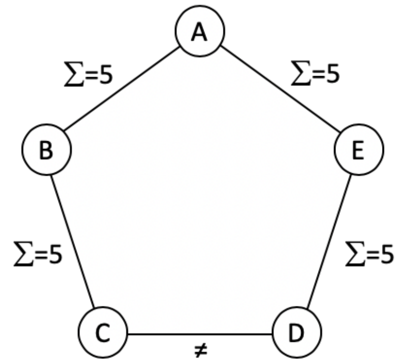
# Constraint Programming

2019/2020– Mini-Test #1
Friday, 31 October, 20:00 h, Room 127-II
Duration: 1.5 h (open book)

## 1. Finite domain Constraints - Propagation (6 pts)




Consider the constraint network on the right, where nodes represent variables, all with domain {**1,2,3,4**}. Arcs labelled Σ=k (e.g. between variables x and y) constrain the connected variables to have a sum equal to k (i.e. x+y = k). Arcs labelled ≠ indicate constraints of difference.

a) **(2 pt)** Is the problem satisfiable? If so, how many different solutions exist? Justify your answer.

b) **(1 pt)** What pruning is achieved initially, if node-consistency is maintained? And arc-consistency?

c) **(2 pt)** Are there any implicit binary constraint in the network? What type of consistency would be needed to infer such constraints? Justify your answer.

d) **(1 pt)** What pruning would be achieved by path-consistency? Justify your answer.

## 2. Global Constraints (5 pts)

Consider a "global" constraint, `pairing,` that enforces the elements of two arrays `A` and `B` of decision variables with equal size , whose domains are composed of values in the range `0..k`, to have the same number of values, i.e. if `q` variables of `A` take value `p`, then `q` variables of `A` should also take that value.

For example, arrays `A = [2,2,7,7,1,7]` and `B = [7,1,7,2,7,2]` satisfy the constraint, but arrays `A` and `C = [2,2,2,7,1,7]` do not (since value 2 appears twice in `A` but three times in `C`).

a) **(2pt)** Implement in Choco this constraint in a function **pairing** with signature

```
function void pairing(Model md, IntVar [] A, IntVar [] B, int k)
```

**Suggestion:** Use the predefined count constraint, with the syntax below, which constrains the number of variables in `A` to have value `v` to be exactly `c`.

```
count(int v, IntVar [] A , IntVar c)
```

b) **(2 pt)** If your implementation of this global constraint were such that it maintained domain consistency, what values would be pruned from the domain of the array A, if the domain of its elements were.

```
A[0] in {6, 8, 9}          B[0] in {1, 3, 5}
A[1] in {1, 3, 5}          B[1] in {1, 2, 4, 6, 8}
A[2] in {3, 5, 7, 9}       B[2] in {1, 4, 5}
A[3] in {4, 9}             B[3] in {1, 2, 7}
A[4] in {6, 8}             B[4] in {1, 2, 6}
```

c) **(1 pt)** In the same conditions of the previous item, assume that constraint `A[1]+B[3] < 7` is posted. Would there be any further pruning of the domains?

## 3. Modelling with Finite Domain Constraints (9 pts)

**Low Autocorrelation Binary Sequences (Prob. 5 of CSPLIB)**

These problems have many practical applications in communications and electrical engineering. The objective is to construct a binary sequence $S_i$ of length n that minimizes the autocorrelations between bits. Each bit in the sequence takes the value +1 or -1. With non-periodic (or open) boundary conditions, the k[th] autocorrelation, $C_k$ is defined to be $\sum_{i=0}^{n-k-1} S_i \times S_{i+k}$.

For example, for sequence $S_1$ = [1, -1, 1, 1, -1] with n = 5 it is

$C_1$ = -2 = $S_0 * S_1 + S_1 * S_2 + S_2 * S_3 + S_3 * S_4$ = – 1 – 1 + 1 –1
$C_2$ = -1 = $S_0 * S_2 + S_1 * S_3 + S_2 * S_4$ = 1 – 1 – 1
$C_3$ = 2 = $S_0 * S_3 + S_1 * S_4$ = 1 + 1
$C_4$ = -1 = $S_0 * S_4$ = -1

The ultimate aim (i.e. problem LABS_OPT(n)), is to minimize a sequence S of size n, for which the sum of the squares of these autocorrelations, $E = \sum_{k=1}^{n-1} C_k^2$, is minimal. Nevertheless we may also consider the satisfaction version of the problem, LABS_SAT(n, Z) that aims to find a sequence S of size n with an autocorrelation less that a certain value Z. In the example above, we have

$E(S_1) = 10 = (-2)^2 + (-1)2 + (2)^2 + (-1)^2$

However, for sequence $S_2$ = [-1, -1, -1, 1, -1] we would have $E(S_2) = 2 = (0)^2 + (-1)^2 + (0)^2 + (1)^2$, which is lower than $E(S_1)$.

Hence, $S_1$ is not a solution of the problem LABS_SAT(5,9) but $S_2$ is. In fact, $S_2$ is a solution of the problem LABS_OPT(5), since there is no sequences $S_x$ of size 5 for which $E(S_x) < 2$.

a) **(2 pt)** Specify a model for this problem in Choco. First, declare the decision variables you adopt as well as their domains, together with the model and solver you propose. Assume an arbitrary value of n.

b) **(4 pt)** (SAT) Next, what constraints would you consider to model the satisfaction problem (n,Z), i.e. find a binary sequence S (± 1 bits) such that E(S) < Z?

**Note:** To simplify notation, assume that to model the summation operator, where X is a decision variable, `b >= a` are integers, and E[i] are expressions on integers (decision) variables.

$$X = \sum_{i=a}^{b} E[i]$$

you can use a built-in constraint (similar to that there is in Comet) with syntax

```
model.sum(i in (a, b), E[i], X).post()
```

Assume further that Boolean (decision) variables are automatically cast to 0/1 (decision) variables, that can be summed.

c) **(1 pt)** (OPT) Adapt the previous model to consider the minimization problem, i.e. find a binary S (± 1 bits) that minimises E(S).

d) **(2 pt)** Notice that for any sequence S there is a symmetric sequence T (i.e. where T[i] = - S[i], for all bits). How would you adapt your model to avoid computing symmetric sequences?