However, if the aim is to classify knots, we need the canonical orientation only for knot projections, and we should stop the computation as soon as we find that (S, f) does not satisfy Rule 2, as in the following algorithm.

Algorithm

Given S, take $A = \{1, a_1\}$, $f(1) = 1$, $f(a_1) = -1$
If $A \neq \emptyset$, select i = least member of A
Take $\phi_i(i) = 1$
If $\phi_i(x-1)$ has been found, where $x-1$ is taken mod $2n$,
  If $a_x \notin [i, a_i]$, $\phi_i(x) = \phi_i(x-1)$
  If $a_x \in [i, a_i]$, $\phi_i(x) = -\phi_i(x-1)$
End with $\phi_i(i-1)$
Take $D_i = \{1, 2, \ldots, 2n\} \backslash [i, a_i]$
  If $D_i \neq \emptyset$, select x = least member of D_i
  If $x < i$
    If $a_x \notin [i, a_i]$
      If $\phi_i(x)\phi_i(a_x) = 1$, remove x and a_x from D_i
      If $\phi_i(x)\phi_i(a_x) = -1$, reject S, proceed to new S
    If $a_x \in [i, a_i]$
      If $f(x)$ is already defined
        If $\phi_i(x)\phi_i(a_x)f(i) = f(x)$, remove x from D_i
        If $\phi_i(x)\phi_i(a_x)f(i) = -f(x)$, reject S, proceed to new S
      If $f(x)$ not already defined, let $f(x) = \phi_i(x)\phi_i(a_x)f(i)$, $f(a_x) = -f(x)$
        If $a_{x-1} = a_x \pm 1$ ($a_x + 1$ is mod $2n$), remove x from D_i
        If $a_{x-1} \neq a_x \pm 1$, add x and a_x to A, remove x from D_i
  If $x > a_i$
    If $a_x \notin [i, a_i]$, remove x and a_x from D_i
    If $a_x \in [i, a_i]$
      If $f(x)$ is already defined, remove x from D_i
      If $f(x)$ not already defined, let $f(x) = \phi_i(x)\phi_i(a_x)f(i)$, $f(a_x) = -f(x)$
        If $a_{x-1} = a_x \pm 1$ ($a_x + 1$ is mod $2n$), remove x from D_i
        If $a_{x-1} \neq a_x \pm 1$, add x and a_x to A, remove x from D_i
  If $D_i \neq \emptyset$, select x = least member of D_i, etc.
  If $D_i = \emptyset$, remove i and a_i from A
If $A \neq \emptyset$, select i = least member of A, etc.
If $A = \emptyset$, record that (S, f) is realizable, proceed to new S.

Thus tabulating knot projections with n crossings, and their orientations, is algorithmic. We also use algorithms for listing the knots with n crossings, but the resulting list may have duplicates. Our methods of eliminating the duplicates, and