AI algorithms: Constraint Relaxation and Hybrids

Lookahead check
AC1
AC3

Input:
A constraint network with n variables v_i
A list C of constraints $c(v_i, v_j)$
A set of possible assignments $a_{i,k}$ for each variable v_i

Output:
An assignment $a_{i,k}$ for each variable v_i where all constraints in C are respected

Algorithm:

```
repeat
  $del\_occured \leftarrow false$
  queue $Q \leftarrow$ all constraints from $C$
  while $Q$ not empty do
    $c(v_i, v_j) \leftarrow$ get (and remove) from $Q$
    if assignment $a_{i,k}$ (or $a_{j,k}$) inconsistent w.r.t. $c(v_i, v_j)$ then
      remove $a_{i,k}$ (or $a_{j,k}$)
      $del\_occured \leftarrow true$
      add all constraints in $C$ involving $v_i$ (or $v_j$) to $Q$
    end if
  end while
until not $del\_occured$
```

(Chronological) Backtracking
Forward checking
Lookahead checking

Input:
A constraint network with n variables v_i
A list C of constraints $c(v_i, v_j)$
A set of possible assignments $a_{i,k}$ for each variable v_i

Output:
An assignment $a_{i,k}$ for each variable v_i where all constraints in C are respected

Algorithm $(depth)$:

```
Lookahead check;
for all $a_{depth,k}$ do
  $v_{depth} \leftarrow a_{depth,k}$
  queue $Q \leftarrow$ constraints from $C$ involving $v_{depth}$
  while $Q$ not empty do
    $c(v_{depth}, v_j)$ (or $c(v_j, v_{depth})$) $\leftarrow$ get (and remove) from $Q$
    if $a_{j,k}$ inconsistent w.r.t. $c(v_{depth}, v_j)$ (or $c(v_j, v_{depth})$) then
      remove $a_{j,k}$
    end if
  end while
  Lookahead check;
  ... see (chronological) backtracking algorithm
end for
```