
Algorithm 1 Constraint Solving over Discrete Domains

```
1: function SOLVE( $C, X = \{x_1, \dots, x_n\}, D_0$ ) ▷ Constraints, variables, initial domains
2:    $\sigma \leftarrow \emptyset$  ▷ Assignment function, initially empty
3:    $D \leftarrow D_0(x_1) \times \dots \times D_0(x_n)$  ▷ Domains
4:    $\Delta \leftarrow \emptyset$  ▷ Decision stack
5:   while true do
6:      $\sigma, D \leftarrow \text{Propagate}(C, \sigma, D)$  ▷ Propagate using arc-consistency
7:     if Conflict  $\notin \sigma$  then
8:       if AllAssigned( $\sigma$ ) then ▷ Check if all variables are assigned
9:         return Solution( $\sigma$ ) ▷ Solved, return the full assignment in solution format
10:      else
11:         $\sigma, x \leftarrow \text{MakeDecision}(C, \sigma, D)$  ▷ Assign some value to an unassigned variable
12:         $\Delta \leftarrow \Delta.\text{push}(\sigma, x, D)$  ▷ Create backtracking point
13:      end if
14:    else
15:      if  $\Delta == \emptyset$  then
16:        return NoSolution
17:      else
18:         $\sigma, D \leftarrow \text{Backtrack}(\Delta)$  ▷ Backtrack to the latest decision
19:      end if
20:    end if
21:  end while
22: end function
23: function PROPAGATE( $C, \sigma, D$ )
24:  while True do
25:    if  $D(x_i) == \{a\}$  for some unassigned  $x_i$  then
26:       $\sigma \leftarrow \sigma \cup \{(x_i, a)\}$  ▷ Make assignment if domain becomes singleton
27:    end if
28:    if  $(x, a) \in \sigma$  and  $|D(x)| > 1$  then ▷ If  $x$  has been assigned a value, update its domain
29:       $D(x) \leftarrow \{a\}$ 
30:    end if
31:    if  $D(x_i)$  is empty for some  $x_i$  then
32:      return  $\sigma \cup \{\text{Conflict}\}, D$ 
33:    end if
34:    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
35:       $D(x_i).\text{Remove}(a_i)$ 
36:    else
37:      return  $\sigma, D$ 
38:    end if
39:  end while
40: end function
41: function MADEDECISION( $C, \sigma, D$ )
42:  if  $x$  is unassigned in  $\sigma$  then
43:     $a \leftarrow \text{some element in } D(x)$ 
44:    return  $\sigma \cup \{(x, a)\}, x$ 
45:  end if
46: end function
47: function BACKTRACK( $\Delta$ )
48:   $\sigma, x, D \leftarrow \Delta.\text{pop}()$ 
49:   $D(x).\text{remove}(\sigma(x))$  ▷ Make sure to remove the previous decision from its domain
50:  return  $\sigma, D$ 
51: end function
```
