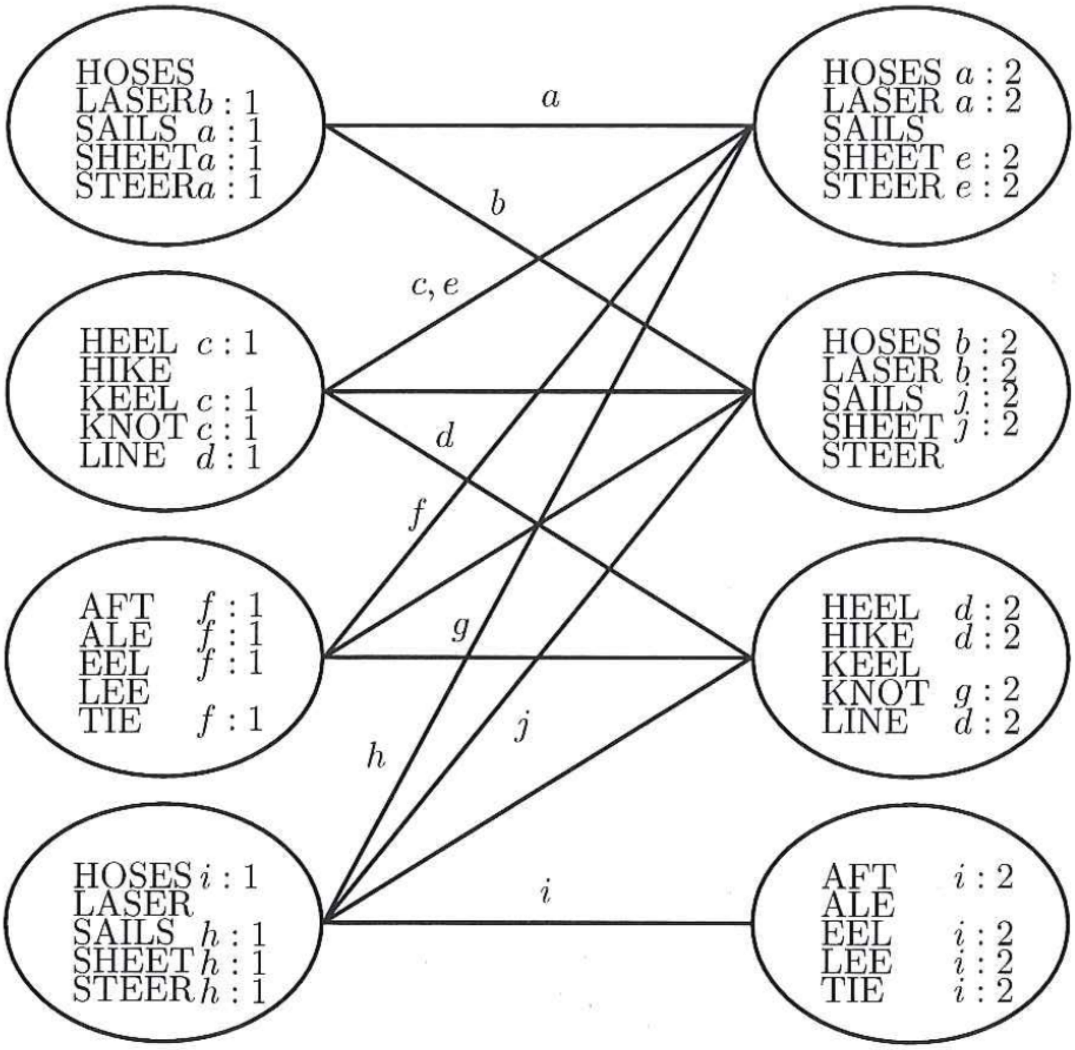


Figure 3:

Is the initial status of the formulated CSP arc consistent? If not, enforce arc consistency.

Solution

Domains: $X(x_1) = D(X_2) = \{HOSES, LASER, SAILS, SHEET, STEER\}$ etc.
Constraints: a constraint for each crossing. For positions 1 and 2:

$$C_{1,2} := \{(HOSES, SAILS), (HOSES, SHEET), (HOSES, STEER), (LASER, SAILS), (LASER, SHEET), (LASER, STEER)\}.$$

It is not arc consistent: no word in $D(x_2)$ begins with letter I, so for the values SAILS for the first variable no value for the second vairable exists such that the resulting pair satisfies the considered constraint.
Apply AC to the constraint network. See figure 3.

References

[1] Roman Barták. Theory and practice of constraint propagation. In *In Proceedings of the 3rd Workshop on Constraint Programming in Decision and Control*, pages 7–14, 2001.