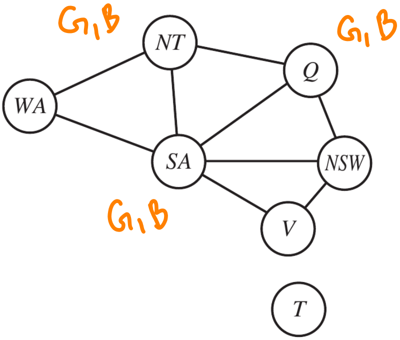
# Intelligent Backtracking




WA = Red

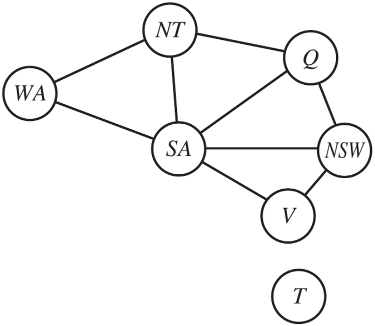
NSW = Red

T = G

SA = G

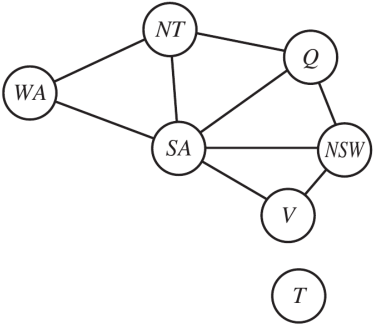
$$\text{Conf}(SA) = \underbrace{\{WA = R,\ NSW = R\}}$$

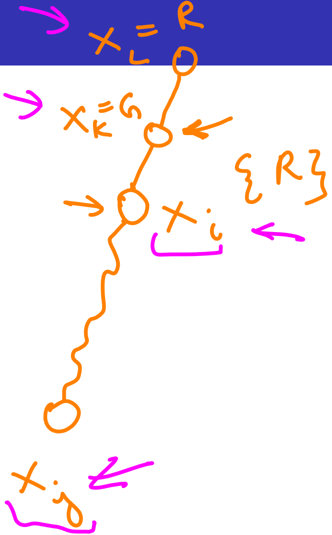
# Intelligent Backtracking




- Chronological backtracking
- Conflict-directed backjumping

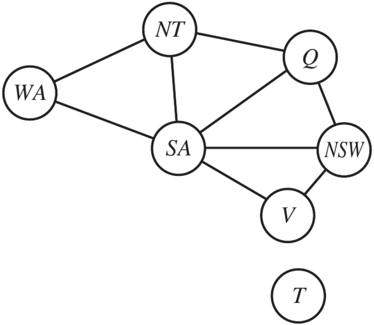
# Intelligent Backtracking



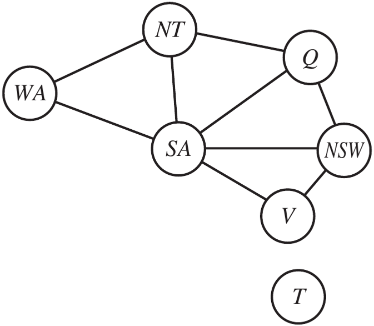

$\text{conf}(x_j) \cup \text{conf}(x_i)$

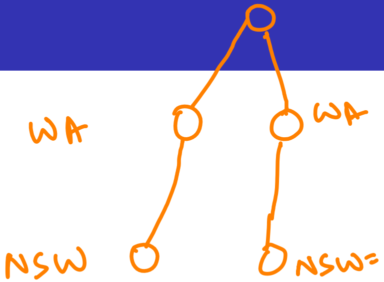
# Intelligent Backtracking



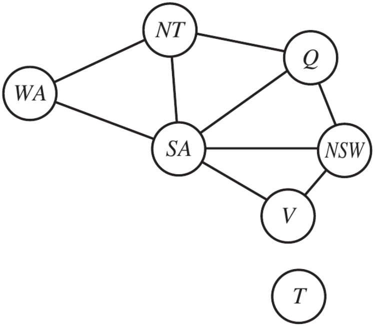

- $conf(X_i) \leftarrow conf(X_i) \bigcup conf(X_j) \setminus \{X_i\}$

# Constraint Learning



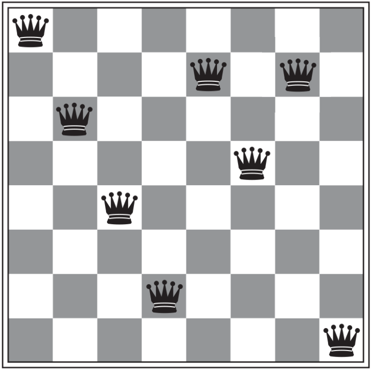



# Constraint Learning



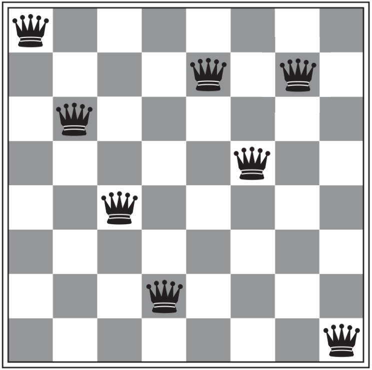

- Add a **new constraint**
- Maintain **no-good** set

# 8-queens Problem



- Goal: No queen must be in an attacking position.

# 8-queens Problem



- Goal: No queen must be in an attacking position.

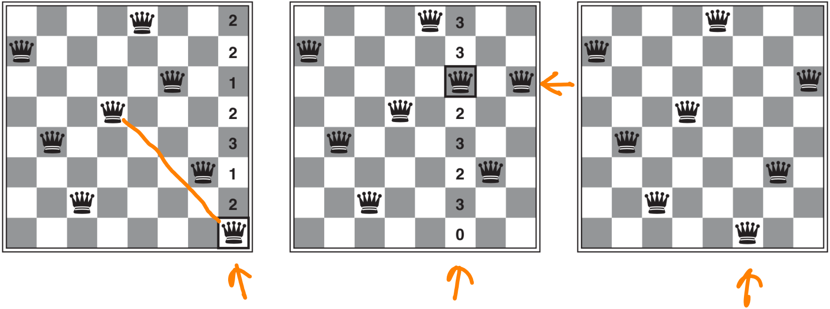
- CSP has eight variables corresponding to each queen.
- A queen can be moved to any position in the **same** column.
- So, $D_i = \{1, 2, \ldots, 8\}$.
- Assumption: Complete assignment (8 queens are on the board, one per column).
- $CONFLICTS(var, v)$ : Number of constraints that value $v$ violates.

# Min Conflicts : Local search

```
function MIN-CONFLICTS(csp, max_steps) returns a solution or failure
  inputs: csp, a constraint satisfaction problem
          max_steps, the number of steps allowed before giving up

  current ← an initial complete assignment for csp
  for i = 1 to max_steps do
      if current is a solution for csp then return current
      var ← a randomly chosen conflicted variable from csp.VARIABLES
      value ← the value v for var that minimizes CONFLICTS(var, v, current, csp)
      set var = value in current
  return failure
```

# Min Conflicts : Local search



# Min Conflicts : Local search

- Surprisingly effective for solving some CSPs.

# Min Conflicts : Local search

- Surprisingly effective for solving some CSPs.
- 1 Million-queens in $\underbrace{\text{50 steps.}}$

# Min Conflicts : Local search

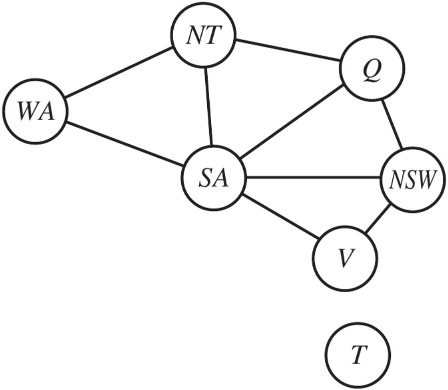
- Surprisingly effective for solving some CSPs.
- 1 Million-queens in 50 steps.
- Effective when solutions are densely distributed.

# Min Conflicts : Local search

- Surprisingly effective for solving some CSPs.
- 1 Million-queens in 50 steps.
- Effective when solutions are densely distributed.
- Constraint weighting

Minimize total weight of violated constraints

# Structure of Problems



# Structure of Problems

$3^{10} =$

$5 \quad\quad 5$

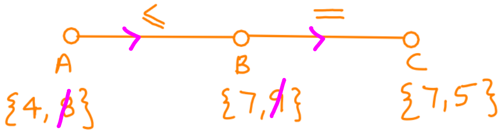
$3^{5} \times 2$

# Constraint graph having a tree structure

**Directed Arc Consistency** : A CSP is defined to be directed arc-consistent under an ordering of variables $X_1, X_2, \ldots, X_n$ if an only if every $X_i$ is arc-consistent with each $X_j$ for $j > i$.

# Constraint graph having a tree structure

**Directed Arc Consistency** : A CSP is defined to be directed arc-consistent under an ordering of variables $X_1, X_2, \ldots, X_n$ if an only if every $X_i$ is arc-consistent with each $X_j$ for $j > i$.




$\underbrace{A}, \underbrace{B}, C$

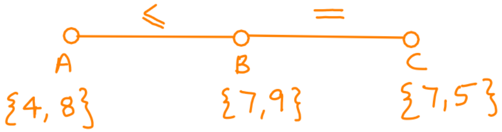
$(A, B)$

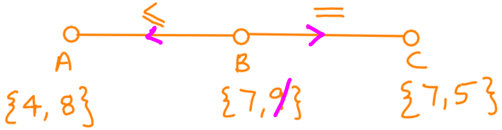
$(B, C)$

$A = 4, \; B = 7, \; C = 7$

# Constraint graph having a tree structure

**Directed Arc Consistency** : A CSP is defined to be directed arc-consistent under an ordering of variables $X_1, X_2, \ldots, X_n$ if an only if every $X_i$ is arc-consistent with each $X_j$ for $j > i$.



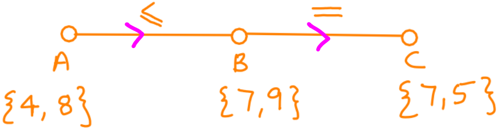

$\underbrace{B, A, C}$

$B = 7, A = 4, C = 7$

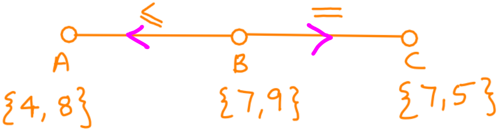
$O(nd)$

# Constraint graph having a tree structure

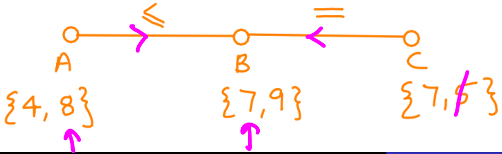
**Directed Arc Consistency** : A CSP is defined to be directed arc-consistent under an ordering of variables $X_1, X_2, \ldots, X_n$ if an only if every $X_i$ is arc-consistent with each $X_j$ for $j > i$.




A B C




B A C




A , C , B

A = 8, C = 7, B = ___