Searching (Exercices)

1 Search Trees

1.1 BT

Recall that BT only checks covered constraints at each node of the search tree. A constraint c is *covered* by BT when the domain of every variable in $scp(c)$ is singleton. Considering that we use a non-binary branching scheme, count the number of nodes and the number of backtracks when BT is run, using a lexicographic order (*lexico* on both variables and values), on the following constraint network P:

- $vars(P) = \{x_1, x_2, x_3, x_4\}$, with $dom(x_i) = \{1, 2, 3\}, \forall i \in 1..4$
- $ctrs(P) = \{x_1 \neq x_2, x_1 \geq x_2 + x_3, 2 \times x_1 \leq x_2 \times x_4\}$

1.2 FC

Recall that FC only enforces arc consistency on almost covered constraints at each node of the search tree. A constraint c is *almost covered* by FC when all variables in $scp(c)$ except one are fixed (i.e., have a singleton domain). Let P be the CN such that:

- $vars(P) = \{x_1, x_2, x_3, x_4, x_5\}$, with
  - $dom(x_1) = dom(x_2) = dom(x_3) = \{0, 1, 2, 3\}$
  - $dom(x_4) = dom(x_5) = \{2, 3\}$
- $ctrs(P) = \{x_1 \neq x_2, x_1 \neq x_3, x_2 \neq x_3, x_2 \neq x_4, x_2 \neq x_5, x_3 \neq x_4, x_3 \neq x_5, x_4 \neq x_5\}$

1. Draw the constraint graph of P.
2. Considering that we use a non-binary branching scheme, draw the portion of the search tree built by FC with *lexico* as variable ordering heuristic and *lexico* as value ordering heuristic until you reach the positive decision $x_1 = 1$. How many nodes are there in this partial search tree?
3. Considering that we use a non-binary branching scheme, draw the search tree built by FC with *min-dom* as variable ordering heuristic and *lexico* as value ordering heuristic until you get a solution. How many nodes are there in this search tree?

1.3 Impact of Variable Ordering

Even when there is no pruning at all, the choice of the ordering of variables may impact the size of the search tree. Let x, y and z be three variables such that $dom(x) = \{1, 2, 3\}$, $dom(y) = \{1, 2\}$ and $dom(z) = \{1, 2, 3\}$.

1. Draw the search trees (using a non-binary branching scheme) for "generating and testing" all possible instantiations of these three variables when:
   - the order is x, y and z
   - the order is x, z and y
2. Compare the number of leaves and the total number of nodes.

1.4 Heuristic *ddeg*

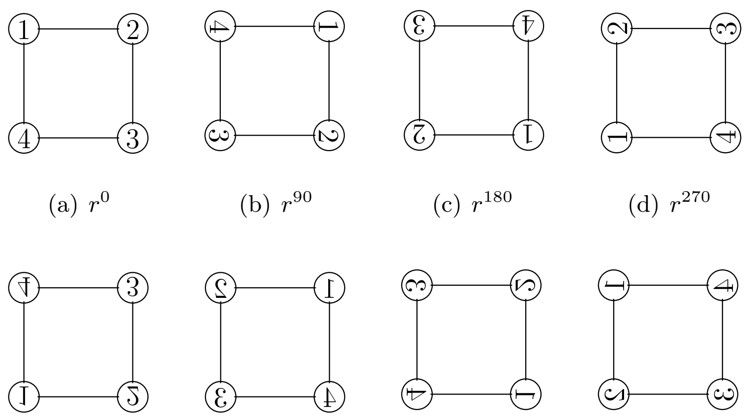
The heuristic *ddeg* considers the dynamic degree of a variable as its score. The dynamic degree of a variable x can be defined with respect to a backtrack search algorithm (such as BT, FC and MAC) as the number of constraints involving x as well as another variable $y \neq x$ that is not explicitly assigned by the search algorithm.

1. Compute the respective order given by *deg* and *ddeg* on the CN of Section 1.2.
2. Find a CN where the orders given by *deg* and *ddeg* are different.
3. Show formally that this heuristic is static, i.e., its ordering can be computed before search.

2 Symmetries

2.1 Dihedral Group

The 8 symmetries of a square forms a group called a dihedral group.



(a) r^0 (b) r^{90} (c) r^{180} (d) r^{270}

(e) f^v (f) f^h (g) f^d (h) f^c

Show that $\{r^{90}, f^d\}$ is a generating set of the dihedral group.

2.2 Breaking Symmetries of the Social Golfer Problem

We are given a model for the social golfer.

```
from pycsp3 import *

nGroups, size, nWeeks = data or (4, 4, 5)  # size is the size of the groups
nPlayers = nGroups * size

# g[w][p] is the group admitting on week w the player p
g = VarArray(size=[nWeeks, nPlayers], dom=range(nGroups))

satisfy(
  # ensuring that two players don't meet more than one time
  [(g[w1][p1] != g[w1][p2]) | (g[w2][p1] != g[w2][p2])
    for w1, w2 in combinations(nWeeks, 2)
    for p1, p2 in combinations(nPlayers, 2)],

  # respecting the size of the groups
  [Cardinality(g[w], occurrences={i: size for i in range(nGroups)})
    for w in range(nWeeks)],
)
```

Propose symmetry-breaking constraints for this model.