SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107
Accredited by NAAC-UGC with 'A' Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of AI &DS

Course Name – 23ADT201 ARTIFICIAL INTELLIGENCE

II Year / III Semester

UNIT 3

GAME THEORY

Topic: Constraint Satisfaction Problem and constraint propagation

CASE STUDY

Background

A university needs to schedule final exams for several courses. The objective is to assign a time slot and room for each exam while ensuring certain constraints are met, such as avoiding conflicts for students enrolled in multiple courses and accommodating room capacities.

Constraint Satisfaction Problem

- Constraint programming or constraint solving is about finding values for variables such that they satisfy a constraint(conditions).
- CSP = {V, D, C}

  Variables: V= {V1,...,Vn}

  Domain: D= { D1,D2,..Dn }

  Constraints: C= {C1,...,Ck}
- **Example:**

  Crossword puzzle

  Crypt-Arithmatic problem

  Map colouring problem

Crypt-Arithmetic problem

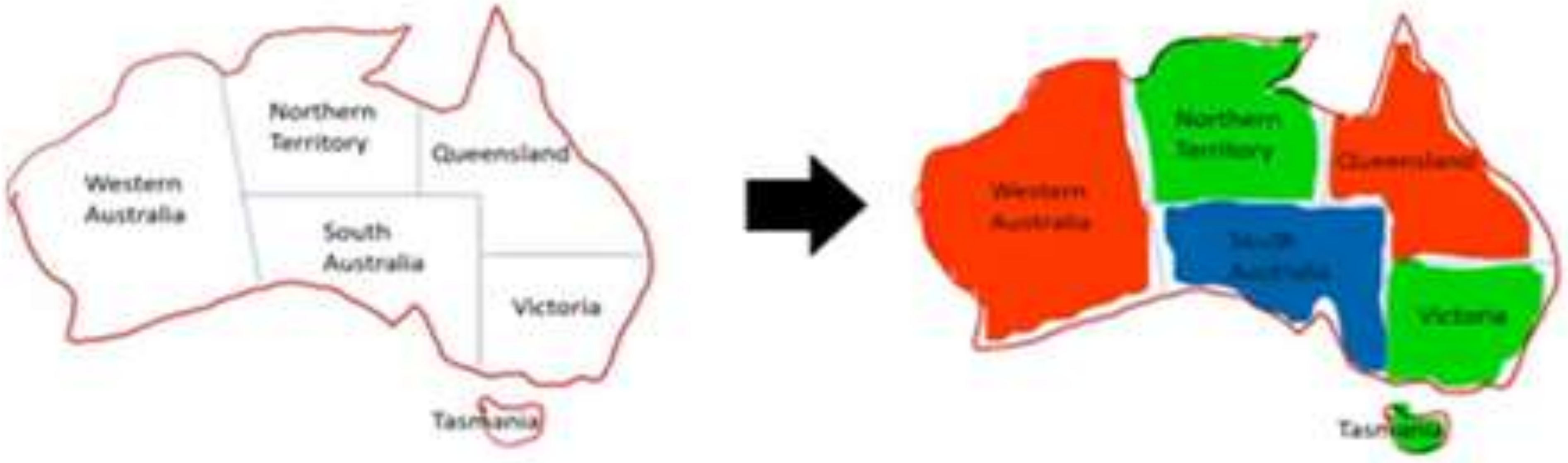
- Crypt-Arithmetic problem is a type of **encryption problem** in which the written message in an alphabetical form which is easily readable and understandable is converted into a numeric form which is neither easily readable nor understandable.

Constraints

- Every Character must have a unique value.
- Digits should be from 0-9 only
- Starting character of number can not be zero.
- Cryptarithmetic problem will have only one solution.
- Addition of number with itself is always even
- In case of addition of two numbers, if there is carry to next step then, the carry can only be 1

Map Colouring

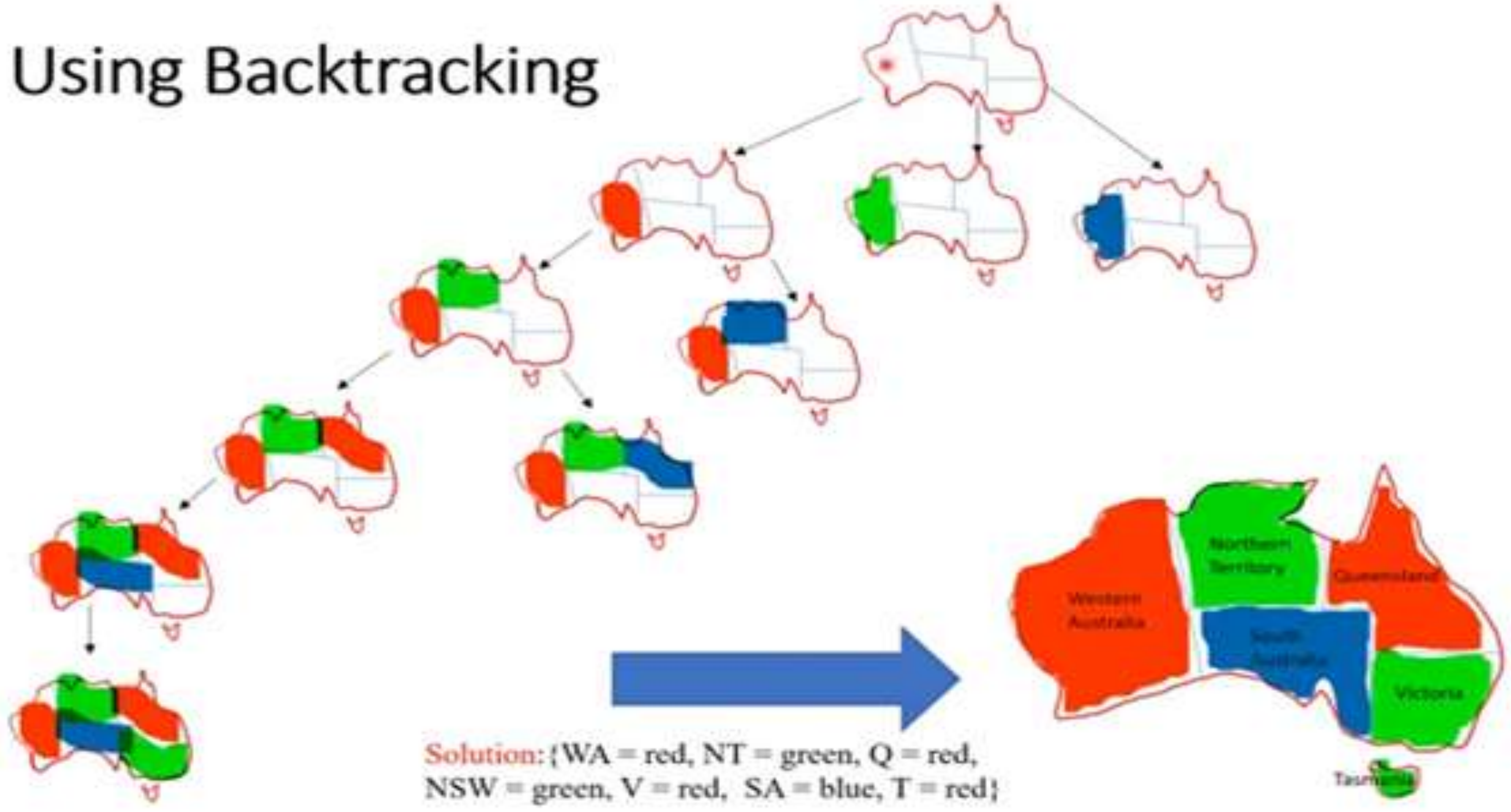
- Two adjacent regions cannot have the same color no matter whatever color we choose.
- The goal is to assign colors to each region so that no neighboring regions have the same color.



Map Colouring-Example

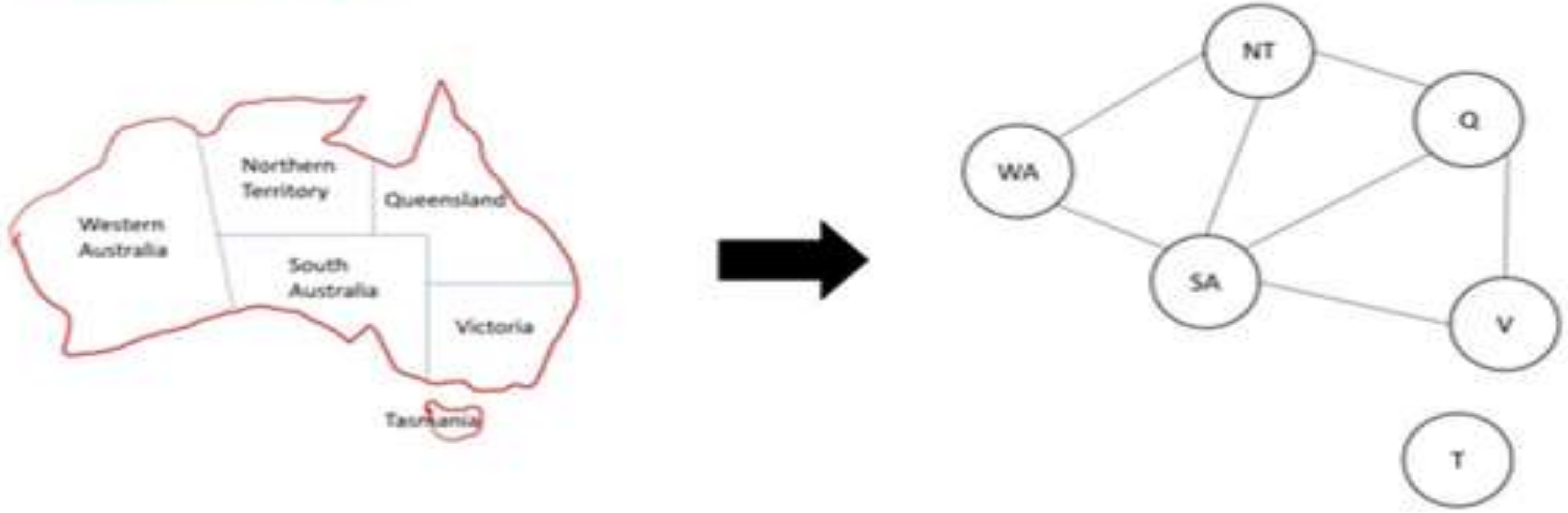
- Color the following map using red, green, and blue such that adjacent regions have different colors

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

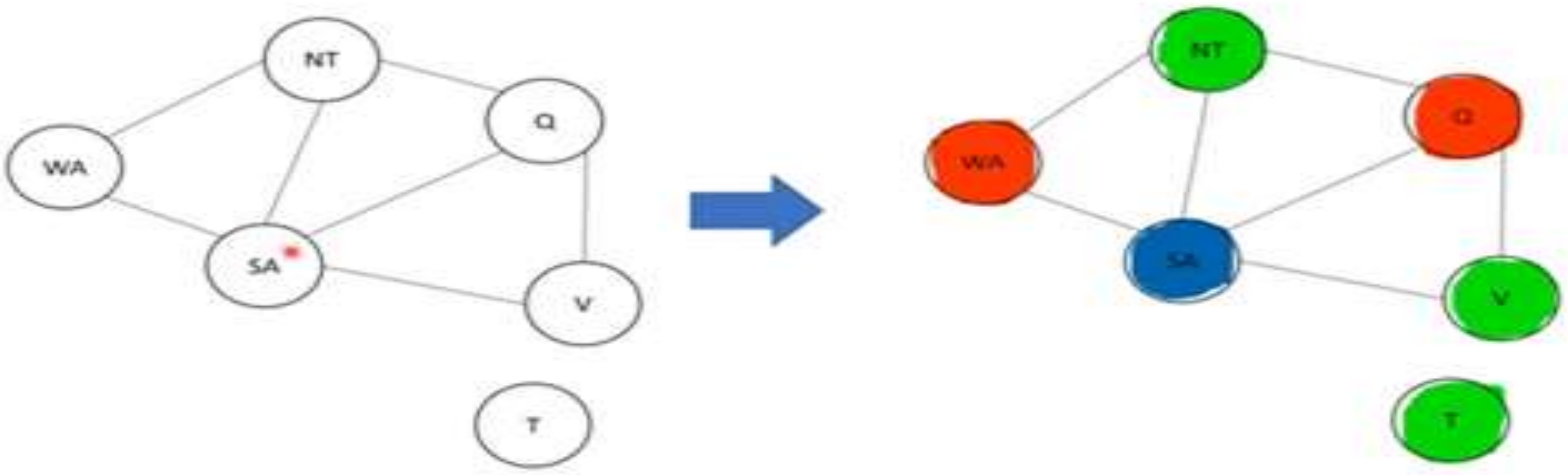

Using Backtracking
Northern Territory
Queensland
Western Australia
South Australia
Victoria
Tasmania
Solution: {WA = red, NT = green, Q = red,
NSW = green, V = red, SA = blue, T = red}

Constraint Graph

- Constraint graph: nodes are variables, arcs are constraints



Constraint Graph



Variations on CSP Formalism
Variable Types

- Discrete Variables
  - Finite Domains/ Discrete Domains
    - Simplest CSPs define variables that have discrete, finite domains.
      - E.g. map coloring, 8-queens.
  - Infinite Domains
    - Domains can be infinite
      - e.g. set of integers or strings.
      - E.g. Job scheduling
      - Infinitely many solutions.
      - Thus need to use a constraint language. T1+D1<=T2
  - Continuous Domain
      - e.g. problems from temporal reasoning->time related
      - e.g. problems from spatial reasoning-> space related
      - Hubble telescope

Variations on CSP Formalism
Variable Types

E.g., for a job scheduling problem,

- might have T 1 + d 1 <= T 2
  - where T 1 and T 2 are start times for two tasks and
  - d 1 is the duration of the first one

Variations on CSP Formalism

Constraint Types

- The simplest type is the unary constraint, which restricts the value of a single variable:
  - <(SA), SA ≠ green>
- A binary constraint relates two variables:
  - <(SA, NSW), SA ≠ NSW>
  - A binary CSP is one with only binary constraints; it can be represented as a constraint graph.
- Higher-order constraints are possible.
  - E.g. Between (X, Y, Z) that specifies that Y is between X and Z

Variations on CSP Formalism
Constraint Types

- A constraint involving an arbitrary number of variables is called a global constraint.
  - One of the most common global constraints is *Alldiff*.
- Many real-world CSPs include preference constraints indicating which solutions are preferred:
  - constraint optimization problem (COP)

Convert n-ary constraint to binary one

- Example: How a single ternary constraint such as “A + B = C” can be turned into three binary constraints?
  - Using an auxiliary variable:
    - First introduce a new variable AB and its domain.
    - Then create appropriate relations between the new variables and old ones.

Constraint Propagation

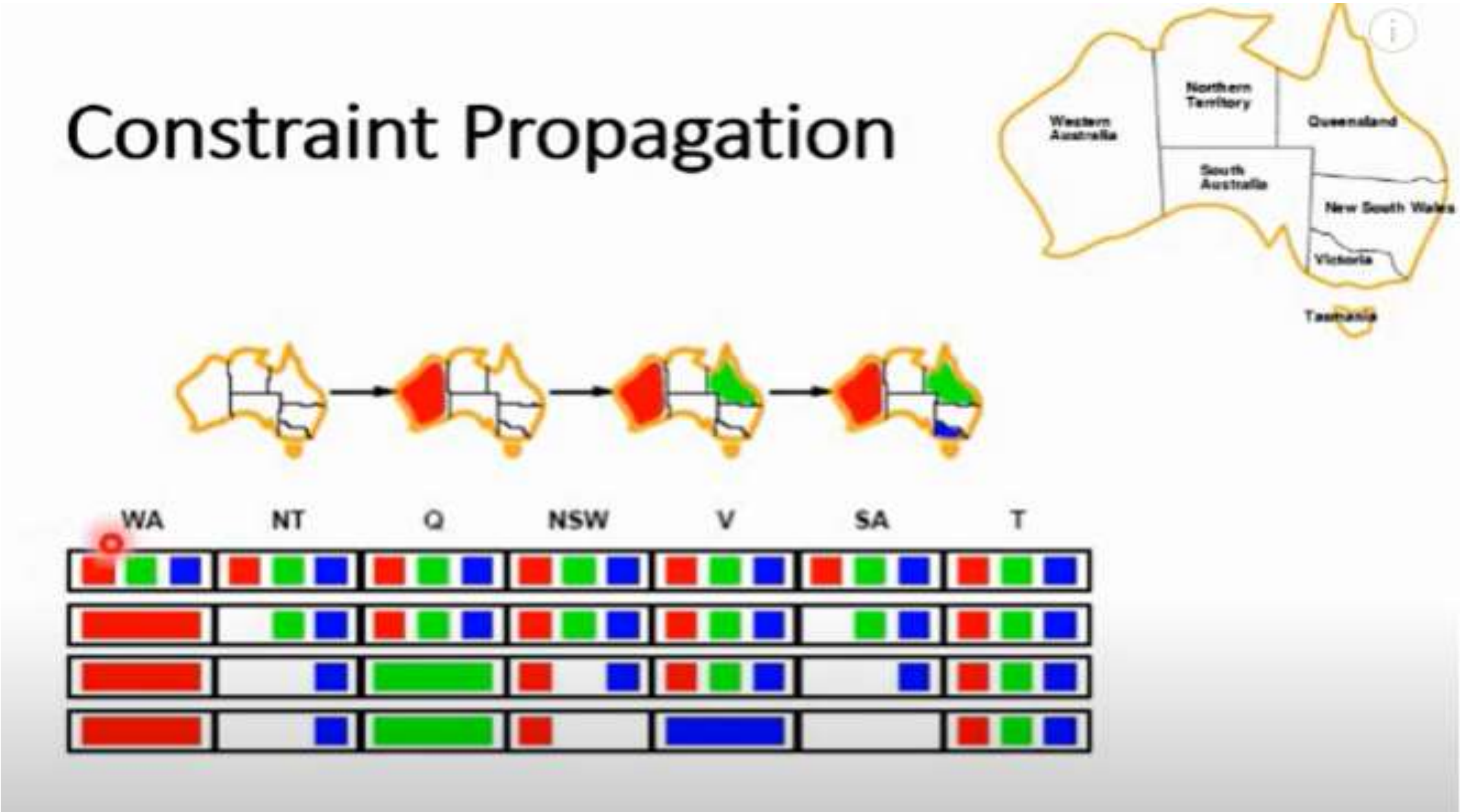
- In regular state-space search, an algorithm can do only one thing: search.
- In CSPs there is a choice:
  - an algorithm can search (choose a new variable assignment from several possibilities) or
  - do a specific type of inference called constraint propagation
- Constraint Propagation
  - using the constraints to reduce the number of legal values for a variable,

Constraint Propagation

X = {WA,NT,Q,NSW, V,SA, T}

Di = {red , green, blue}

C = {SA ≠ WA,SA ≠ NT,SA ≠ Q,SA ≠ NSW,SA ≠ V, WA ≠ NT,NT ≠ Q,Q ≠ NSW,NSW ≠ V }


Constraint Propagation
Western Australia
Northern Territory
Queensland
South Australia
New South Wales
Victoria
Tasmania
WA
NT
Q
NSW
V
SA
T

Local Consistency

- Node consistency
- Arc consistency
- Path consistency
- k-consistency

Local Consistency
Node Consistency

- A single variable is node-consistent if all the values in the variable's domain satisfy the variable's unary constraints.

X={V}

D={1,2,3,4}

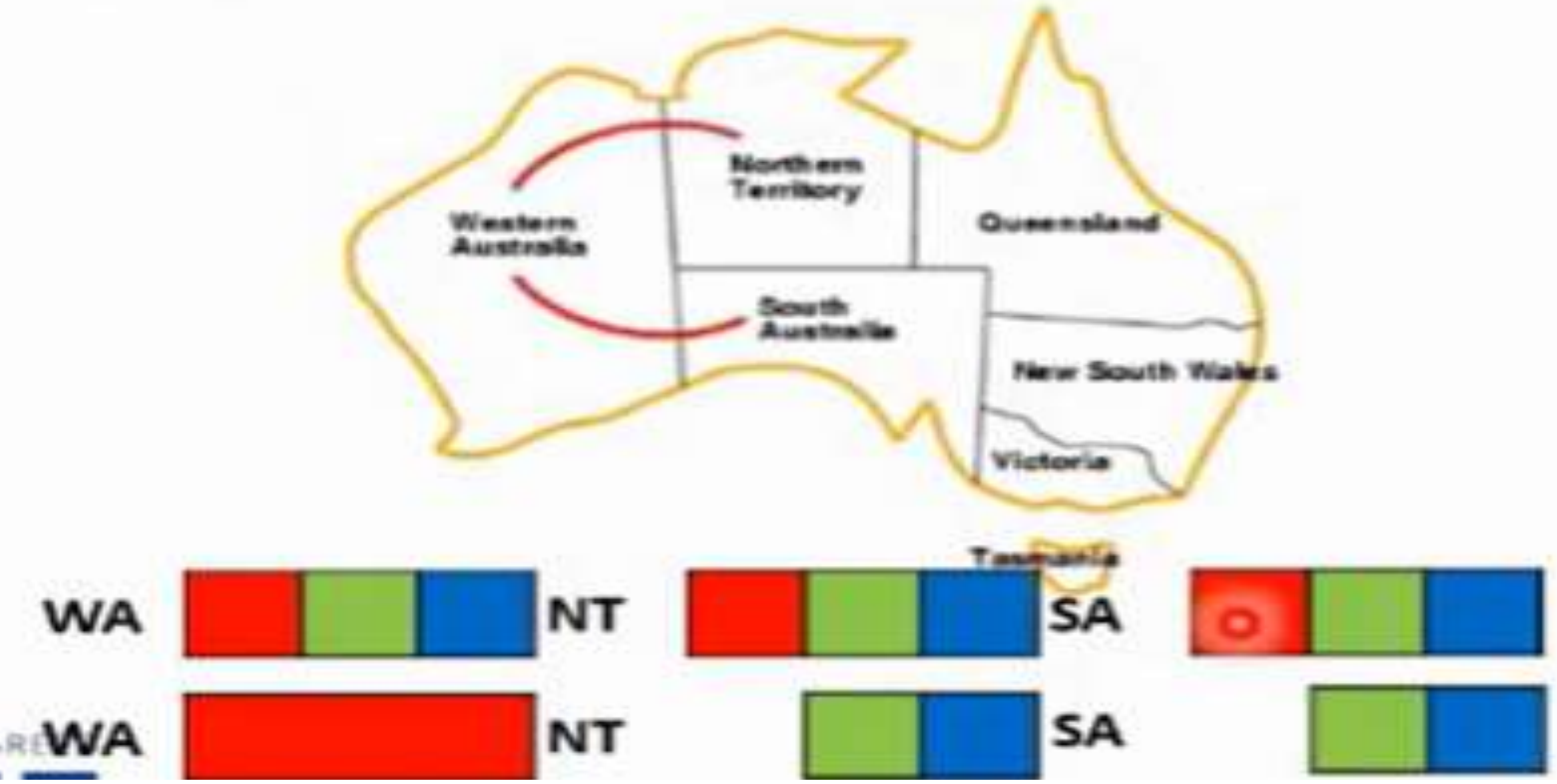
C={V<=3}

ΔD={1,2,3}

- A network is node-consistent if every variable in the network is node-consistent.
- It can be done as a preprocessing step:
  - eliminate all inconsistent values from variables' domains.

Local Consistency
Arc Consistency

- A variable in a CSP is arc-consistent if every value in its domain satisfies the variable's binary constraints.



Arc Consistency

- More formally:
  - Xi is arc-consistent with respect to another variable Xj if
  - for every value in the current domain Di there is some value in the domain Dj that satisfies the binary constraint on the arc (Xi , Xj).
- A network is arc-consistent if every variable is arc consistent with every other variable.

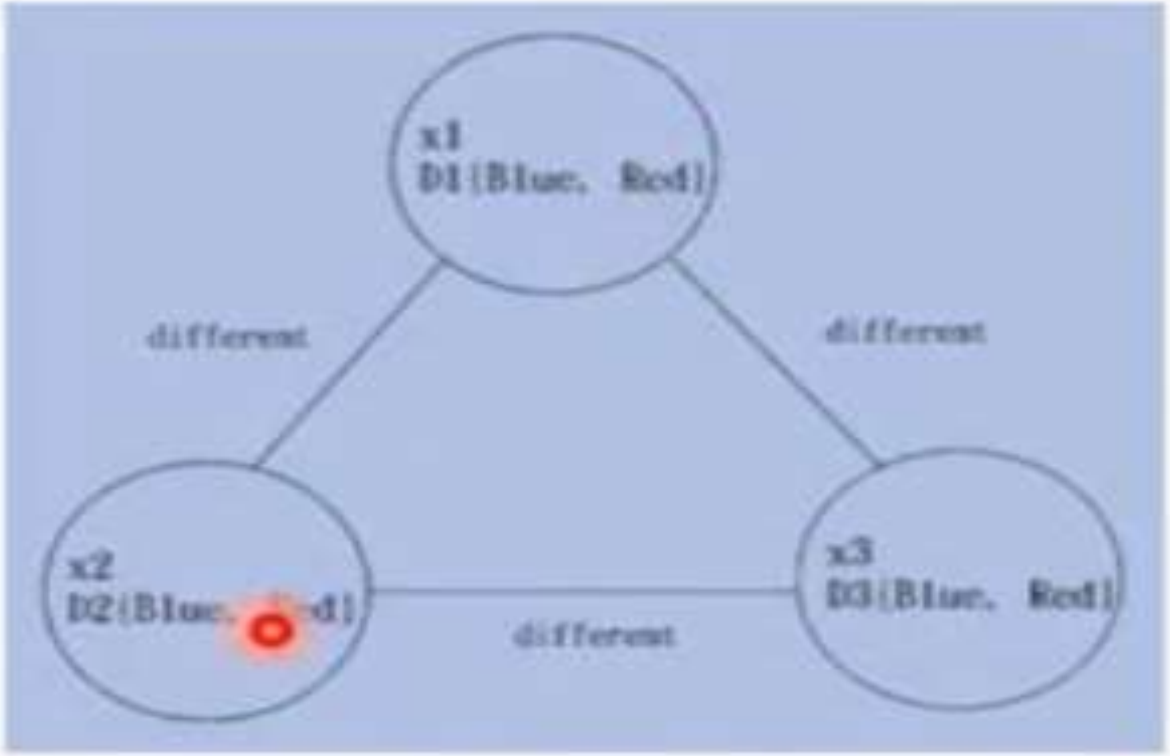
- Was there any variable in map coloring problem that was **node** consistent?

Local Consistency
Path Consistency

- For many kind of problems, arc consistency fails to make enough inferences. Consider the map-coloring problem on Australia, but with only two colors allowed, red and blue...

Local Consistency
Path Consistency

- A two-variable set {X1 , X2 } is path-consistent with respect to a third variable X3 if, for every assignment {X1 = blue, X2 = Red} consistent with the constraints on {X1 , X2 }, there is an assignment to X3 that satisfies the constraints on {X1 , X3 } and {X3 , X2}



- This is called path consistency because one can think of it as looking at a path from X1 to X2 with X3 in the middle.

Local Consistency
Path Consistency (Example)

- We will make the set {WA, SA} path consistent with respect to NT .
- We start by enumerating the consistent assignments to the set.
  - In this case, there are only two: {WA = red , SA = blue} and {WA = blue, SA = red}.
- We can see that with both of these assignments NT can be neither red nor blue.
  - We eliminate both assignments, and we end up with no valid assignments for {WA, SA}.

Local Consistency
K Consistency

- A CSP is k-consistent if, for any set of k – 1 variables and for any consistent assignment to those variables
  - A consistent value can always be assigned to any kth variable.
    - 1-consistency: Node consistency.
    - 2-consistency: Arc consistency.
    - 3-consistency: Path consistency.
- A CSP is strongly k-consistent if it is k-consistent and is also (k−1)-consistent, (k−2)-consistent, . . . all the way down to 1-consistent.

Local Consistency
Find a Soluction

- Suppose there is a CSP with n nodes and strongly n-consistent. Now a solution can be found in this way:
  - Choose a consistent value for X_1 .
  - It is guaranteed to be able to choose a value for X_2 because the graph is 2-consistent, for X_3 because it is 3-consistent, and so on.
- It is guaranteed to find a solution in time $O(n^2d)$:
  - For each variable X_i , the algorithm only searches through the d values in the domain to find a value consistent with $X_1, \ldots, X_{i-1}$.
- Any algorithm for establishing n-consistency must take time exponential in n in the worst case.

THANK YOU