Constraint Satisfaction Problems (CSPs)

Introduction

CPSC 322 Lecture 10

Something to think about

Suppose you were tasked with coming up with the UBC final exam schedule this term

- Can we set every exam to happen at the same time and in the same place?
- What constraints would you have to take into account?
- *You could set the exam for course* ***C*** *consisting of the set of students* ***S*** *on day* ***D*** *at time* ***T*** *in location* ***L*** *if…*
  - Do all the constraints involve the same “variables”?

Learning Goals for today's class

- Define **possible worlds** in term of variables and their domains.
- Compute number of possible worlds on real examples
- Specify constraints to represent real world problems differentiating between:
  - Unary and k-ary constraints
  - List vs. function format.
- Verify whether a possible world satisfies a set of constraints (i.e., whether it is a model, a solution)

Lecture Overview

- **Generic Search vs. Constraint Satisfaction Problems**
- Variables
- Constraints
- CSPs

Standard Search

To learn about **search** we have used it as the *reasoning strategy* for a **simple goal-driven planning agent**…..

vacuum cleaner
8-puzzle
delivery robot

Solution?

start -> path -> goal

Standard search problem: An agent can solve a problem by searching in a space of states

- state is a "black box" – any arbitrary data structure that supports **three problem-specific routines**

(1) neighbors(n) (2) heuristic(n) (3) goal(n)

Standard Search vs. Specific R&R systems

Constraint Satisfaction (Problems):

- State
- Successor function
- Goal test
- Solution

Planning :

- State
- Successor function
- Goal test
- Solution

Inference

- State
- Successor function
- Goal test
- Solution

coming weeks

R&R systems we'll cover in this course

		Environment	
Problem		Deterministic	Stochastic
Static	Constraint Satisfaction	*Variables + Constraints* Search Arc Consistency Local Search	
	Query	*Logics* Search	*Bayesian (Belief) Networks* Variable Elimination
Sequential	Planning	*STRIPS* Search	*Decision Networks* Variable Elimination

Representation
Reasoning Technique

Lecture Overview

- **Generic Search vs. Constraint Satisfaction Problems**
- Variables/Features
- Constraints
- CSPs

Variables/Features, domains and Possible Worlds

- Variables / features
  - we denote variables using capital letters
  - each variable V has a domain *dom*(V) of possible values

- Variables can be of several main kinds:
  - Boolean: |*dom*(V)| = 2
  - Finite: the domain contains a finite number of values
  - Infinite but Discrete: the domain is countably infinite
  - Continuous: e.g., real numbers between 0 and 1

  } not in this course

- Possible world: a complete assignment of values to a set of variables

Example (from a previous lecture)

Mars Explorer Example

Weather	{S, C}
Temperature	[-100, 10]
Longitude	[0, 359]
Latitude	[-90,90]

One possible world (state) {S, -30, 320, 83}

Number of possible (mutually exclusive) worlds (states)

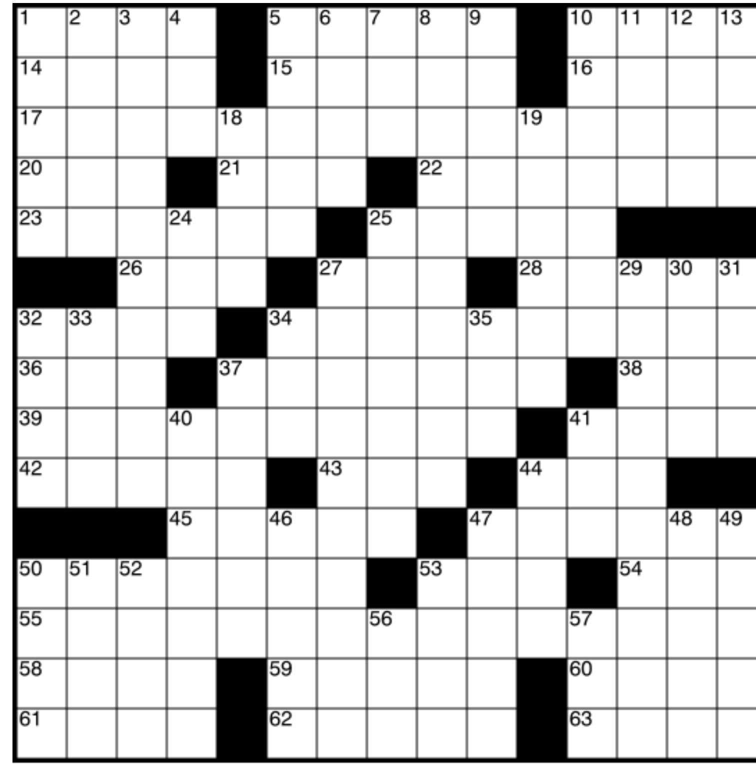
2 x 111 x 360 x 181

Product of cardinality of each domain

… always *exponential* in the number of variables

Examples

- Crossword Puzzle:
  - variables are words that have to be filled in *(63 in this case)*
  - domains are valid English words of required length
  - possible worlds: all ways of assigning words

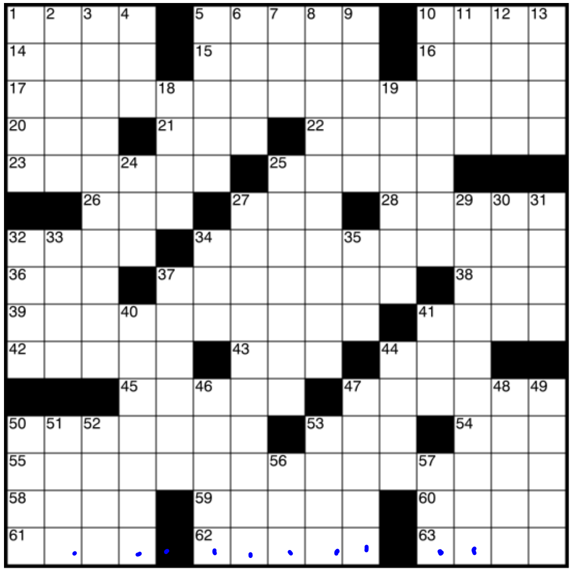


- *Number of English words? (~150,000)*
- *Number of words of length k ? (~15,000)*
- *So, (roughly) how many possible worlds?*

A. 15,000*63 B. $63^{15,000}$ C. $15,000^{63}$ D. $1,563^{63}$ E. 42

More Examples

- Crossword 2:
  - variables are cells (individual squares) in the 15x15 grid
  - domains are letters of the alphabet
  - possible worlds: all ways of assigning letters to cells
    - *So, how many possible worlds?*



- Sudoku:
  - variables are empty cells
  - domains are numbers between 1 and 9
  - possible worlds: all ways of assigning numbers to cells

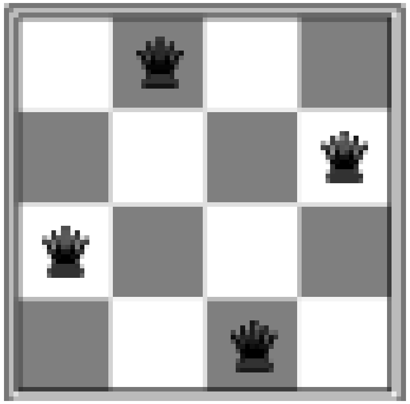
5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

- *So, how many possible worlds?*

More examples

- n-Queens problem
  - variable: location of a queen on a chess board
    - there are *n* of them in total, hence the name
  - domains: grid coordinates (n^2)
  - possible worlds: locations of all queens

$$(n^2)^n$$



no overlaps, indistinguishable queens

$$\binom{n^2}{n} = \frac{n^2!}{(n^2-n)!\,n!}$$

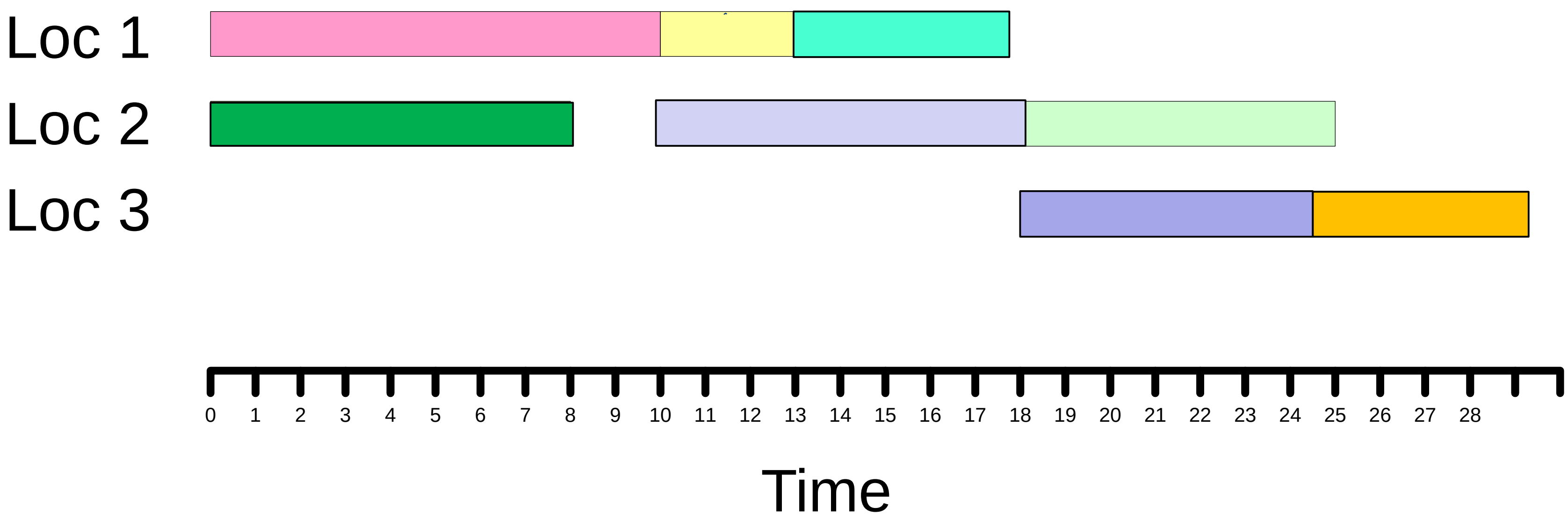
More examples

- Scheduling Problem:
  - variables are different tasks that need to be scheduled (e.g., course in a university; job in a machine shop)
  - domains are the different combinations of times and locations for each task (e.g., time/room for course; time/machine for job)
  - possible worlds: time/location assignments for each task

Scheduling possible world

How many possible worlds?

<possible assignments to a variable>$^{\textit{<number of variables>}}$



More examples….

- Map Coloring Problem
  - variable: regions on the map
  - domains: possible colors
  - possible worlds: color assignments for each region

How many possible worlds?

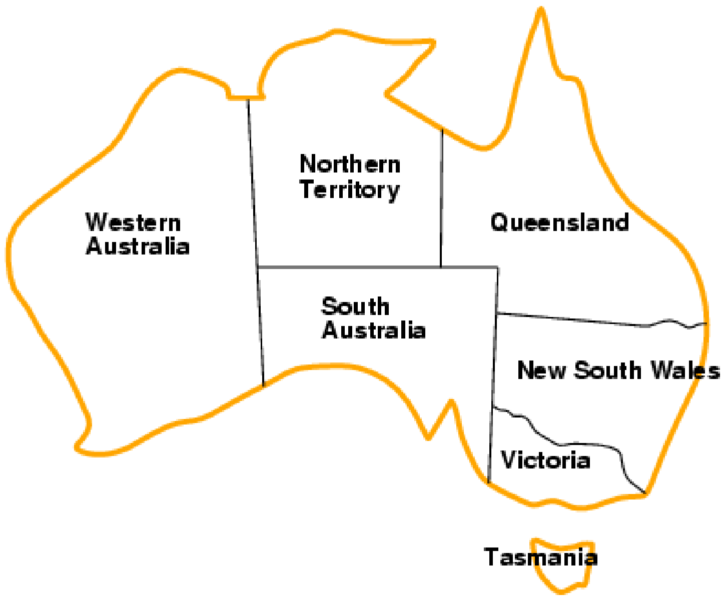
A. $2^{num\_regions}$

B. 2^{num_colors}

C. num_colors * num_regions

D. $num\_regions^{num\_colors}$

E. $num\_colors^{num\_regions}$



Lecture Overview

- **Generic Search vs. Constraint Satisfaction Problems**
- Variables/Features
- Constraints
- CSPs

Constraints

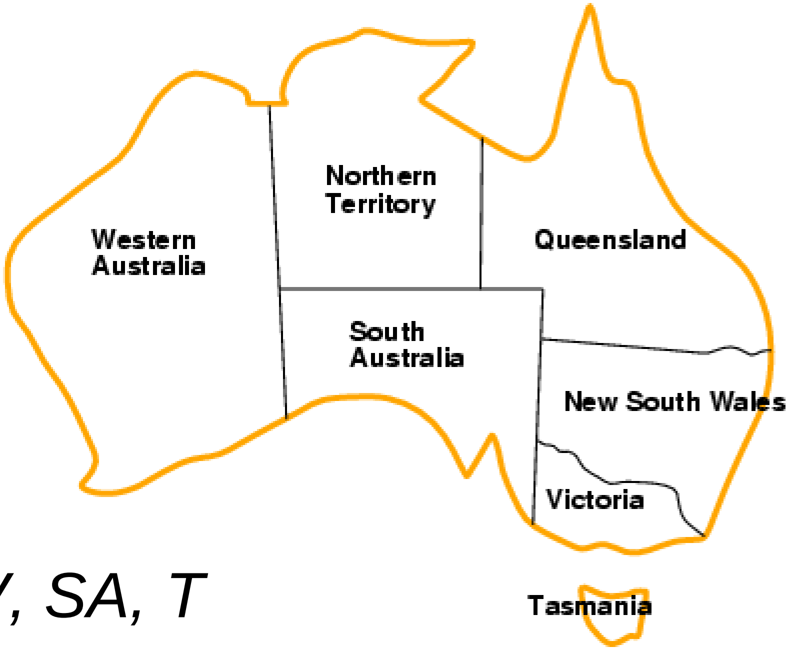
Constraints are restrictions on the values that one or more variables can take

- Unary constraint: restriction involving a single variable

- k-ary constraint: restriction involving the domains of k different variables
  - it turns out that k-ary constraints can always be represented as binary constraints, so we'll *mainly* only talk about this case

- **Constraints can be specified by**
  - giving a function that returns true when given values for each variable satisfy the constraint
  - giving a list of valid domain values for each variable participating in the constraint

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

or, (WA,NT) in {(red,green),(red,blue),(green,red), (green,blue),(blue,red),(blue,green)}

Constraints (cont.)

- A **possible world** satisfies **a set of constraints** if the **set of variables** involved in each constraint take values that are consistent with that constraint

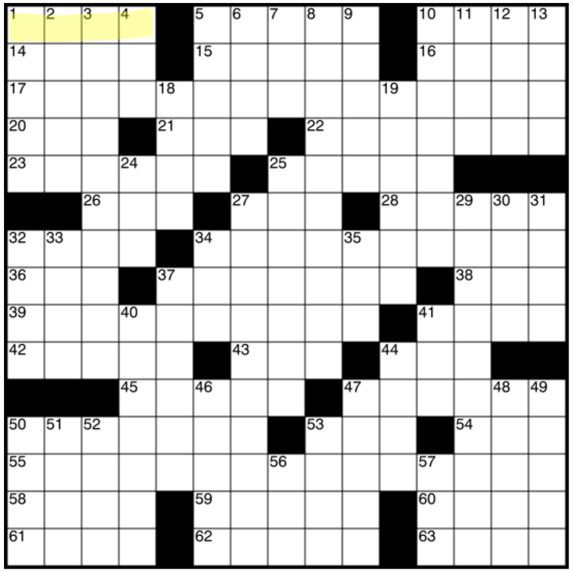
- **Variables:** A,B,C domains [1 .. 10]
- **Possible world W**: {A= 1 , B = 2, C = 10}
- **Constraint set1** {A = B, C>B}
- **Constraint set2** {A $\neq$ B, C>B, (A,C) in {(10,1),(1,10)}

A. W satisfies both set1 and set2
B. W satisfies set1 but not set2
C. W does not satisfy any of the two constraint sets
D. W satisfies set2 but not set1
E. It's okay, I know where my towel is

iclicker.

Examples

- Crossword Puzzle:
  - variables are words that have to be filled in
  - domains are valid English words
  - *constraints:* words have the same letters at points where they intersect



- Crossword 2:
  - variables are cells (individual squares)
  - domains are letters of the alphabet
  - *constraints:* sequences of letters form valid English words

Examples

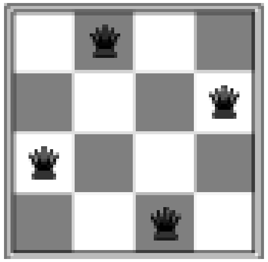
- Sudoku:
  - variables are cells
  - domains are numbers between 1 and 9
  - *constraints:* rows, columns, boxes contain all different numbers

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

More examples

- n-Queens problem
  - variable: location of a queen on a chess board
    - there are *n* of them in total, hence the name
  - domains: grid coordinates
  - *constraints*: no queen can attack another



- Scheduling Problem:
  - variables are different tasks that need to be scheduled (e.g., course in a university; job in a machine shop)
  - domains are the different combinations of times and locations for each task (e.g., time/room for course; time/machine for job)
  - *constraints*:
    - ✓ tasks can't be scheduled in the same location at the same time;
    - ✓ certain tasks can be scheduled only in certain locations;
    - ✓ some tasks must come earlier than others; etc.

Lecture Overview

- **Generic Search vs. Constraint Satisfaction Problems**
- Variables/Features
- Constraints
- CSPs

Constraint Satisfaction Problems: definitions

Definition (Constraint Satisfaction Problem)

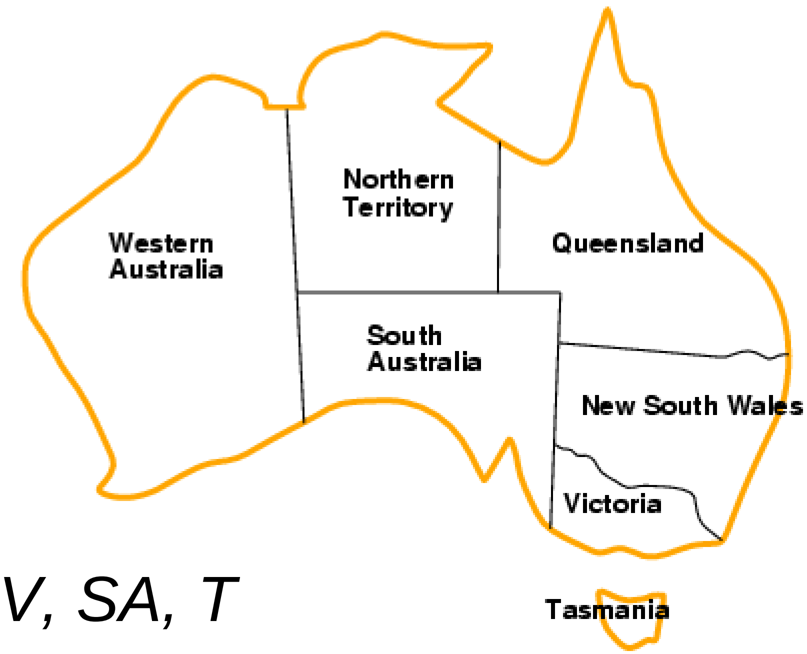
A constraint satisfaction problem consists of

- a set of variables
- a domain for each variable
- a set of constraints

Definition (model / solution)

A model of a CSP is an assignment of values to variables (i.e. a **possible world**) that satisfies all of the constraints.

Example: Map-Coloring



Variables *WA, NT, Q, NSW, V, SA, T*

Domains D_i = {red,green,blue}

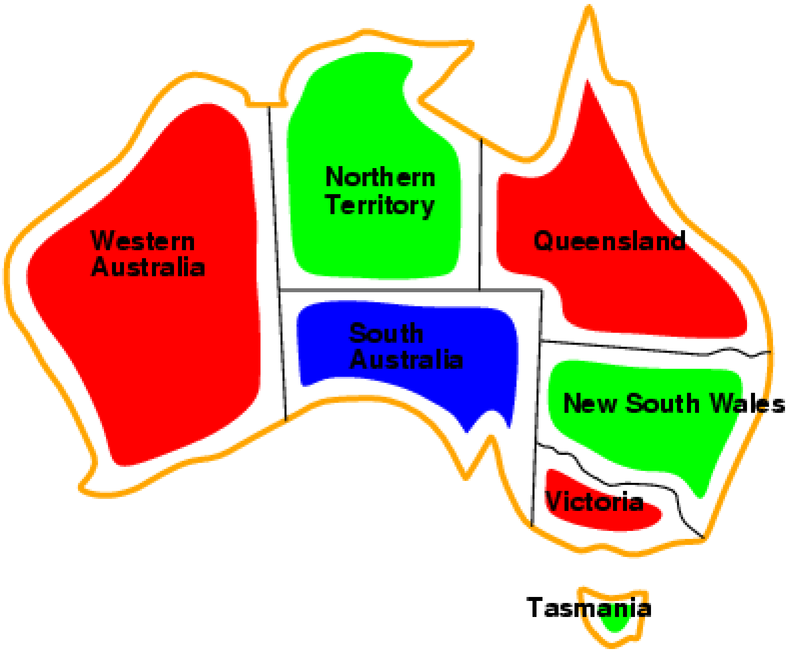
Constraints: adjacent regions must have different colors

e.g., WA ≠ NT, or

(WA,NT) in {(red,green),(red,blue),(green,red), (green,blue),(blue,red),(blue,green)}

Example: Map-Coloring

How many solutions?



Models / Solutions are complete and consistent assignments, e.g., WA = red, NT = green, Q = red, NSW = green, V = red,SA = blue, T = green

Constraint Satisfaction Problem: Variants

We may want to solve the following problems using a CSP

A. determine whether or not a model exists

B. find a model

C. find all of the models

D. ~~count the number of the models~~

E. find the best model given some model quality

- this is now an optimization problem

F. ~~determine whether some properties of the variables hold in all models~~

To summarize

- Need to think of search beyond simple goal driven planning agent.
- We started exploring the first AI Representation and Reasoning framework: **CSPs**

Next class

CSPs: Search and Arc Consistency

(Textbook Chpt 4.3-4.5)