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

Thursday 28th September, 2023
Marco Chiarandini

DM841 - Constraint Programming

Exercises, Autumn 2023

Exercise 1 – Modelling

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

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

Viceversa, show how to encode CSPs in SAT.

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: $\mathcal{C} = \{C(x_2, x_3, x_4) \equiv x_2 \vee x_3 \vee \neg x_4; C(x_1, x_4) \equiv \neg x_1 \vee \neg x_4; C(x_1, x_2, x_4) \equiv x_1 \vee \neg x_3 \neg x_4\}$
In Gecode:

```
clause(*this, BOT_OR, positives, negatives, 1);
```

See `examples/sat.cpp`.
The constraints can be also written as 0–1 linear inequalities of the form $a^T x \geq a_0$ Let $\neg x = \bar{x} = 1 - x$:

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

SAT Encodings of CSP:
There are several ways of translating finite-domain CSP into SAT problems. Three main encodings are the *sparse encoding*, the *order encoding* and the *log encoding*. For details you are referred to [2]. In Minizinc they are called *value*, *unary* and *binary*, respectively.

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 an example CSP with: $X = x_1, x_2, x_3, x_4, x_5, x_6$, $D(x_i) = \{0, 1\}$ $\forall i = 1..6$ and $\mathcal{C}$ made of:

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

the equivalent binary CSP is shown in Figs. 1 and 2.

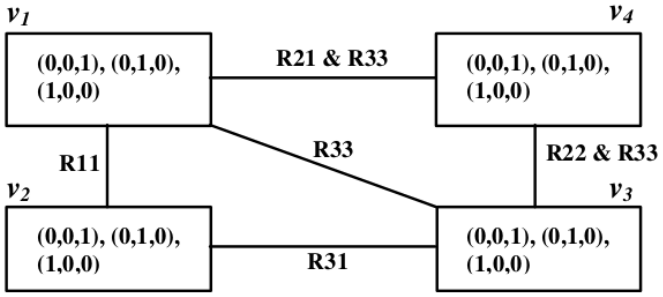


Figure 1: **Dual encoding**

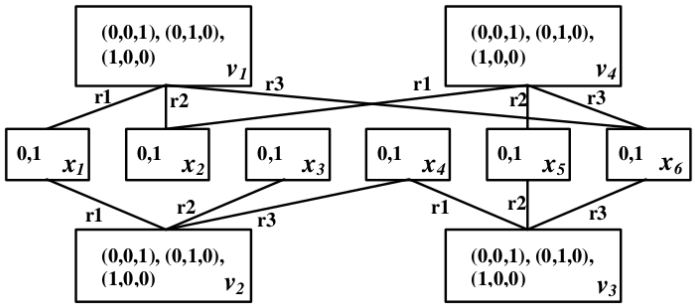


Figure 2: **Hidden variables encoding**

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 neither the relation $\mathcal{P} \preceq \mathcal{P}'$ holds nor $\mathcal{P}' \preceq \mathcal{P}$ (which shows that domain tightenings establish a partial order and not a total order amond the problems). Assume first that $\mathcal{C}$ admits any combination of values as valid. Then, consider the case in which $\mathcal{C} \equiv \{x \neq y\}$. What does it change?

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

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

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.
The addition of the constraint $x \neq y$ changes the possible tightenings that you might propose, for example, the problem:

$$\mathcal{P}_3 = \langle X = \{x, y\}, \mathcal{DE} \equiv \{D(x) = \{1\}, D(y) = \{1\}\}$$

would not be a tightening because it does not satisfy the requirement that any instantiation I on $Y \subseteq X_{\mathcal{P}}$ locally inconsistent in $\mathcal{P}$ is locally inconsistent in $\mathcal{P}_{\ni}$. For $Y = \{x, y\}$ this does not hold.

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. There exists a support for every value in every variable.

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

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

  What happens to your answers if the constraint $y \neq z$ is added to the set of constraints?

  Solution The first problem remains arc consistent even though it is inconsistent. The second problem can be discovered inconsistent after making the problem arc consistent with respect to one of the two constraints.

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:

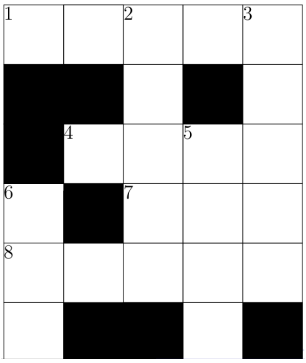
$$D(x) \leftarrow \{n \in D(x) | n < \max[D(y)]\}$$
$$D(y) \leftarrow \{n \in D(y) | n > \min[D(x)]\}$$

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.

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

Solution

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

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

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.

Exercise 9 – Mathematical Proof of Tournament Ranking

A common approach to decide a winner in sport competitions is to run a tournament, in which participants are grouped in one or more pools and have to play an equal number of matches against opponents from the same or other pools. The winner is the participant that at the end of the tournament occupies the first position in the standings list. (An alternative approach is to

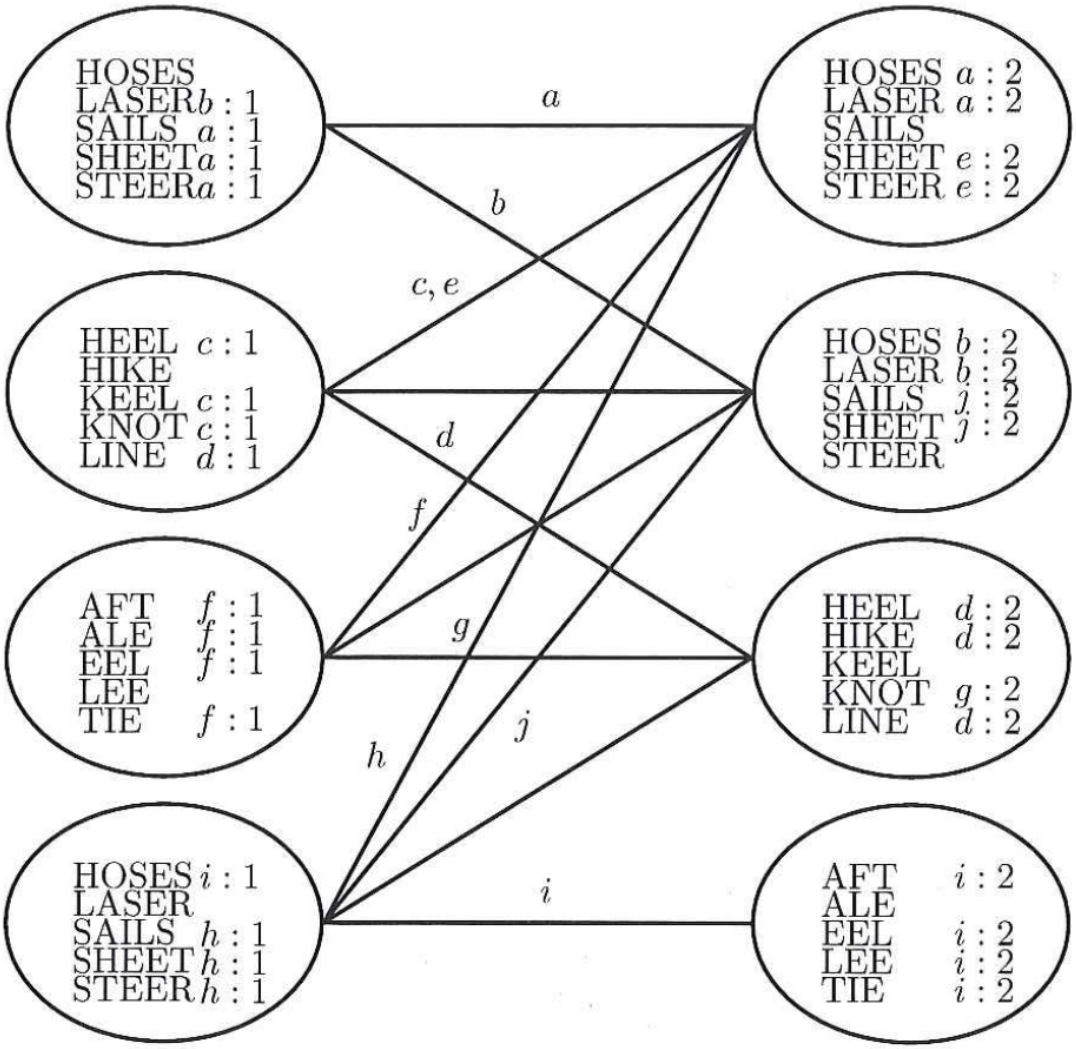

HOSES
LASER $b:1$
SAILS $a:1$
SHEET $a:1$
STEER $a:1$
HEEL $c:1$
HIKE
KEEL $c:1$
KNOT $c:1$
LINE $d:1$
AFT $f:1$
ALE $f:1$
EEL $f:1$
LEE
TIE $f:1$
HOSES $i:1$
LASER
SAILS $h:1$
SHEET $h:1$
STEER $h:1$
a
b
c, e
d
f
g
h
j
i
HOSES $a:2$
LASER $a:2$
SAILS
SHEET $e:2$
STEER $e:2$
HOSES $b:2$
LASER $b:2$
SAILS $j:2$
SHEET $j:2$
STEER
HEEL $d:2$
HIKE $d:2$
KEEL
KNOT $g:2$
LINE $d:2$
AFT $i:2$
ALE
EEL $i:2$
LEE $i:2$
TIE $i:2$

Figure 3:

run a single-elimination tournament where the loser of each match is immediately eliminated from winning the championship or the first prize in the event.)

Often the interest arises to prove mathematically whether after a number of rounds a partecipant can still become the champion given its current standing and all matches still to play in the tournament. For example, in football leagues the standing of a team is determined from the results by summing 3 points for each match won, 1 point for each match draw and 0 points for each match loss. Then, after a certain number of matches we may want to check whether a team can still be the winner of the league. Alternatively one may check which is the highest position that a team can reach.
Model this application as a CSP.
(Thanks to Anders Knudsen and Jan Christensen for providing the data)

Solution

Let $x_{hj} \in \{0,1,3\}$ and $x_{aj} \in \{0,1,3\}$ be the variables that indicate the points made on match j by the home team and the away team, respectively.
Let $y_i \in \mathbb{N}$ be the total points made by team i.

- the constraints ensuring that the right amount of points is distrubted at each match are:

  $$C(x_{hj}, x_{aj}) \equiv \{(0,3),(3,0),(1,1)\} \qquad \forall j$$

  alternatively, or redundant:

  $$x_{hj} + x_{aj} = z$$

  where $z \in \{2,3\}$ is an auxiliary variable.

- the linking between variables y and x:

  $$y_i = \sum_j (\delta_{ihj} x_{aj} + \delta_{iaj} x_{hj}) \; \forall i$$

  where δ_{ihj} and δ_{iaj} are binary parameters indicating whether the team i is the home team or the away team on game j.

- to find the position in the standings: we introduce an array of auxiliary variables w that will be the sorted version of y and an auxiliary vector z such that:

  $$y_i = w_{z_i}$$

  In gecode a global constraint takes care of this: sorted(y, w, z)

  Alternatively, we could use reification:

  $$y_i \geq y_l \Leftrightarrow b_{il} \forall l \neq i$$

  which in gecode is implemented by rel$(y_i, IRT_GE, y_l, b_{il})$.

For a given team i the objective will be to minimize z_i or to maximize $\sum_{il} b_{il}$.

A MIP model for solving the problem of finding the worst position of a team could be the following: Let T indexed by i and j be the set of teams. Let S indexed by s be the set of all matches, played and not yet played, and let $K = \{0,1,3\}$ be the set of possible points gained in a match by a team. Let $h(s)$ and $a(s)$ be the home and away team of match s, respectively, and let $A_s = \{(i,j) | i,j \in T, i = h(s), j = a(s)\}$ be the set of pairs of teams playing in s. For each match $s \in S$ we define a binary variable x_{sik} for any of the two teams in A_s and for any point gain $k \in K$. We denote by t the reference team and we assume that variables corresponding

to played matches are fixed. We assume that t looses all remaining matches and that ties are broken at disadvantage of t. We also define the set $B_i \subseteq S$ to indicate the matches played by team i.

$$\min \sum_i w_{it} \tag{1}$$

$$\text{s.t.} \sum_k x_{isk} = 1 \qquad \forall i \in A_s, \forall s \in S \tag{2}$$

$$x_{i,s,k} + x_{s,j,k} \leq 1 - z_s \qquad \forall k \in \{0,3\}, \forall (i,j) \in A_s, \forall s \in S \tag{3}$$

$$x_{s,i,1} + x_{s,j,1} = 2z_s \qquad \forall (i,j) \in A_s, \forall s \in S \tag{4}$$

$$y_i = \sum_j x_{si1} + 3\sum_j x_{si3} \qquad \forall i \in T, \forall s \in B_i \tag{5}$$

$$y_i \geq y_t - 3(n-1)w_{ti} \qquad \forall i \in T \tag{6}$$

$$y_i \geq 0 \qquad \forall i \in T \tag{7}$$

$$x_{sik} \in \{0,1\} \qquad \forall k \in K, i \in A_s, s \in S \tag{8}$$

$$w_{ij} \in \{0,1\} \qquad \forall i,j, i \neq j \tag{9}$$

Constraints 2 impose that exactly one of the three results is obtained by each team on each game. Constraints 3 and 4 take care of linking the variables at each game. The auxilary binary variables z_s are used to represent a draw. Constraints 5 calculate the total points of the teams and constraints 6 impose w_{ij} to be 1 if the team i has a better position in the ranking than the team j. The objective function 1 minimizes the number of teams that do worse than team t.

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.

[2] Van-Hau Nguyen. SAT encodings of finite-CSP domains: A survey. In *Proceedings of the Eighth International Symposium on Information and Communication Technology*, SoICT 2017, page 84–91, New York, NY, USA, 2017. Association for Computing Machinery.