Checking Consistency

Node consistency:

Ensure values in domain of variable X satisfy unary constraints

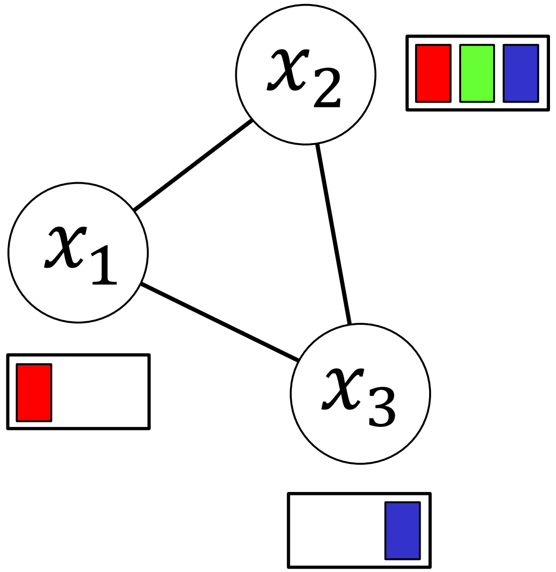
Arc Consistency (AC):

Ensure each value in domain D_i of X_i satisfies binary constraints

Binary constraint $C_{i,j}$ for vars $X_i \leftrightarrow X_j$

If X_i is AC with X_j then:

for each $x \in D_i$ there exists $y \in D_j$ such that $(x, y) \in C_{i,j}$



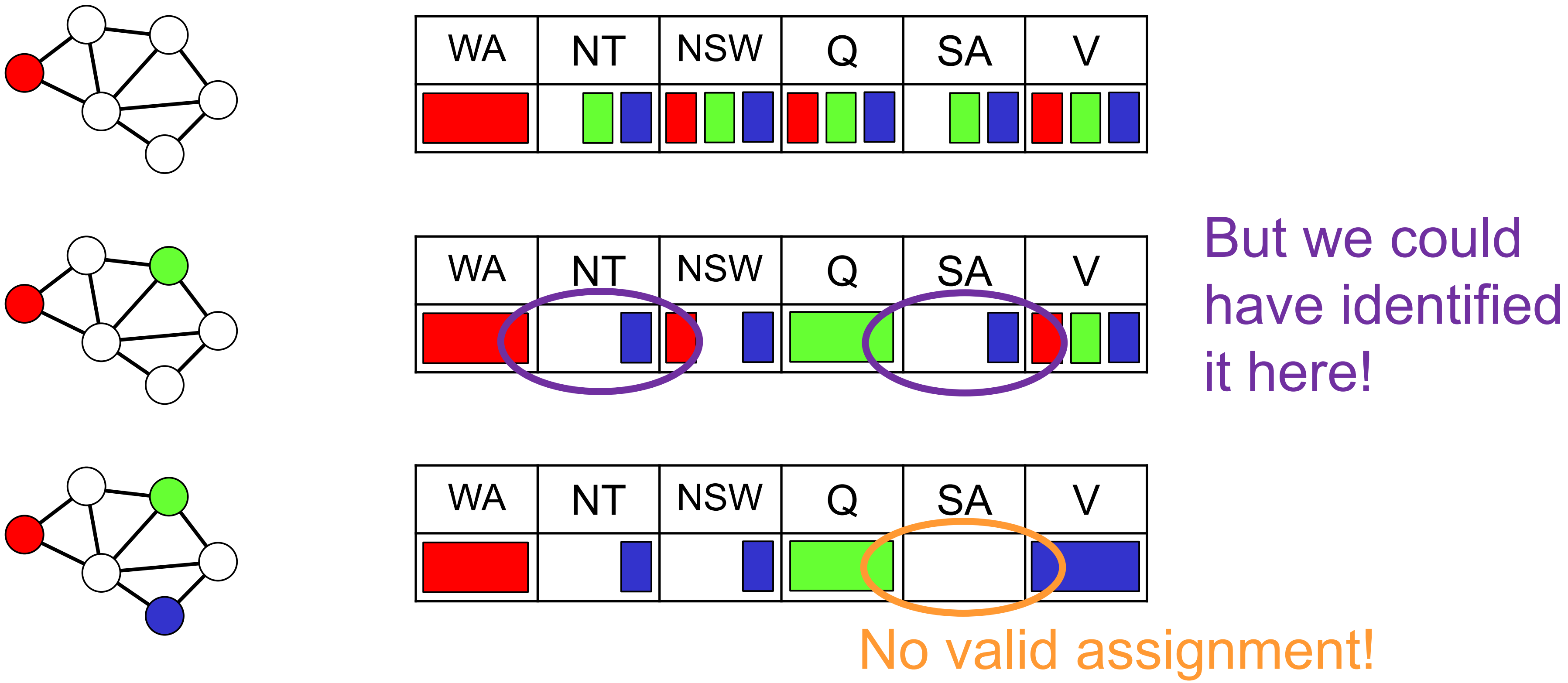
x_1 is AC with x_2 and x_3

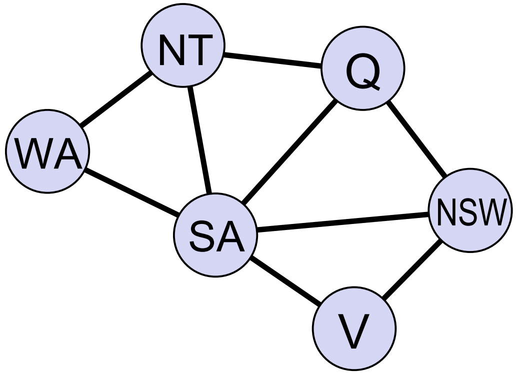
x_3 is AC with x_1 and x_2

x_2 is **not** AC with x_1 and x_3

Forward Checking

Whenever a variable X is assigned, enforce arc consistency for all links $Y \rightarrow X$ for unassigned variables Y





Recursively Enforcing Arc Consistency

Alan Mackworth
Univ. of British Columbia

function AC-3(*csp*)

  while *queue* is not empty **do**

    $(X_i, X_j) \leftarrow$ *queue*.POP()

    revised ← *false*

    for each x **in** D_i **do**

      if there is no value $y \in D_j$ satisfying $C_{i,j}$ **then**

        remove x from D_i

        revised ← *true*

    if *revised* **then**

      if size(D_i) = 0 **then** return *false*

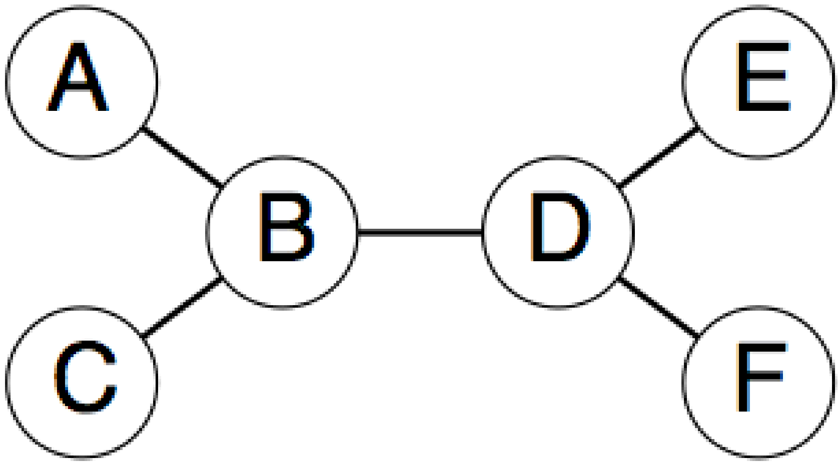
      for each X_k neighbor of X_i (excluding X_j) **do** queue.PUSH((X_k, X_i))

  return *true*

Goes beyond forward checking by propagating constraints

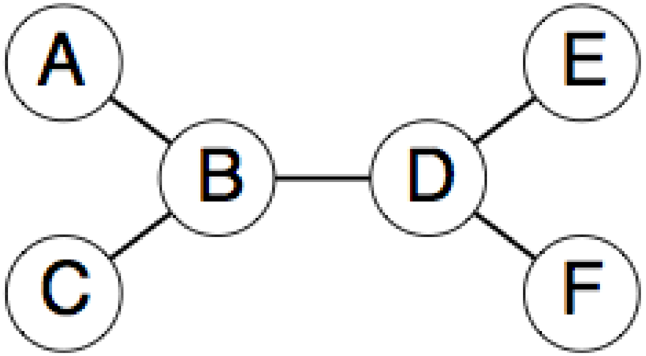
Tree-Structured CSPs

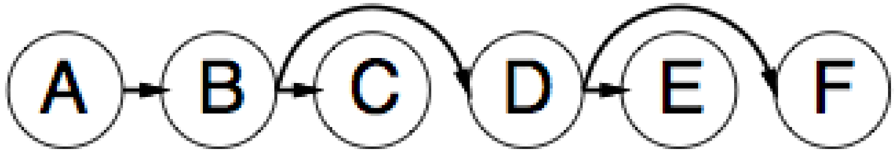
If the constraint graph has no loops (also called cycles), then the CSP can be solved in $O(nd^2)$ time (vs $O(d^n)$ in general)



Solving Tree-Structured CSPs

1. Choose a variable as root, order variables from root to leaves such that every node's parent precedes it in the ordering





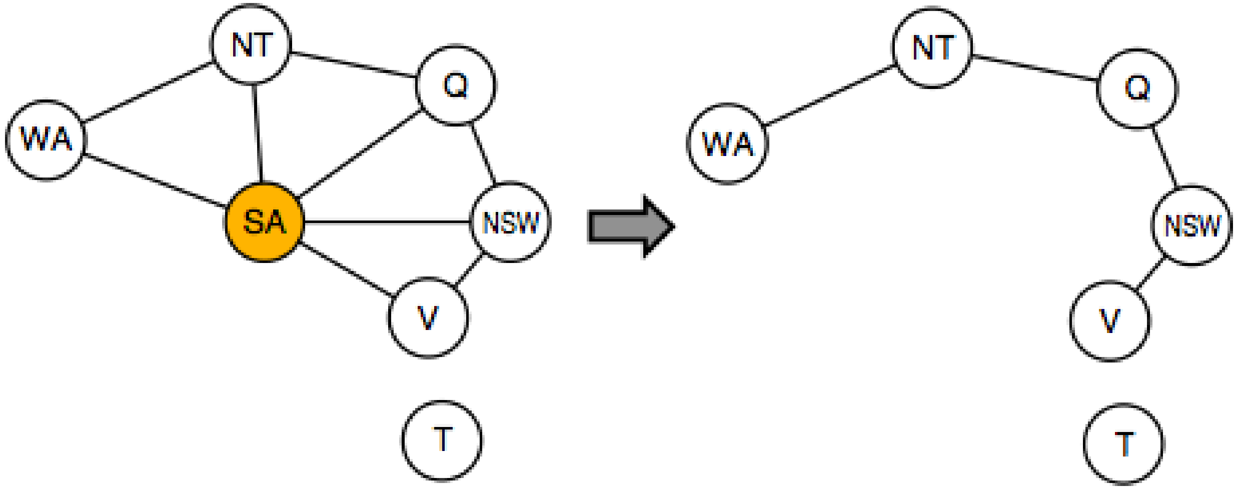
2. For j from n down to 2, apply $\textsc{RemoveInconsistent}(Parent(X_j), X_j)$

3. For j from 1 to n, assign X_j consistently with $Parent(X_j)$

Cutset Conditioning

Rina Dechter
Univ. of California, Irvine

Conditioning: instantiate a variable, prune its neighbors' domains



Cutset conditioning: instantiate (in all ways) a set of variables such that the remaining constraint graph is a tree

Cutset size c $\Rightarrow$ runtime $O(d^c \cdot (n-c)d^2)$, very fast for small c

Summary

Node and arc consistency refer to whether the domains for a set of variables satisfy the constraints

Forward checking enforces arc consistency for all neighbors of a variable that has been assigned during search

Constraint propagation goes further and enforces arc consistency recursively until all domains are consistent (AC-3 algorithm)

The Maintaining Arc Consistency version of backtracking search calls AC-3 after each variable assignment

Tree-structured graphs can be solved orders of magnitude faster than general graphs. Cutset conditioning can be used with graphs that are “close” to trees