Example 5 (Classic 2-n)

A first remark is that in this example, there is a variable symmetry between all the pairs \((A_i, B_i)\) of variables corresponding to places. This symmetry is easy to detect (purely syntactical) and can be eliminated through the usual ordering of variables, by adding the constraints \(A_i \leq B_i\).

This classical CSP optimization is enough to avoid most of the trivial exponential blow-ups and corresponds to the initial phase of parallel places detection and merging of the equality classes optimization [21] for the standard Fourier-Motzkin algorithm. Note however that in that method, classes of equivalent variables are detected and eliminated before and during the invariant computation, which would correspond to local symmetry detection and was not implemented in our prototype.

Moreover, in [21], equality class elimination is done through replacement of the symmetric places by a representative place. The full method reportedly improves by a factor two the computation speed. Even if in the context of the original article this is done only for ordinary Petri nets (Petri nets where the weights are only 0 or 1), we can see that it can be even more efficient to use this replacement technique in our case:

Example 6

\[ A + B \Rightarrow 4 \times C \]

Instead of simply adding \(A \leq B\) to our constraints, which will lead to 3 solutions when \(C = 1\) before symmetry expansion: \((A, B) \in \{(0, 4), (1, 3), (2, 2)\}\), replacing \(A\) and \(B\) by \(D\) will reduce to a single solution \(D = 4\) before expansion of the subproblem \(A + B = D\).

This partial detection of independent subproblems, which can be seen as a complex form of symmetry identification, can once again be done syntactically at the initial phase, and can be stated as follows: