Algorithm 1 Constraint Solving over Discrete Domains

function SOLVE($C, X = \{x_1, ..., x_n\}, D_0$) ▷ Constraints, variables, initial domains
  $\sigma \leftarrow \emptyset$ ▷ Assignment function, initially empty
  $D \leftarrow D_0(x_1) \times \cdots \times D_0(x_n)$ ▷ Domains
  $\Delta \leftarrow \emptyset$ ▷ Decision stack
  while true **do**
    $\sigma, D \leftarrow$ Propagate(C, σ, D) ▷ Propagate using arc-consistency
    if Conflict $\notin \sigma$ **then**
      if AllAssigned(σ) **then** ▷ Check if all variables are assigned
        return Solution(σ) ▷ Solved, return the full assignment in solution format
      else
        $\sigma, x \leftarrow$ MakeDecision(C, σ, D) ▷ Assign some value to an unassigned variable
        $\Delta \leftarrow \Delta$.push(σ, x, D) ▷ Create backtracking point
      end if
    else
      if $\Delta == \emptyset$ **then**
        return NoSolution
      else
        $\sigma, D \leftarrow$ Backtrack(Δ) ▷ Backtrack to the latest decision
      end if
    end if
  end while
end function
function PROPAGATE(C, σ, D)
  while True **do**
    if $D(x_i) == \{a\}$ for some unassigned x_i **then**
      $\sigma \leftarrow \sigma \cup \{(x_i, a)\}$ ▷ Make assignment if domain becomes singleton
    end if
    if $(x, a) \in \sigma$ and $|D(x)| > 1$ **then** ▷ If x has been assigned a value, update its domain
      $D(x) \leftarrow \{a\}$
    end if
    if $D(x_i)$ is empty for some x_i **then**
      return $\sigma \cup \{\text{Conflict}\}, D$
    end if
    if there exists i, j such that $a_i \in D(x_i)$ is not consistent with any $a_j \in D(x_j)$ in C **then**
      $D(x_i)$.Remove(a_i)
    else
      return σ, D
    end if
  end while
end function
function MAKEDECISION(C, σ, D)
  if x is unassigned in σ **then**
    $a \leftarrow$ some element in $D(x)$
    return $\sigma \cup \{(x, a)\}, x$
  end if
end function
function BACKTRACK(Δ)
  $\sigma, x, D \leftarrow \Delta.pop()$
  $a \leftarrow \sigma(x)$
  σ.remove((x, a))
  $D(x)$.remove(a) ▷ Make sure to remove the previous decision from its domain
  return σ, D
end function