where $j' < j$ forcing the second number to be greater than the first. We may also subtract each value from $n-1$ to obtain another sequence so we can add clauses to ensure that the first value is no greater than $n/2$:

$$\overline{s_{0,j}}$$

where $j \leq n/2$. Gent et al. also use a clever technique to boost the performance of a constraint solver by a factor of 50. They point out a *conditional symmetry* of AIS, which is a symmetry that occurs during search when a subset of the variables have been assigned values. Breaking the conditional symmetry is not easy in the original AIS model, but they generalise AIS to a new problem containing additional symmetry (rotation of sequence values) that can easily be broken, which incidentally also breaks the symmetry and conditional symmetry in AIS. This technique does not appear to have been exploited in SAT but would make an interesting exercise.

The AIS example illustrates the unpredictable but potentially beneficial effects of adding implied clauses, the selective use of Tseitin variables, the exploitation of subformula polarity to reduce the size of the encoding, the use of non-obvious implied clauses, and the possibility of better models that might be found by ingenuity.

2.3.3. Stable marriages

In the above examples there was nothing very surprising about the variable definitions themselves: they simply corresponded to assignments of constraint variables to domain values. In our next example, taken from a study by Gent et al. [GIM+01], the variable definitions are much less obvious. In the stable marriage problem we have n men and n women. Each man ranks the women in order of preference, and women similarly rank men. The problem is to marry men and women in a *stable* way, meaning that there is no incentive for any two individuals to elope and marry each other. In the version of the problem considered, incomplete preference lists are allowed, so that a man [woman] might be unacceptable to some women [men]. A person is willing to leave his/her current partner for a new partner only if he/she is either unmatched or considers the new partner better than the current partner. A pair who mutually prefer each other are a blocking pair, and a matching without blocking pairs is stable. The problem of finding a stable matching is not NP-complete but an extended version allowing preference ties is, and can be modelled in a similar way; here we discuss only the simpler problem.

A direct encoding of Gent et al.'s first constraint model would contain a Boolean variable for each man-woman pair, which is true when they are matched. However, they also present a second model that is more compact and already in SAT form. They define a variable $x_{i,p}$ to be true if and only if man i is either unmatched or matched to the woman in position p or later in his preference list, where $1 \leq p \leq L_{m,i}$ and $L_{m,i}$ is the length of his list. They also define $x_{i,L_{m,i}+1}$ which is true if and only if man i is unmatched. Similarly they define variables $y_{j,q}$ for each woman j. Note that each SAT variable corresponds to a set of possibilities, unlike those in our previous examples; also that marriages are modelled indirectly via the preference lists.

The clauses are as follows. Each man or woman is either matched with someone in their preference list or is unmatched, which is enforced by the unit clauses

$$x_{i,1} \wedge y_{j,1}$$

If a man gets his p^{th} choice then he certainly gets his $p+1^{th}$ choice:

$$\overline{x_{i,p}} \vee x_{i,p+1}$$

Similarly for women:

$$\overline{y_{j,q}} \vee y_{j,q+1}$$

The x and y variables must be linked so that if man i is matched to woman j then woman j is also matched to man i. We can express the fact that man i is matched to the woman in position p in his preference list indirectly by the assignments $[x_{i,p}, \overline{x_{i,p+1}}]$ so the clauses are:

$$(\overline{x_{i,p}} \vee x_{i,p+1} \vee y_{j,q}) \wedge (\overline{x_{i,p}} \vee x_{i,p+1} \vee \overline{y_{j,q+1}})$$

where p is the rank of woman j in the preference list of man i, and q is the rank of man i in the preference list of woman j. Similarly:

$$(\overline{y_{j,q}} \vee y_{j,q+1} \vee x_{i,p}) \wedge (\overline{y_{j,q}} \vee y_{j,q+1} \vee \overline{x_{i,p+1}})$$

Stability is enforced by clauses stating that if man i is matched to a woman he ranks below woman j then woman j must be matched to a man she ranks no lower than man i:

$$\overline{x_{i,p}} \vee \overline{y_{j,q+1}}$$

where the man finds the woman acceptable (mentioned in the preference list). Similarly:

$$\overline{y_{j,q}} \vee \overline{x_{i,p+1}}$$

This problem provides a nice illustration of the use of non-obvious variable definitions.

Here it is worth pointing out another common source of error in SAT encoding: omitting common-sense axioms. The meaning of variable $x_{i,p}$ being true is that man i is matched to the woman in position p or later in his list. This must of course be explicitly stated in the SAT encoding via clauses $\overline{x_{i,p}} \vee x_{i,p+1}$. In the same way, if we are trying to construct a tree, a DAG or some other data structure then we must include axioms for those structures. However, in some cases these axioms are implied by other clauses. For example, if we are trying to construct a permutation of n numbers with some property then we would normally use at-least-one and at-most-one clauses, because the permutation is a list of length n with each position containing exactly one number. But if we add clauses stating that no two positions contain the same number then the at-most-one clauses become redundant.