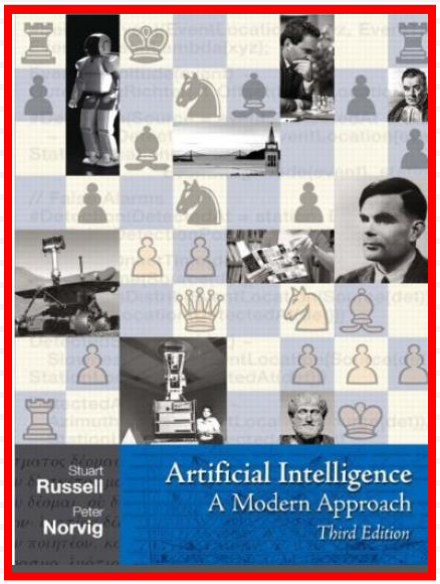
Princess Nora University
Faculty of Computer & Information Systems

جامعة الأميرة نورة بنت عبد الرحمن
Princess Nora Bint Abdul Rahman University

ARTIFICIAL INTELLIGENCE

(CS 370D)

جامعة الأميرة نورة بنت عبد الرحمن
Princess Nora Bint Abdul Rahman University



(CHAPTER-6)
CONSTRAINT SATISFACTION PROBLEMS

Outline

- What is a CSP
- CSP applications
- Backtracking search
- Problem structure and decomposition

Dr. Abeer Mahmoud
(course coordinator)

Mean of Constraint Satisfaction Problems

Dr. Abeer Mahmoud
(course coordinator)

Constraint Satisfaction Problems

- A constraint satisfaction problem consists of three components,

  X, D, and C

X is a set of variables, **{Xi, . , XT,}.**

D is a set of domains, {Di,…, Dn.}, one for each variable.

C is a set of constraints that specify allowable values.

  - Each constraint C_i limits the values that variables can take,
    - e.g., $V_1 \neq V_2$

A *state* is defined as an *assignment* of values to some or all variables.

Dr. Abeer Mahmoud
(course coordinator)

CSPs (continued)

- An assignment is *complete* when every variable is mentioned.
- A *solution* to a CSP is a complete assignment that satisfies all constraints.
- Some CSPs require a solution that maximizes an *objective function*.

Dr. Abeer Mahmoud
(course coordinator)

Varieties of constraints

- Unary constraints involve a single variable.
  - e.g. *SA* ≠ *green*
- Binary constraints involve pairs of variables.
  - e.g. *SA* ≠ *WA*
- Higher-order constraints involve 3 or more variables.
  - Professors A, B,and C cannot be on a committee together
  - Can always be represented by multiple binary constraints
- Preference (soft constraints)
  - e.g. *red* is better than *green* often can be represented by a cost for each variable assignment
  - combination of optimization with CSPs

Dr. Abeer Mahmoud
(course coordinator)

Constraint Satisfaction Applications

Dr. Abeer Mahmoud
(course coordinator)

CSPs (continued) Examples of Applications

- Scheduling :
  - Scheduling the timetable of lectures
  - Airline schedules
  - Scheduling your MS or PhD thesis exam ☺
- Configuration
- To arrange objects of different sizes in a container, Map Coloring.
- CSP solvers can be faster than state-space searchers because the CSP solver can quickly eliminate large swatches of the search space.

Dr. Abeer Mahmoud
(course coordinator)

Example problem – timetable of university (1)

- state - collection of schedule objects
  - schedule
    - professor(~10^2) (~350 at LU)
    - course (~10^3)
    - time slot (~10) (16 at LU)
    - location (~10^2)
- constraints
- objective function

Dr. Abeer Mahmoud
(course coordinator)

Example problem – timetable of university (1)

- **Constraints**
  - professors assigned to courses
  - one course per location per time slot
  - location capacity >= course (projected) enrollment
  - professors assigned only once per time slot
  - rooms assigned only once per time slot
  - courses in program in different time slots

Dr. Abeer Mahmoud
(course coordinator)

Example problem – timetable of university (1)

- **objective function**

  - minimize total course enrollment in early morning time slots
  - distribute program courses across days
  - accommodate special requests

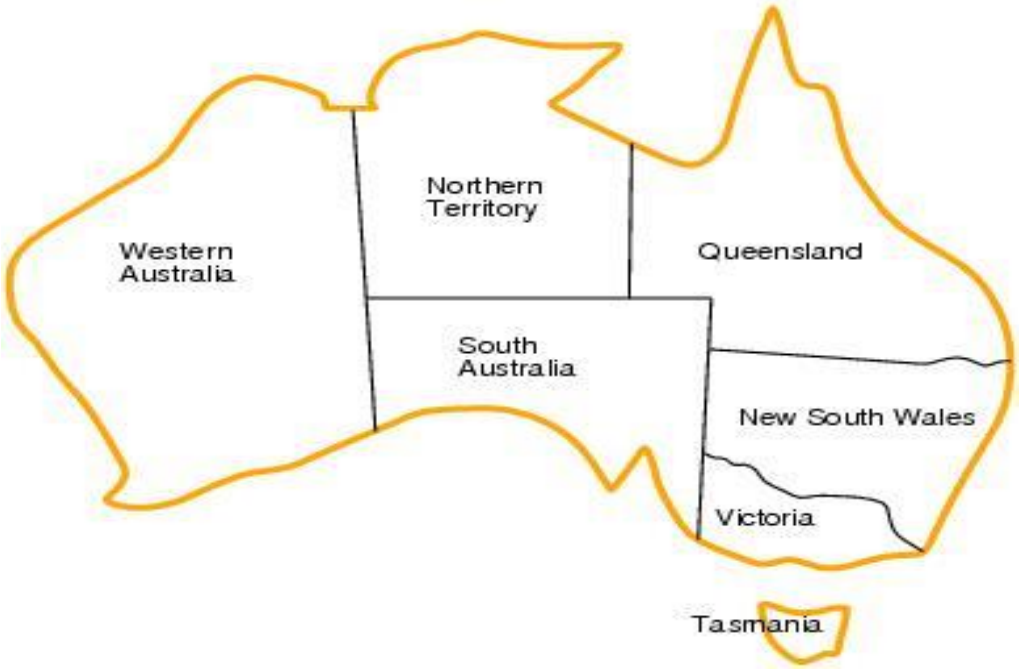
Dr. Abeer Mahmoud
(course coordinator)

Solving CSPs

- analysis of problem state structure is critical
- two basic approaches:
  - start state is null: assign state variables one at a time and backtrack if constraint is violated
  - assign variables 'randomly/intelligently' to make a start state and use greedy search by changing variable assignments

Dr. Abeer Mahmoud
(course coordinator)

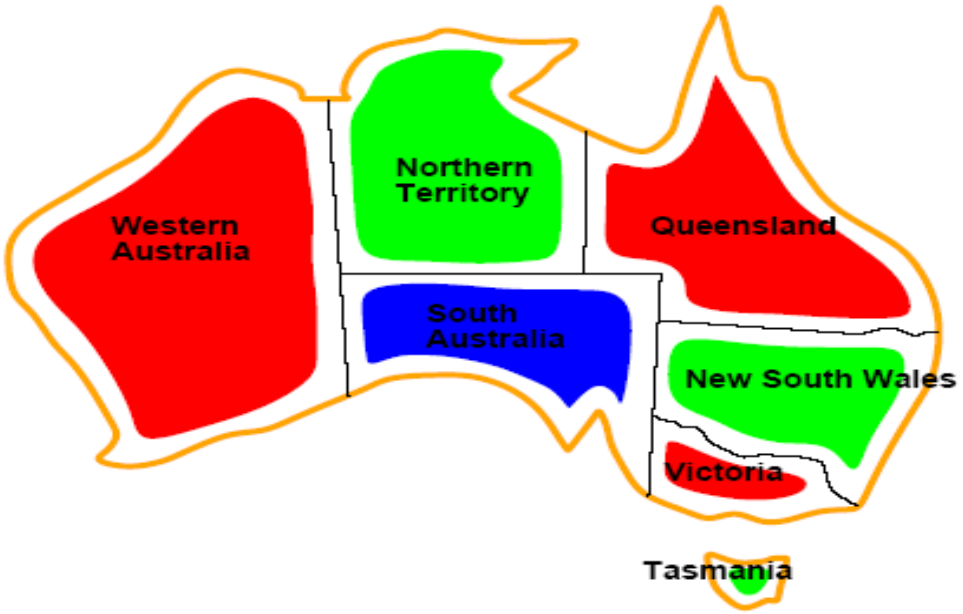
CSP example: map coloring



- Variables: *WA, NT, Q, NSW, V, SA, T*
- Domains: D_i={*red,green,blue*}
- Constraints: adjacent regions must have different colors.
  - E.g. *WA* ≠ *NT*

Dr. Abeer Mahmoud
(course coordinator)

CSP example: map coloring



- Solutions are assignments satisfying all constraints, e.g.

{*WA=red,NT=green,Q=red,NSW=green,V=red,SA=blue,T=green*}

Dr. Abeer Mahmoud
(course coordinator)

Solving CSPs

- As general searching problem.
- Back-Tracking.
- Arc-consistency Checking.
- Heuristics.

Dr. Abeer Mahmoud
(course coordinator)

As General Searching Problem

Dr. Abeer Mahmoud
(course coordinator)

CSP as a standard search problem

- A CSP can easily be expressed as a standard search problem.
- **Incremental formulation**
  - ***Initial State***: the empty assignment {}
  - ***Successor function***: Assign a value to any unassigned variable provided that it does not violate a constraint
  - ***Goal test***: the current assignment is complete
    (by construction its consistent)
  - *Path cost*: constant cost for every step (not really relevant)

Backtracking Search

Dr. Abeer Mahmoud
(course coordinator)

Backtracking Search

- Variable assignments are commutative if the order of any given set of actions has no effect on the outcome.

  i.e., [WA = red then NT = green] same as [NT = green then WA = red].

- Only need to consider assignments to a single variable at each node. and backtracks when a variable has no legal values left to assign.
- Depth-first search for CSPs with single-variable assignments is called **Backtracking Search**:

- ✓ Chooses values for one variable at a time.
- ✓ Checks for constraint violations at each assignment.

- ✓ Backtracks when a variable has no legal values left to ass

Dr. Abeer Mahmoud
(Course Coordinator)

Backtracking search

function BACKTRACKING-SEARCH(*csp*) **return** a solution or failure

return RECURSIVE-BACKTRACKING({} , *csp*)

function RECURSIVE-BACKTRACKING(*assignment*, *csp*) **return** a solution or failure

if *assignment* is complete **then return** *assignment*

var ← SELECT-UNASSIGNED-VARIABLE(VARIABLES[*csp*],*assignment*,*csp*)

for each *value* **in** ORDER-DOMAIN-VALUES(*var, assignment, csp*) **do**

if *value* is consistent with *assignment* according to CONSTRAINTS [*csp*]

then

add {*var*=*value*} to assignment

result ← RRECURSIVE-BACTRACKING(*assignment, csp*)

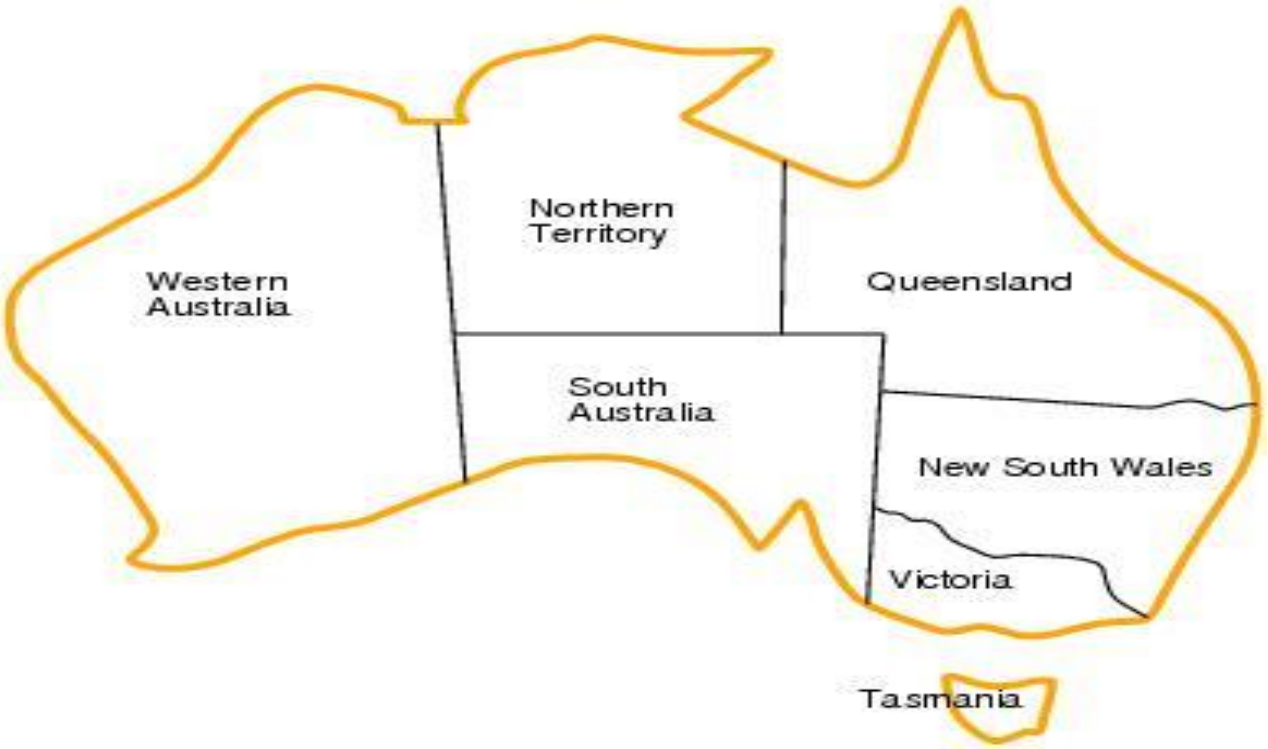
if *result* ≠ *failure* **then return** *result*

remove {*var*=*value*} from *assignment*

return *failure*

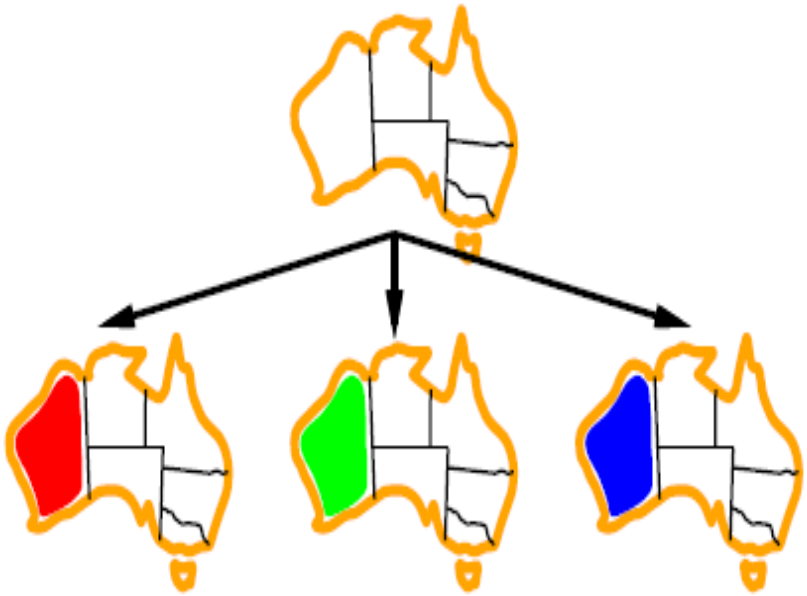
Backtacking

Backtracking example



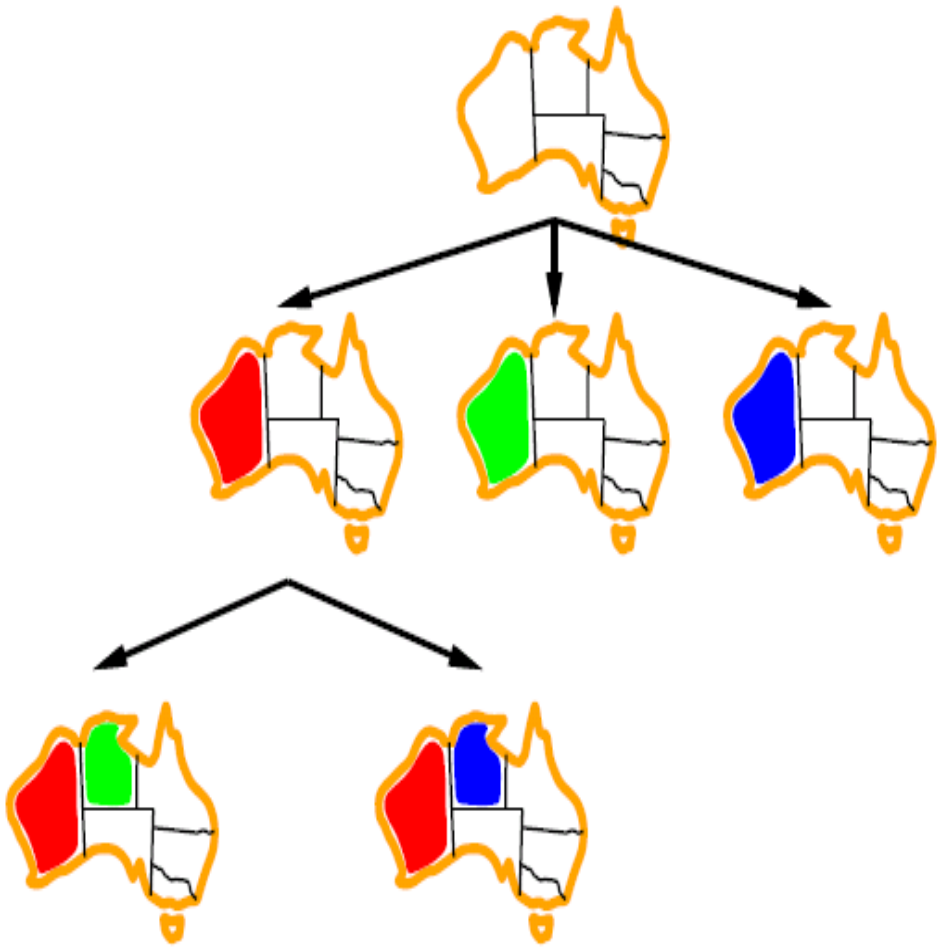
Dr. Abeer Mahmoud
(course coordinator)

Backtracking example



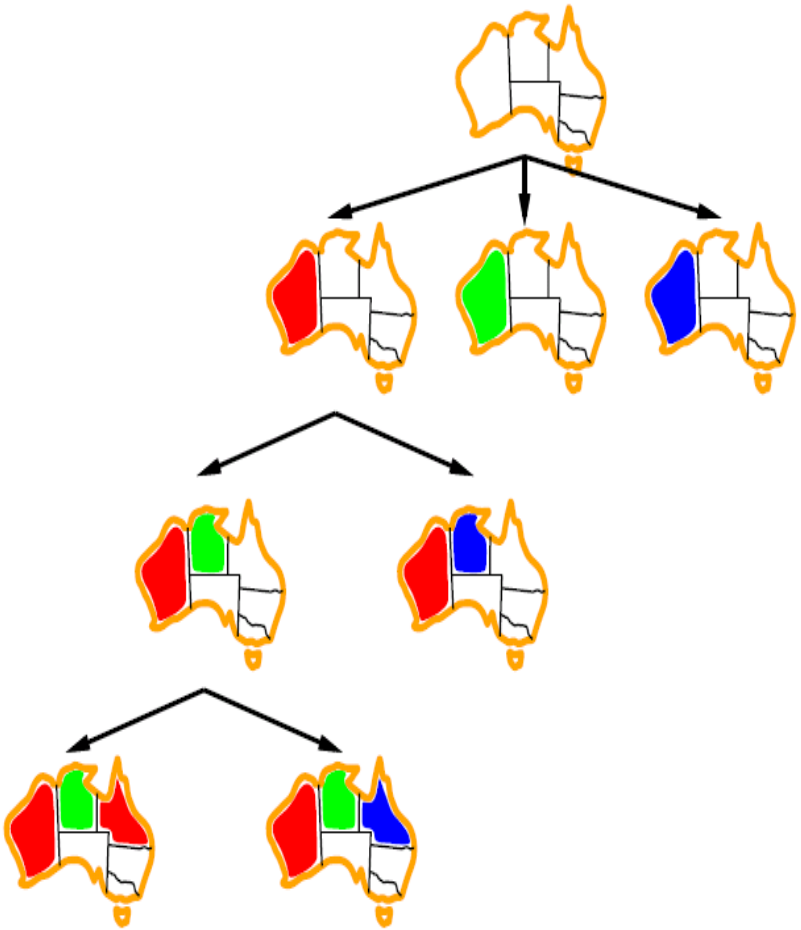
Dr. Abeer Mahmoud
(course coordinator)

Backtracking example



Dr. Abeer Mahmoud
(course coordinator)

Backtracking example



Constraints on SA will eventually cause failure when WA != Q. When not the same color (bottom right), SA cannot be assigned.

The algorithm will backtrack to a node with unexplored states.

For example, **such as WA=red, NT=blue.**

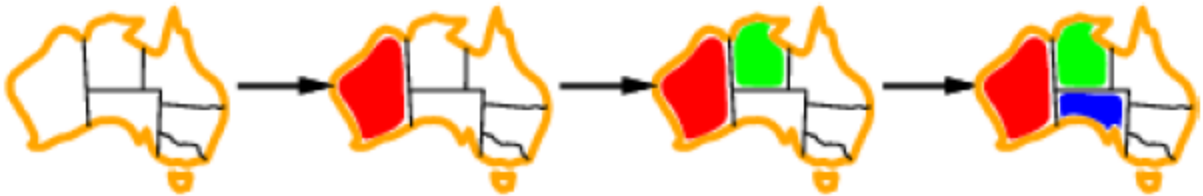
Dr. Abeer Mahmoud
(course coordinator)

Improving Backtracking Efficiency

- **Consider**
  - Which variable should be assigned next?
  - In what order should the variables be tried?
  - Can we detect inevitable failure early?
  - Can we take advantage of problem structure?

Most Constrained Variable or minimum remaining values (MRV)

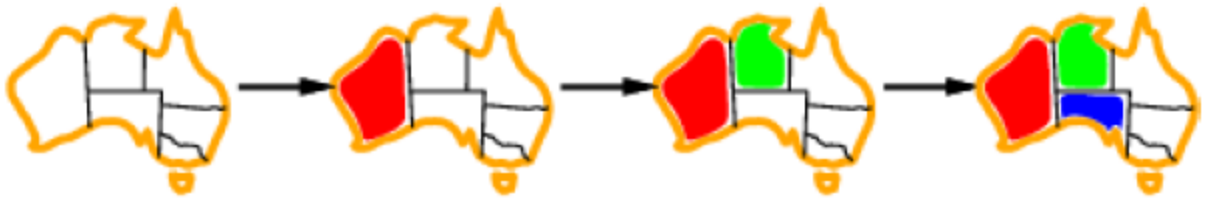
- Heuristic Rule: Choose the variable with the **fewest** legal values.
- **Idea** is that variable with *fewest* choices will likely fail sooner allowing branch to be *pruned* early.
  - Legal values are those that do not conflict with any already assigned to a neighbor (i.e. do not violate constrains).
  - After WA=red, NT and SA have only 2 remaining legal values, the other unassigned variables all have 3.
  - **Choose between NT and SA.**



Dr. Abeer Mahmoud
(course coordinator)

Most constraining variable

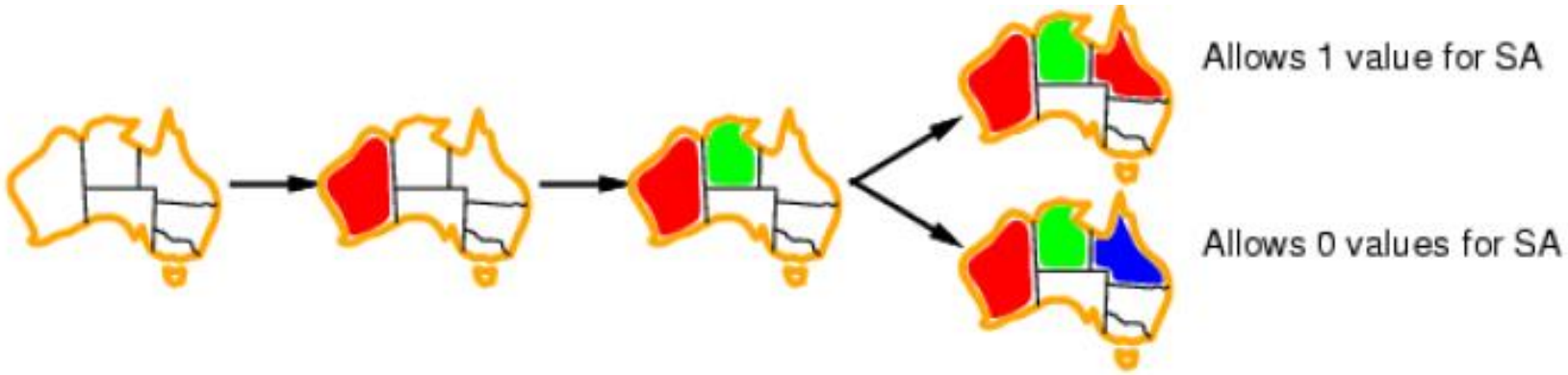
- Choose the variable that constrains the most remaining variables
- SA places constrains on 5 variables; WA, NT, Q, V, NSW only on 3, T on 0.
- Purpose is to prune early by reducing number of choices.
- Tie breaker among minimum remaining variables (MRV)



Dr. Abeer Mahmoud
(course coordinator)

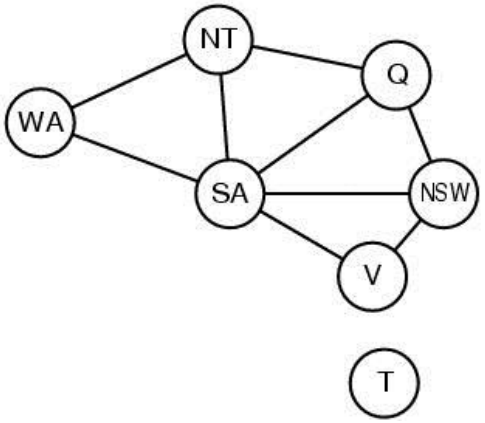
Least Constraining Value

- Heuristic Rule:
- Given a variable, choose the least constraining value



Idea is to explore paths, those with most options, most likely to lead to **a solution.**

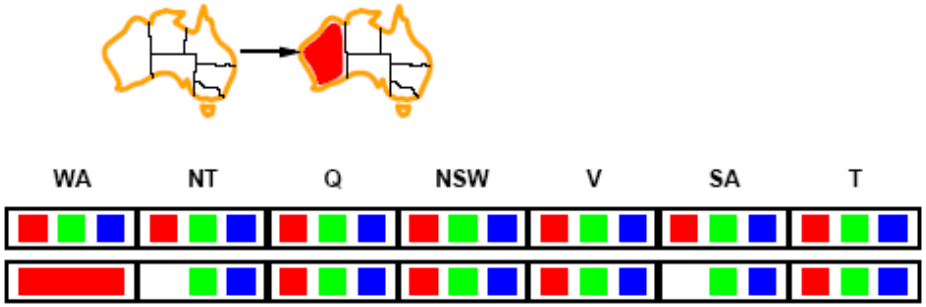
Forward checking

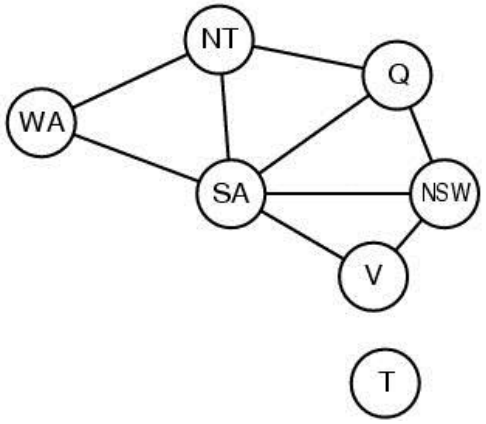


- Can we detect inevitable failure early?
  - *And avoid it later?*
- ***Forward checking idea***: keep track of remaining legal values for unassigned variables.
- Terminate search when any variable has *no legal values*.

Dr. Abeer Mahmoud
(course coordinator)

Forward checking



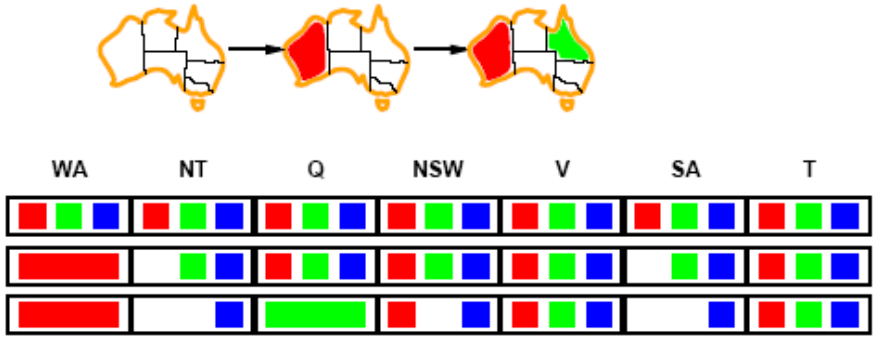


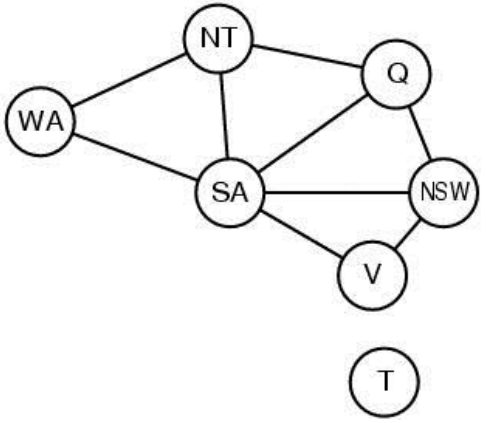
- Assign {*WA=red*}

- Effects on other variables connected by constraints to WA
  - *NT can no longer be red*
  - *SA can no longer be red*

Dr. Abeer Mahmoud
(course coordinator)

Forward checking



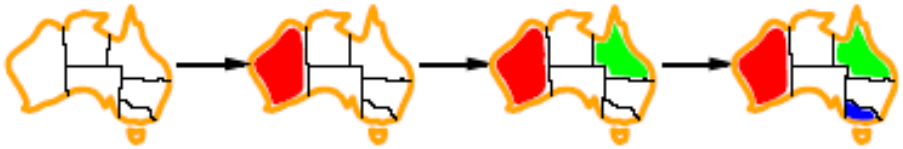


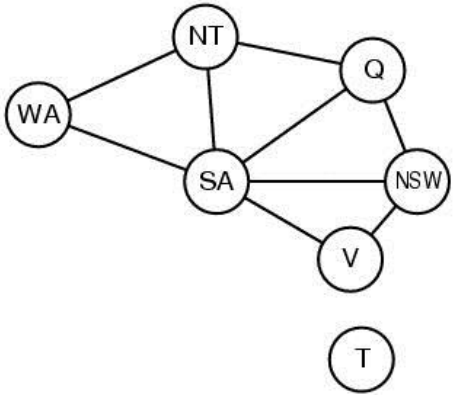
- Assign {Q=*green*}

- Effects on other variables connected by constraints with WA
  - *NT can no longer be green*
  - *NSW can no longer be green*
  - *SA can no longer be green*

Dr. Abeer Mahmoud
(course coordinator)

Forward checking



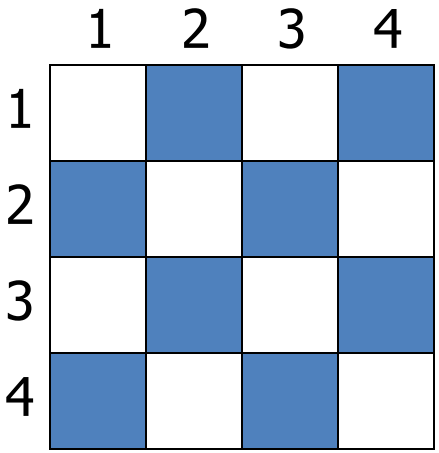


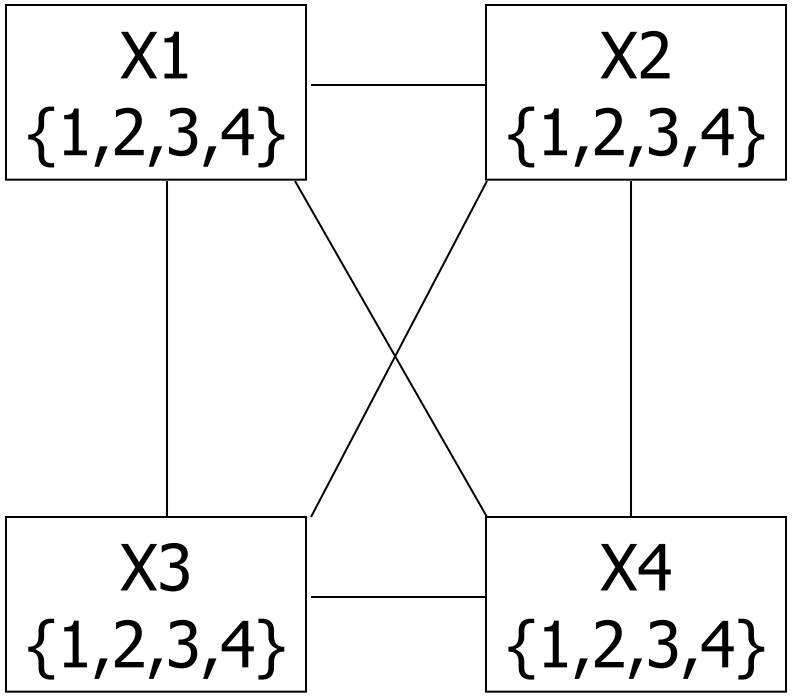
- If *V* is assigned *blue*
- Effects on other variables connected by constraints with WA
  - *NSW can no longer be blue*
  - *SA is* ***empty***

- FC has detected that partial assignment is *inconsistent* with the constraints and backtracking can occur.

Forward checking

Example: 4-Queens Problem

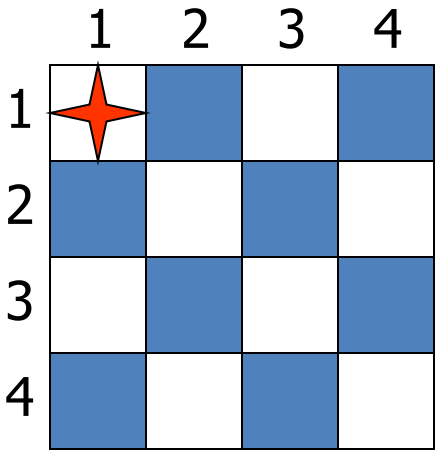


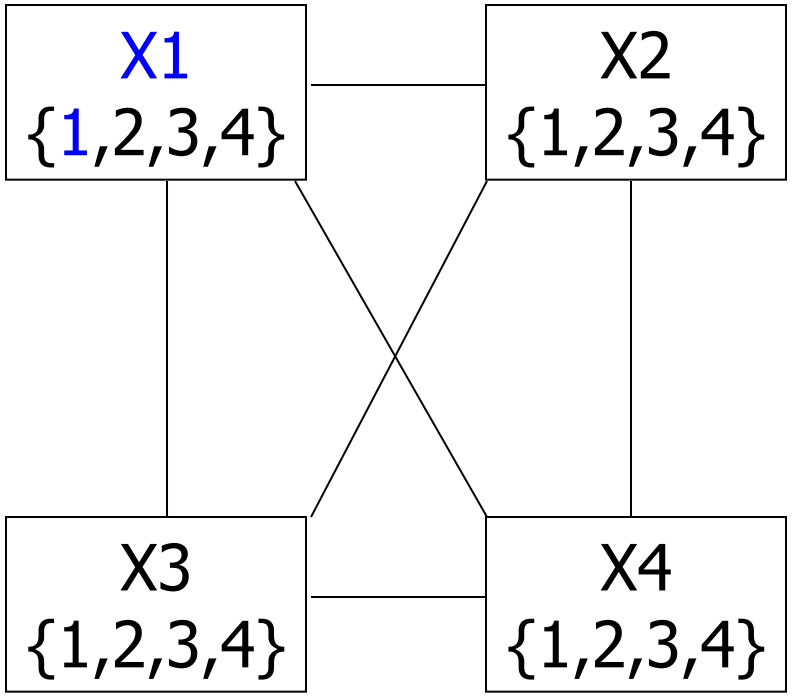


Dr. Abeer Mahmoud
(course coordinator)

Example: 4-Queens Problem

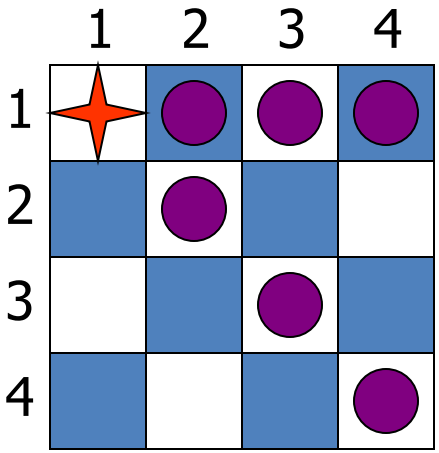
Forward checking

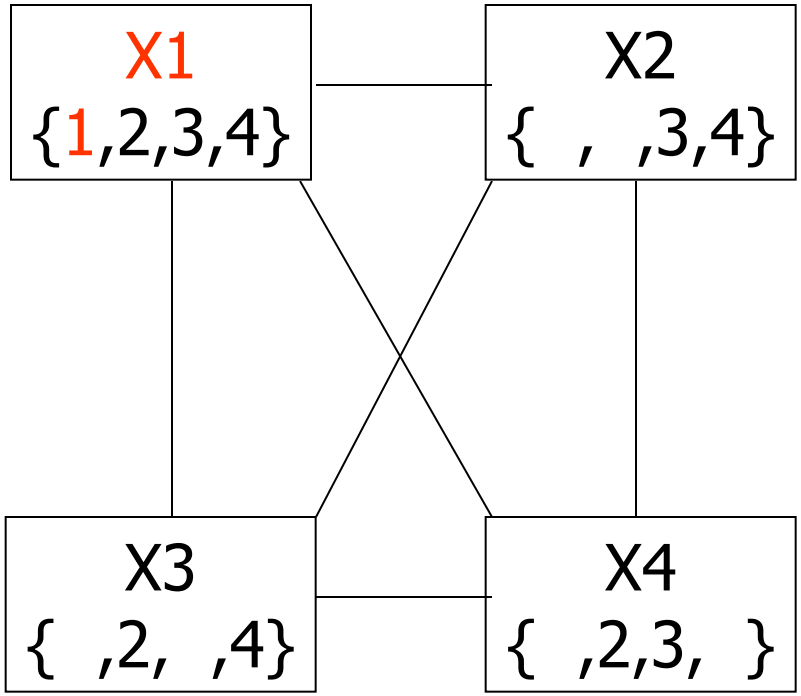




Dr. Abeer Mahmoud
(course coordinator)

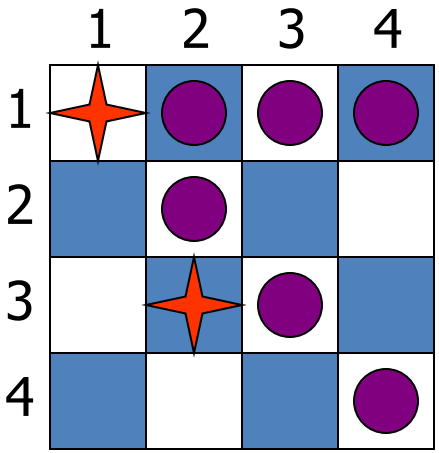
Example: 4-Queens Problem

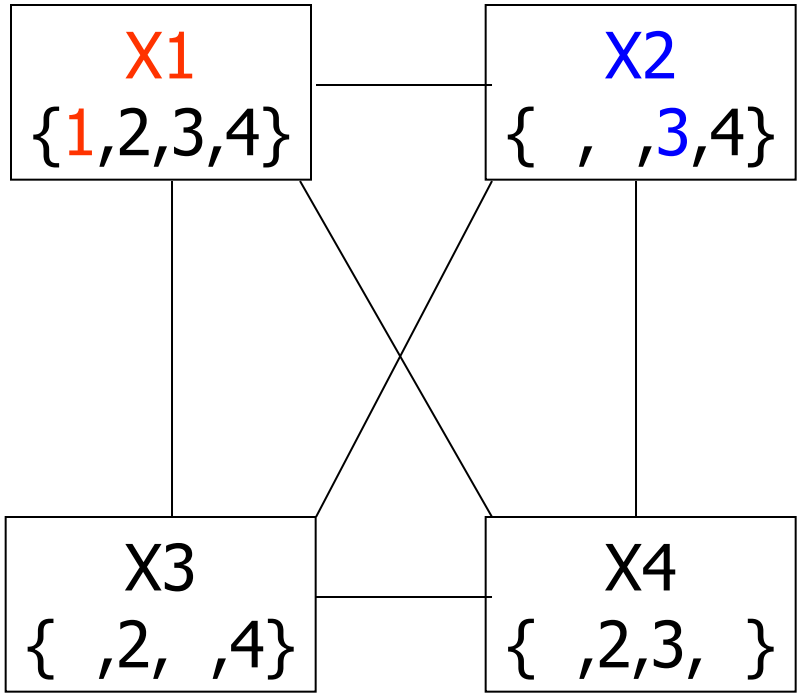




Dr. Abeer Mahmoud
(course coordinator)

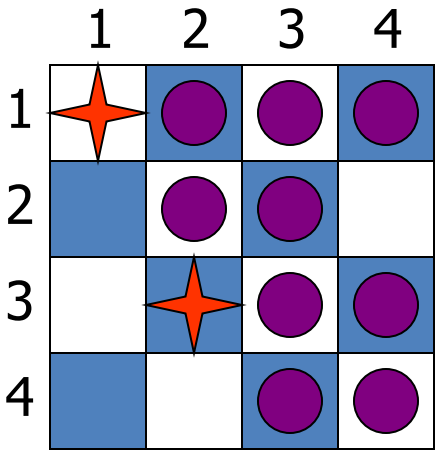
Example: 4-Queens Problem

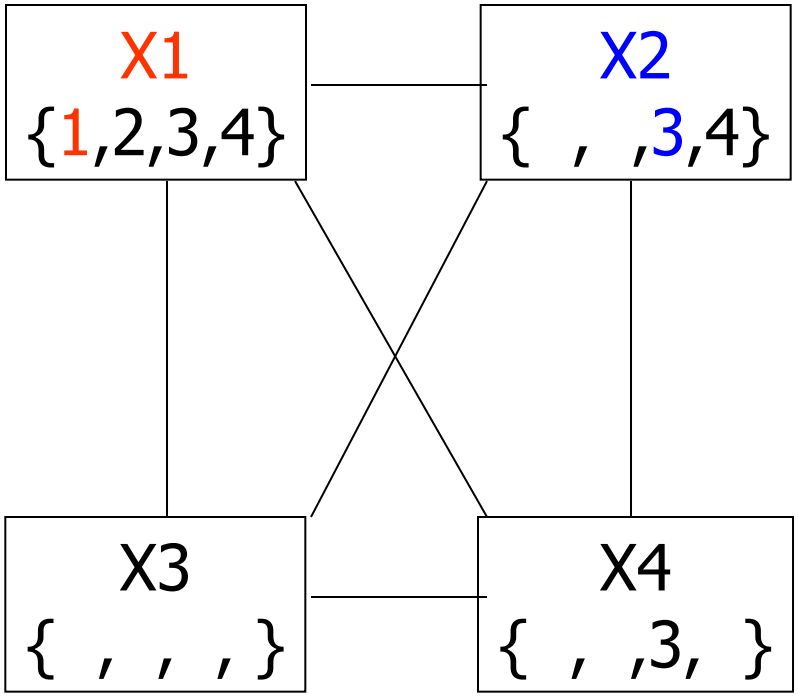




Dr. Abeer Mahmoud
(course coordinator)

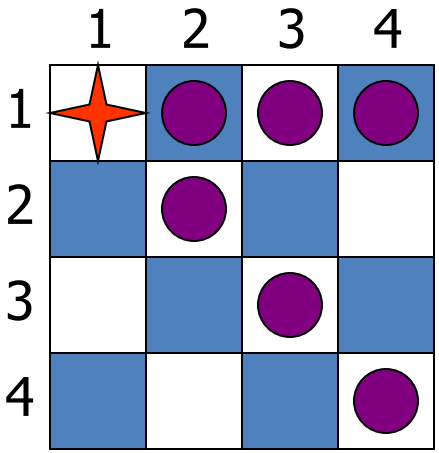
Example: 4-Queens Problem

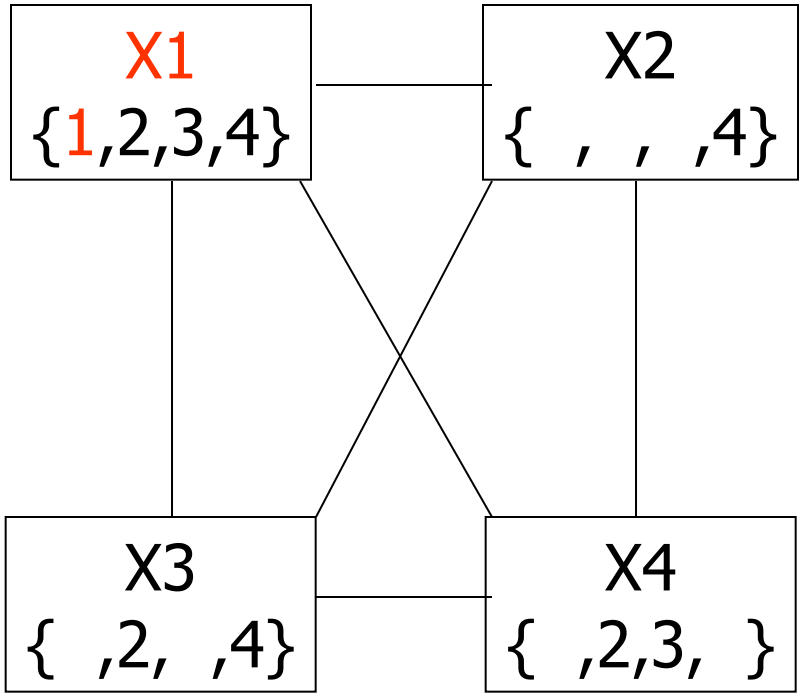




Dr. Abeer Mahmoud
(course coordinator)

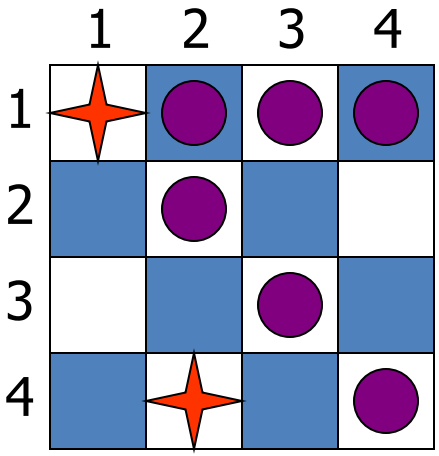
Example: 4-Queens Problem

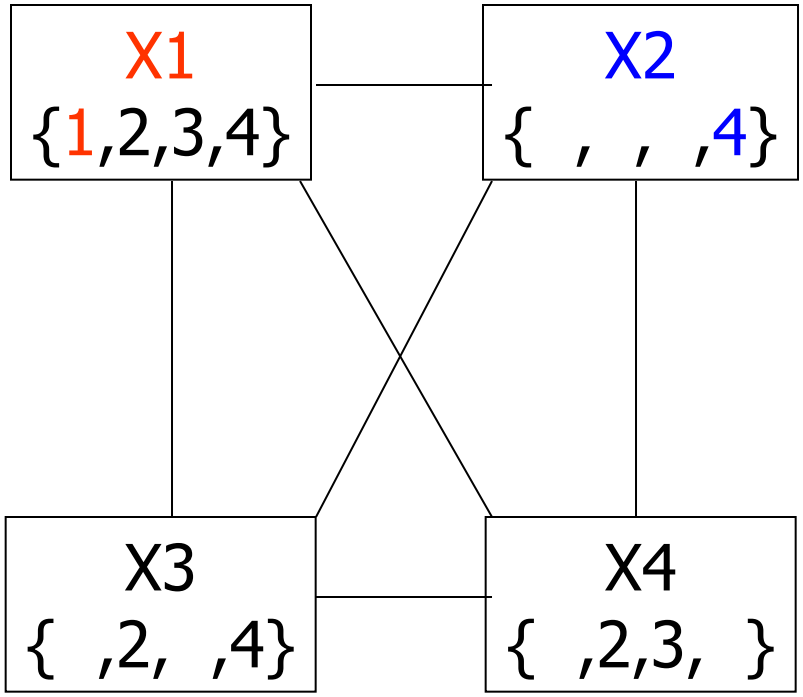




Dr. Abeer Mahmoud
(course coordinator)

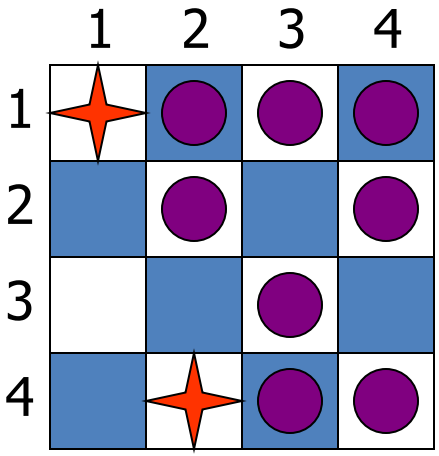
Example: 4-Queens Problem

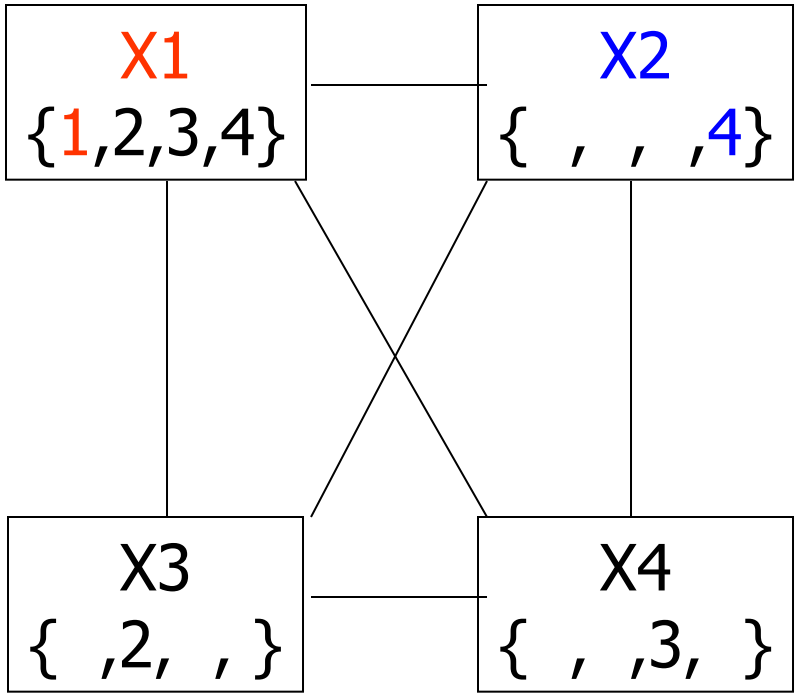




Dr. Abeer Mahmoud
(course coordinator)

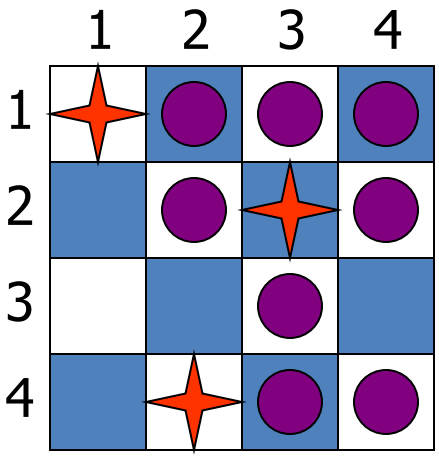
Example: 4-Queens Problem

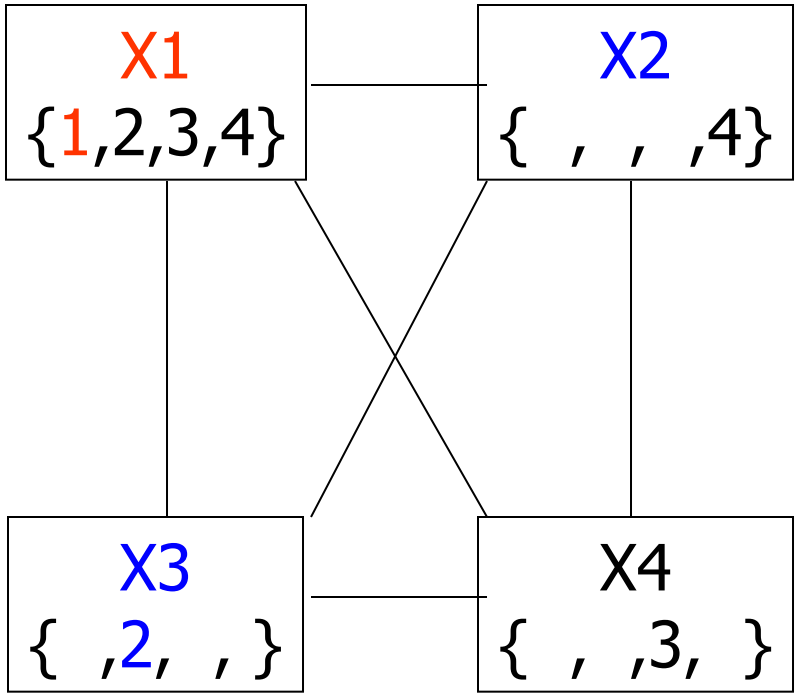




Dr. Abeer Mahmoud
(course coordinator)

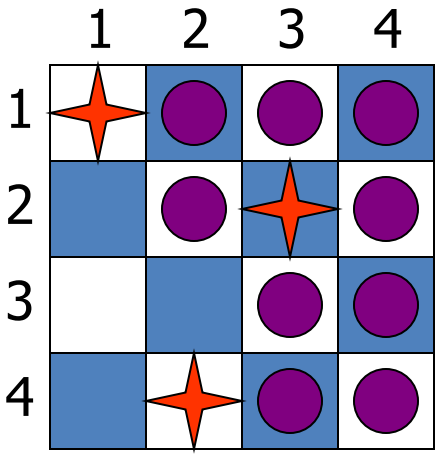
Example: 4-Queens Problem

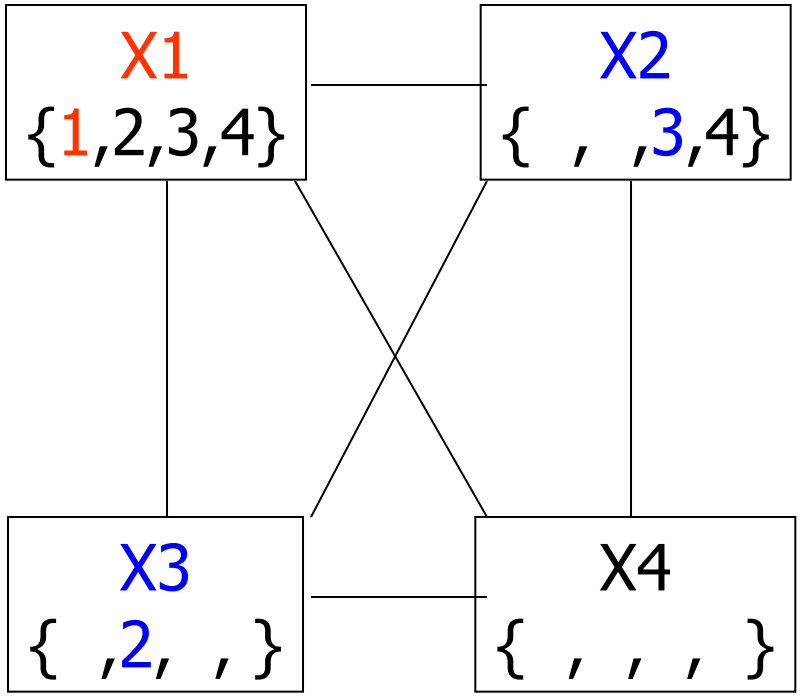




Dr. Abeer Mahmoud
(course coordinator)

Example: 4-Queens Problem

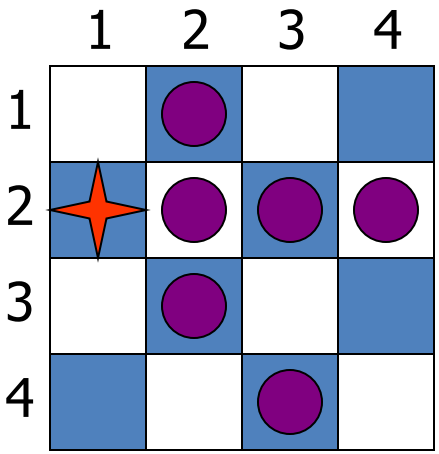


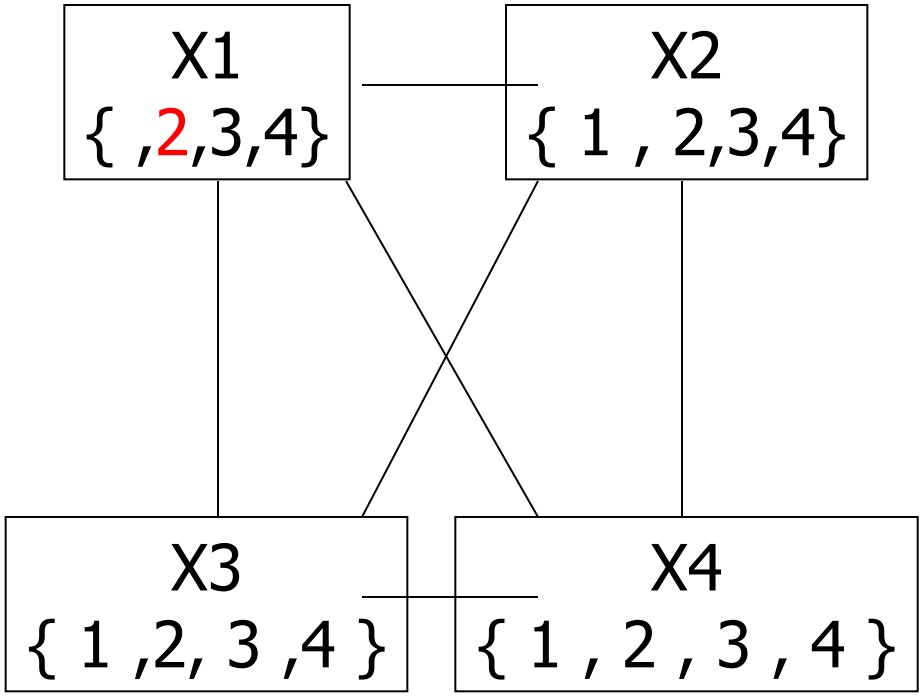


Dead End → Backtrack

Dr. Abeer Mahmoud
(course coordinator)

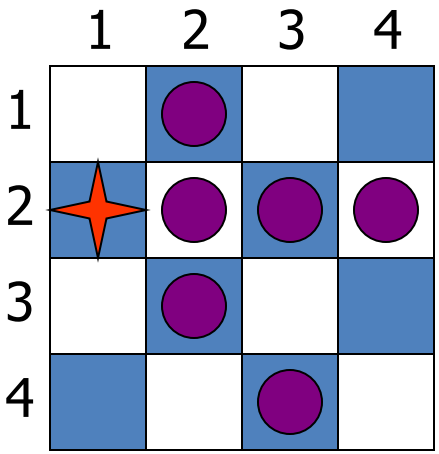
Example: 4-Queens Problem

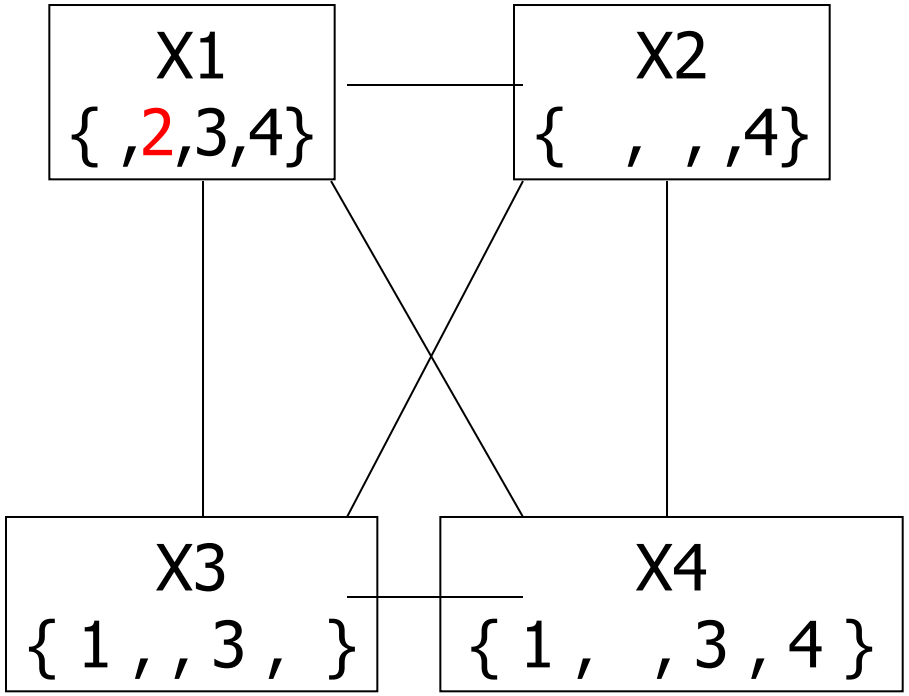




Dr. Abeer Mahmoud
(course coordinator)

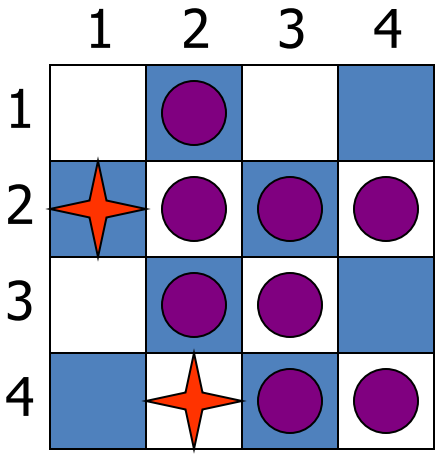
Example: 4-Queens Problem

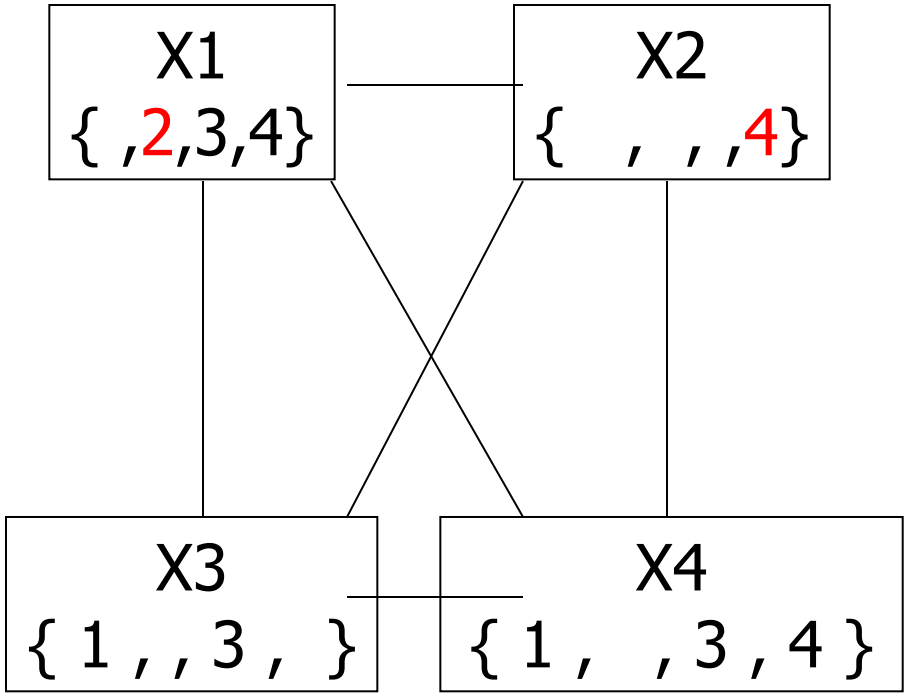




Dr. Abeer Mahmoud
(course coordinator)

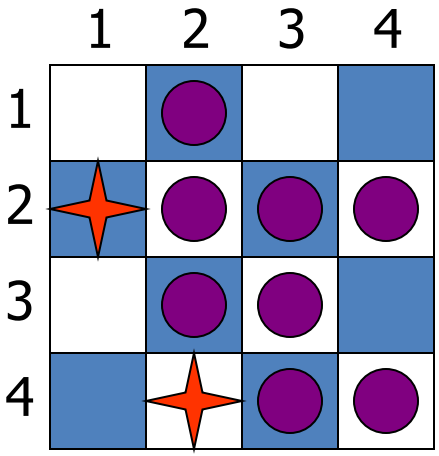
Example: 4-Queens Problem

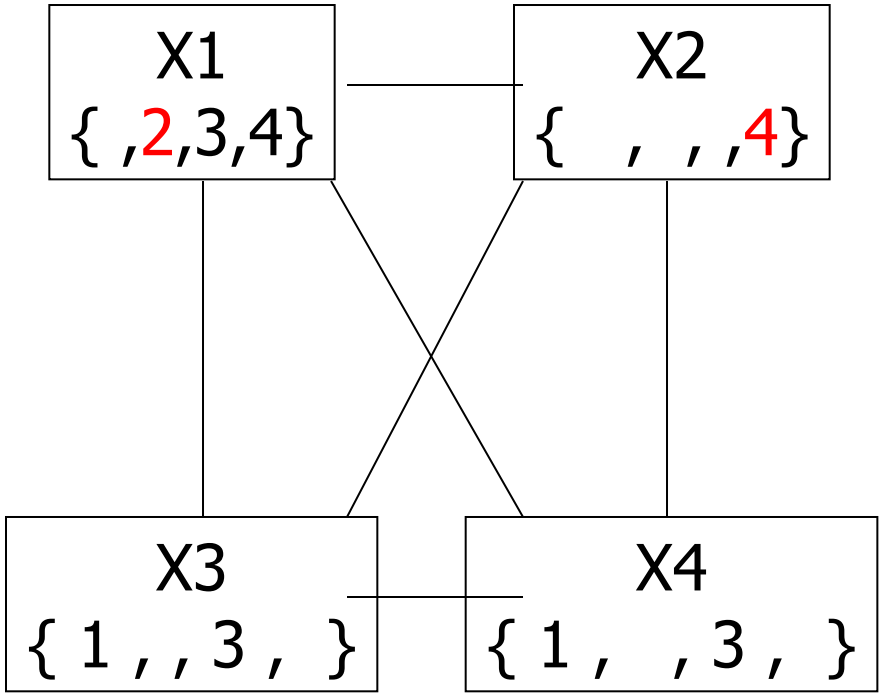




Dr. Abeer Mahmoud
(course coordinator)

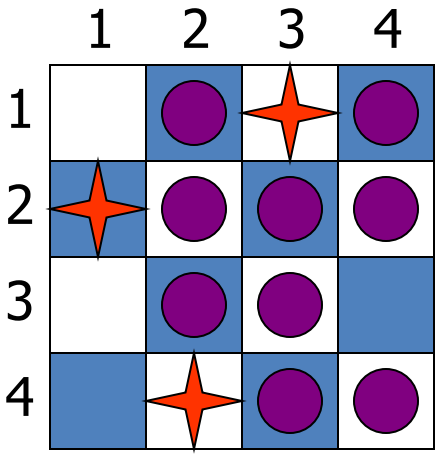
Example: 4-Queens Problem

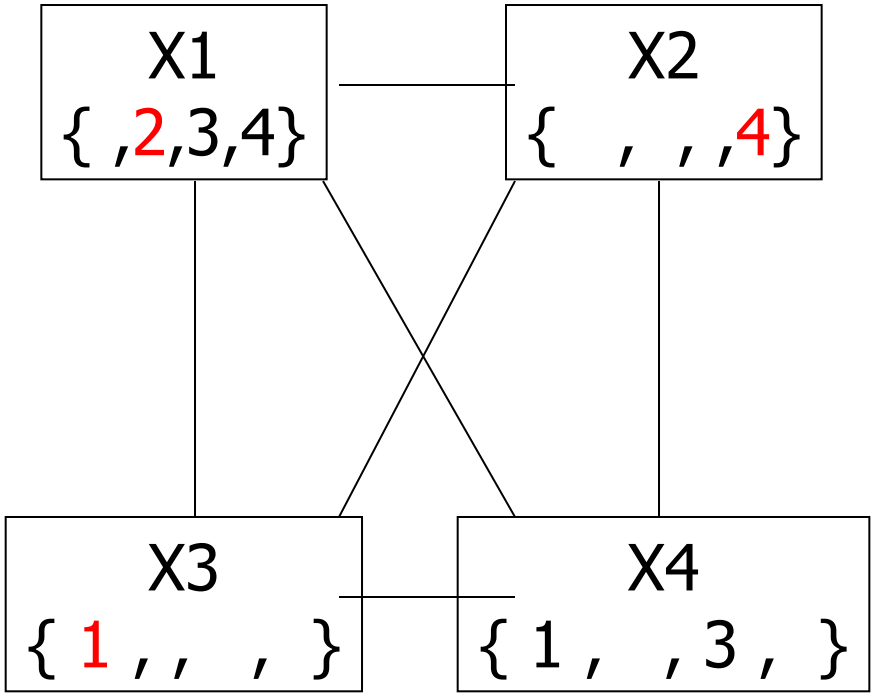




Dr. Abeer Mahmoud
(course coordinator)

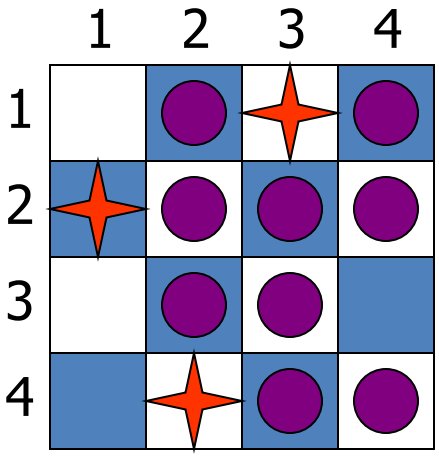
Example: 4-Queens Problem

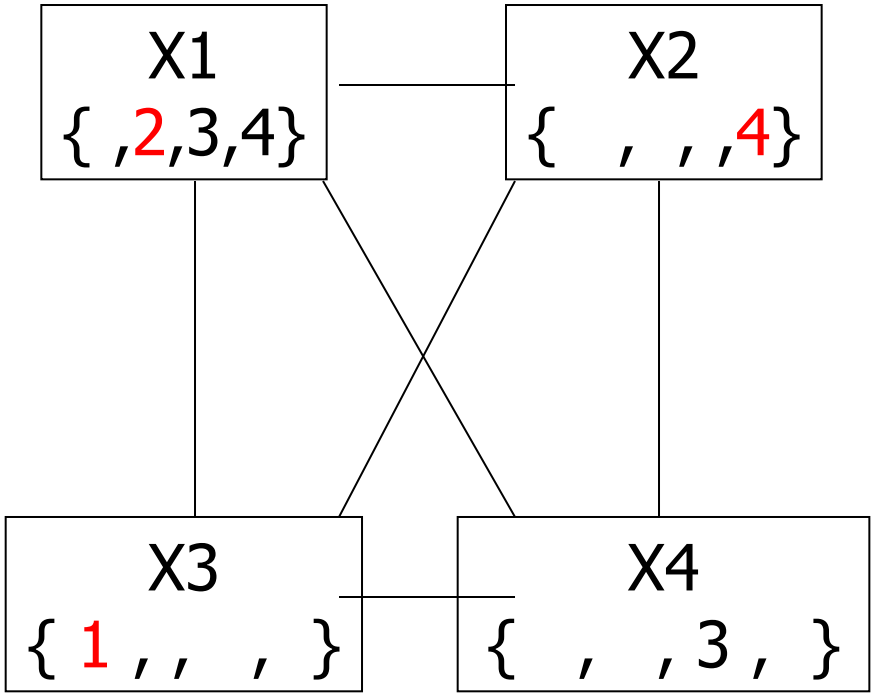




Dr. Abeer Mahmoud
(course coordinator)

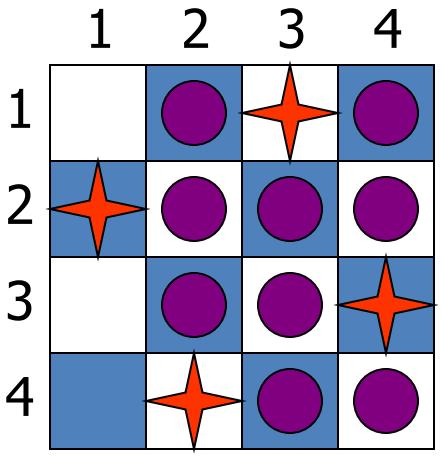
Example: 4-Queens Problem

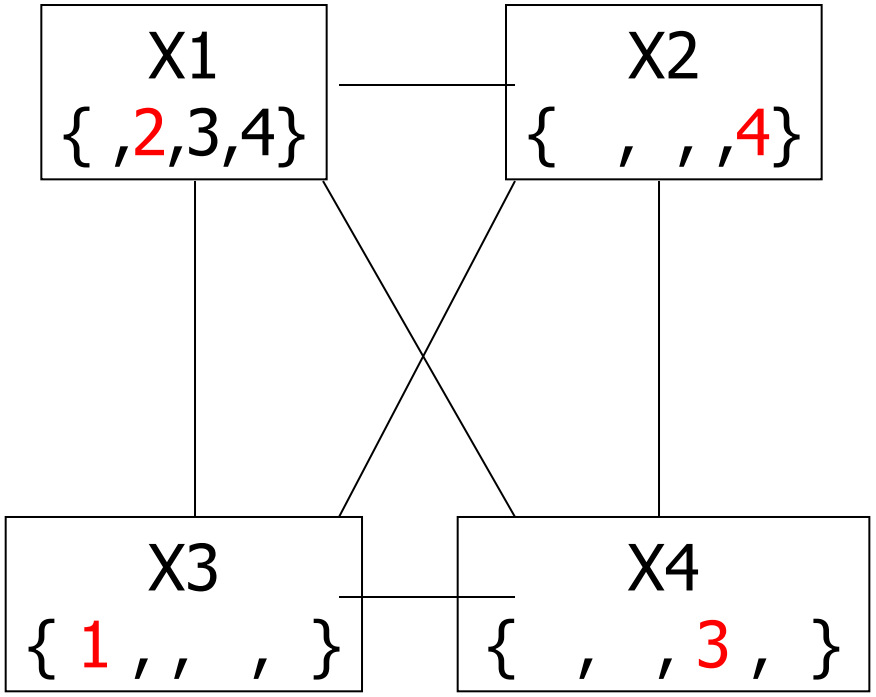




Dr. Abeer Mahmoud
(course coordinator)

Example: 4-Queens Problem





Solution !!!!

Dr. Abeer Mahmoud
(course coordinator)

Consistency Techniques

- removing inconsistent values from variables' domains
- graph representation of the CSP

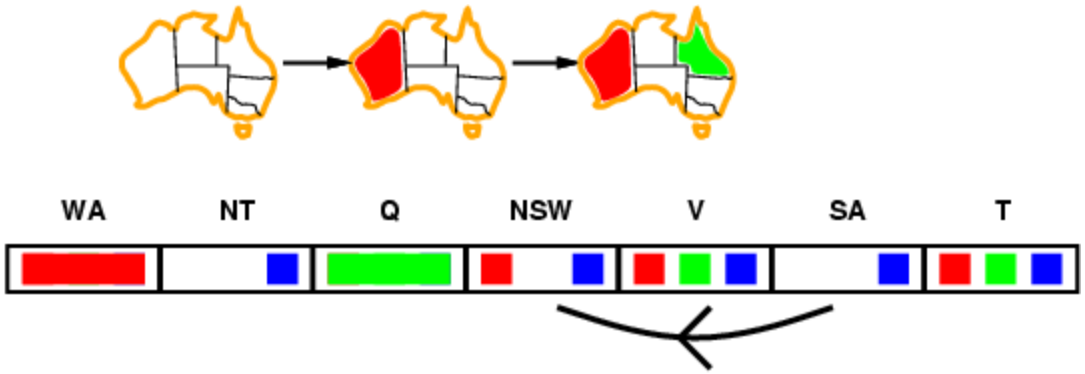
- **arc consistency (AC)**

Dr. Abeer Mahmoud
(course coordinator)

Arc consistency

- Simplest form of propagation makes each arc consistent
- $X \rightarrow Y$ is consistent iff

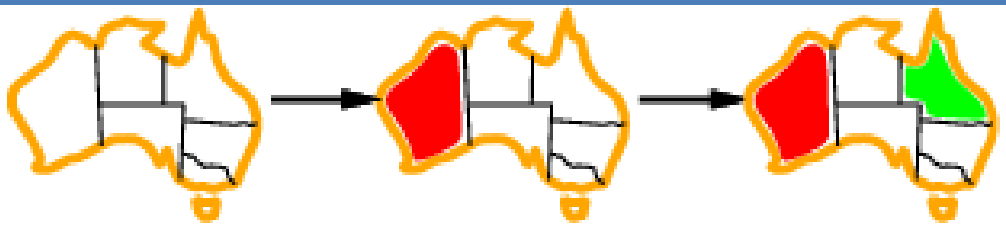
  for every value x of X there is some allowed y

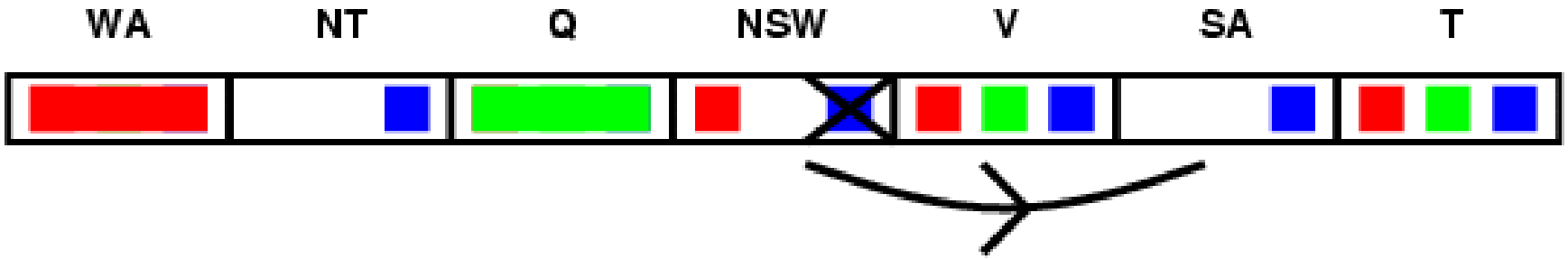


Arc from SA to NSW is consistent because SA (blue) allows NSW (red)

Dr. Abeer Mahmoud
(course coordinator)

Arc consistency



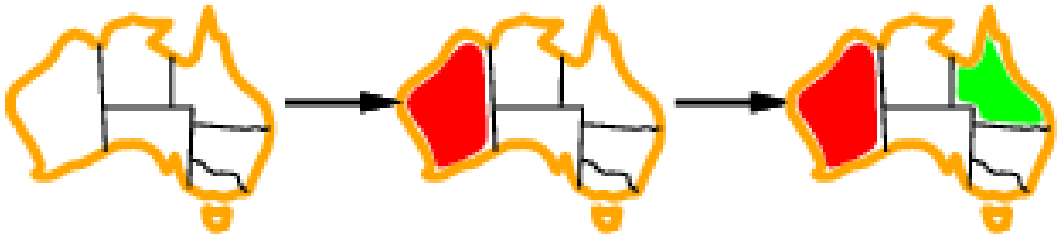


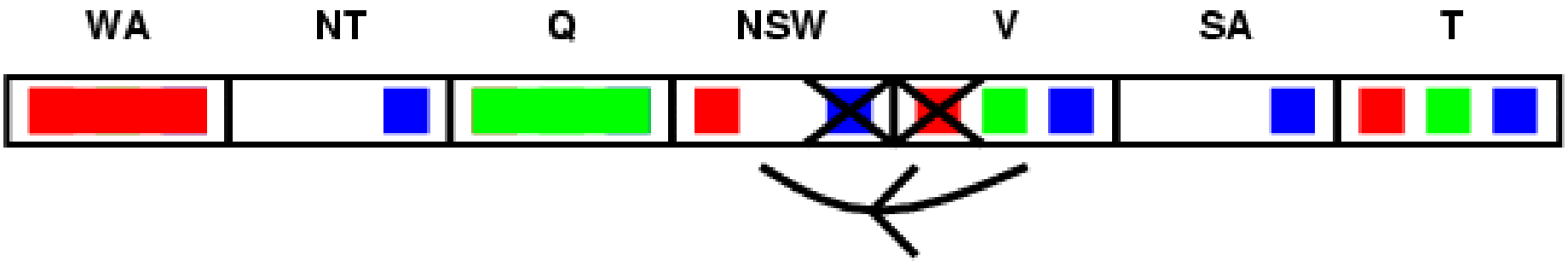
Arc from NSW to SA is **not consistent** because NSW (blue) does not allow SA (blue)

Can be made consistent by removing blue from NSW domain

Dr. Abeer Mahmoud
(course coordinator)

Arc consistency

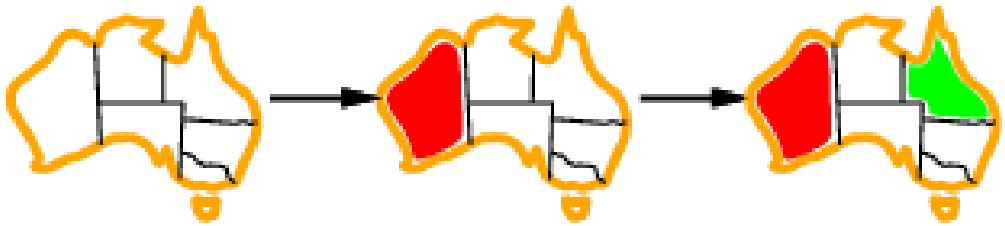


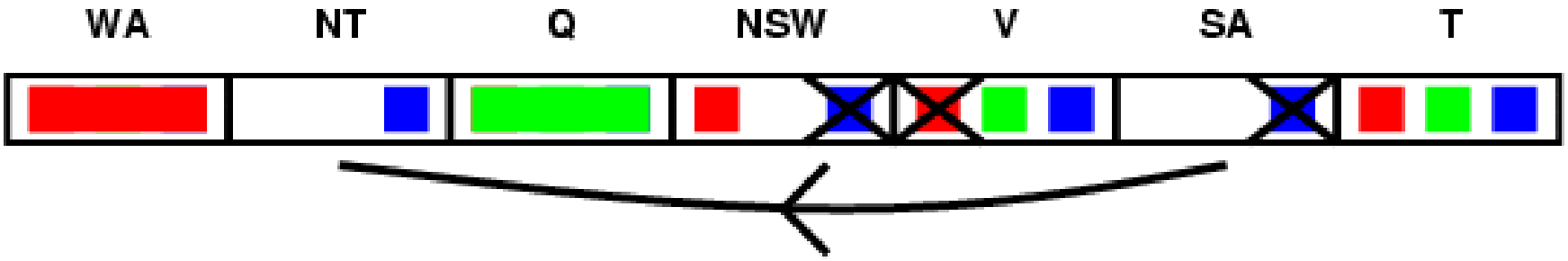


If X loses a value, neighbors of X need to be rechecked.

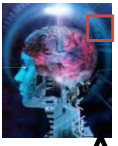
After removing blue from NSW, must check neighbors for consistency leading to removal of red from V.

Arc consistency



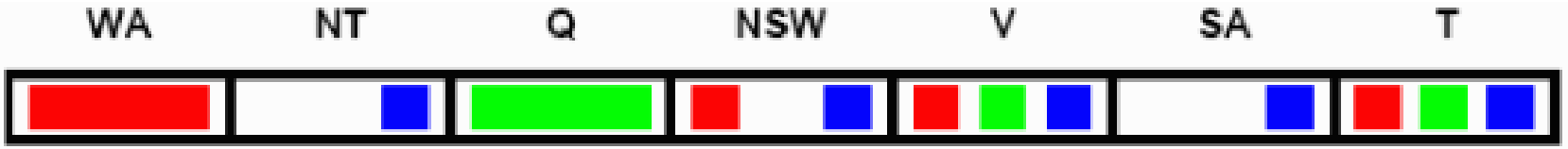


- If *X* loses a value, neighbors of *X* need to be rechecked
- Arc consistency detects failure earlier than forward checking
- Can be run as a preprocessor or after each assignment



After removing blue, SA domain is {}, pruning SA completely for the two assignments {NSW=red, Q=green} and forcing backtracking.

Dr. Abeer Mahmoud
(Course coordinator)



- Is the arc (WA,NSW) consistent?
- Is the arc (NSW,WA) consistent?
- If not, what should be the domain of NSW and WA?
- Is the arc (SA,T) consistent?
- is the arc (T, SA) consistent?

Dr. Abeer Mahmoud
(course coordinator)

Thank you

End of Chapter 6

Dr. Abeer Mahmoud
(course coordinator)