Constraint Satisfaction Problems (CSPs)

Introduction

Computer Science cpsc322, Lecture 11

(Textbook Chpt 4.0 – 4.2)

Setp, 30, 2013

Announcements

- Only one more week for assignment1
- Post questions on Connect

- **Search wrap-up**
  - Go back to learning goals (end of slides)
  - Make sure you understands the inked slides
  - More details or different examples on textbook
  - Work on the practice exercises
  - If still confused, come to office hours

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

Marina-Problem

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 successors / neighbors(n) 2 heuristic(n) 3 goal(n)

Modules we'll cover in this course: R&Rsys

Environment

Problem		Deterministic	Stochastic
Static	Constraint Satisfaction	*Vars + Constraints* Arc Consistency Search → for CSP	
Static	Query	*Logics* Search → for Inference	*Belief Nets* Var. Elimination
Sequential	Planning	*STRIPS* actions Search → for complex planning	*Decision Nets* Var. Elimination *Markov Processes* Value Iteration

Representation

Reasoning Technique

Standard Search vs. Specific R&R systems

Constraint Satisfaction (Problems):

- State
- Successor function
- Goal test
- Solution

two next lectures

Planning :

- State
- Successor function
- Goal test
- Solution

following weeks

Inference

- State
- Successor function
- Goal test
- Solution

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 A, B
  - each variable V has a domain *dom*(V) of possible values

    dom(B) = dom(A) = {0, 1}

- Variables can be of several main kinds:
  - Boolean: |*dom*(V)| = 2 propositions
  - 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

  e.g. {A = 1, B = 0}

Example (lecture 2)

Mars Explorer Example

Weather	{S, C}
Temperature	[-40, 40]
Longitude	[0, 359]
Latitude	[0, 179]

One possible world (state) {S, -30, 320, 91}

Number of possible (mutually exclusive) worlds (states)

2 x 81 x 360 x 180

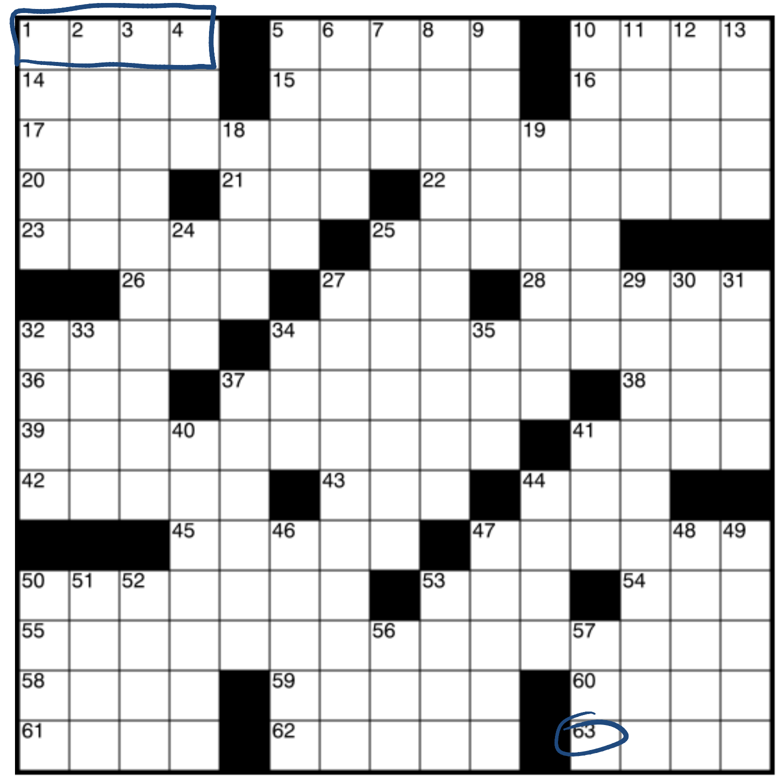
Product of cardinality of each domain

… always exponential in the number of variables

Examples

h_1

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



63

- *Number of English words?* $\sim 150*10^3$
- *Number of words of length k ?* $1\text{-}10$ $\sim 15\times10^3$
- *So, how many possible worlds?*

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

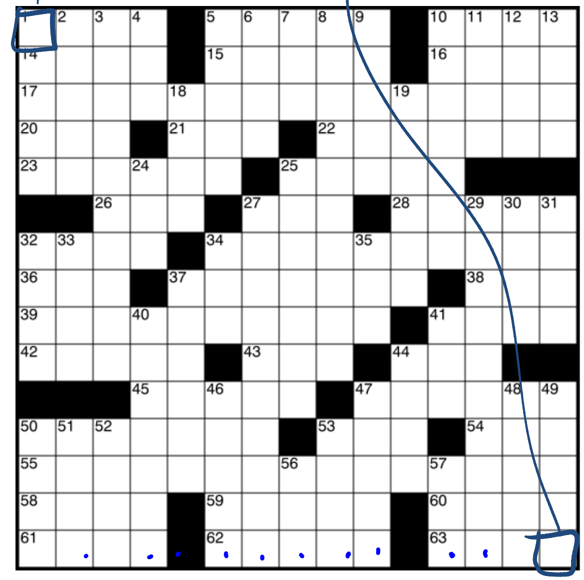
$(15\times10^3)^{63}$

More Examples

$V_{0,0}$ $V_{15,15}$

- Crossword 2: 15×15
  - variables are cells (individual squares) ~ 225 vars
  - domains are letters of the alphabet 26
  - possible worlds: all ways of assigning letters to cells
    - *So, how many possible worlds?*

$(26)^{225}$

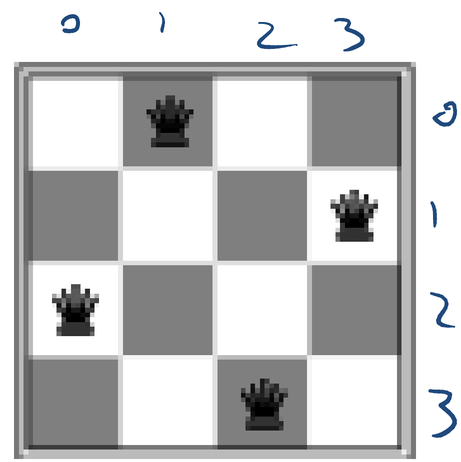


- Sudoku:
  - variables are empty cells
  - domains are numbers between 1 and 9
  - possible worlds: all ways of assigning numbers to cells
- *So, how many possible worlds?* $9^{\#\text{empty cells}}$

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

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

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

possible ways to choose n location out of n^2

$\frac{16!}{12!\,4!}$

More examples

- Scheduling Problem: task 1, task 2, ...
  - 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) (start_time, location)
  - possible worlds: time/location assignments for each task

e.g. $task_1 = \{11am.., room\ 310\}$
$task_2 = \{12pm, room101\}$
......

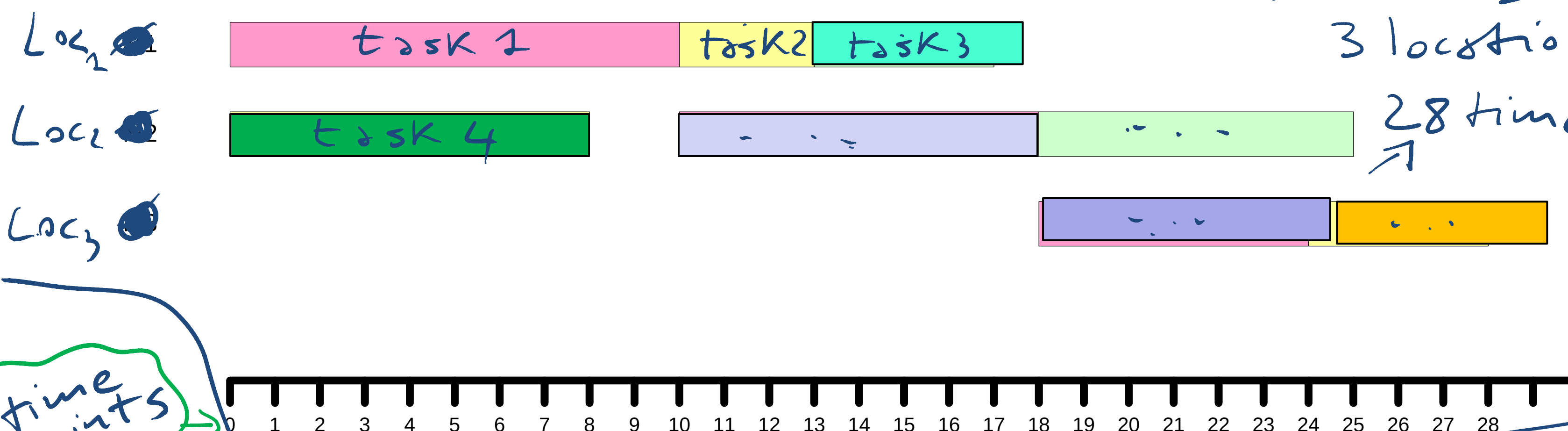
Scheduling possible world

- *how many possible worlds?*

in general

$$(\#locs * timepoints)^{\#tasks}$$

$$(28 * 3)^{8}$$



→ 8 tasks
3 locations
28 timepoints

time points

$task_1$ (start_time=0, Loc_1)
$task_2$ (" " =10, Loc_2
. . . .

possible world

More examples….

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

- *how many possible worlds?*

$(\#\text{colors})^{\#\text{regions}}$

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 $A \quad B \quad C \quad \{0, 1\}$

- Unary constraint: restriction involving a single variable
  - $\{A=1\} \quad \{B<1\}$
- k-ary constraint: restriction involving the domains of k different variables $A=B \quad A>B+C$
  - 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 which satisfy the constraint
  - giving a list of valid domain values for each variable participating in the constraint $\{A=0 \quad B=0\}$ $\{A=1 \quad B=1\}$

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, SA ≠ NT, SA ≠ WA . . . . . .

or, NT(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

in class C was correct

C. W does not satisfy any of the two constraint sets

D. W satisfies set2 but not set1

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

h_1 15 x 15 V_1

$h_1[0] = V_1[0]$, ...

~ 225 constraints

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

63

A

63 constraints

concatenate$(A[0,0] \ldots A[0,3]) \in$ English word of length 4

Examples

actually 20 because some overlap

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

$\# \text{ of constraint } = \; \sim \; \#\text{empty-cells} \times 24 - \{8+8+8\}$

A 0 1 2 . . . . . .

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

for all i, $i \neq 2$

$A[2,0] \neq A[i,0]$

which means

$A[2,0] \neq A[0,0]$

$A[2,0] \neq A[1,0]$

⋮

solution

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

e.g. two Queens cannot be on the same column / row

$Q_1 = \{x_1, y_1\}$
$Q_2 = \{x_2, y_2\}$

$x_1 = x_2$ and
$y_1 = y_2$

- 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*: e.g. $task_1(loc_1, start\text{-}t_1)$ $task_2(loc_2, start\text{-}t_2)$ if $start\text{-}t_1 = start\text{-}t_2$ then $loc_1 \neq loc_2$
    - ✓ 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

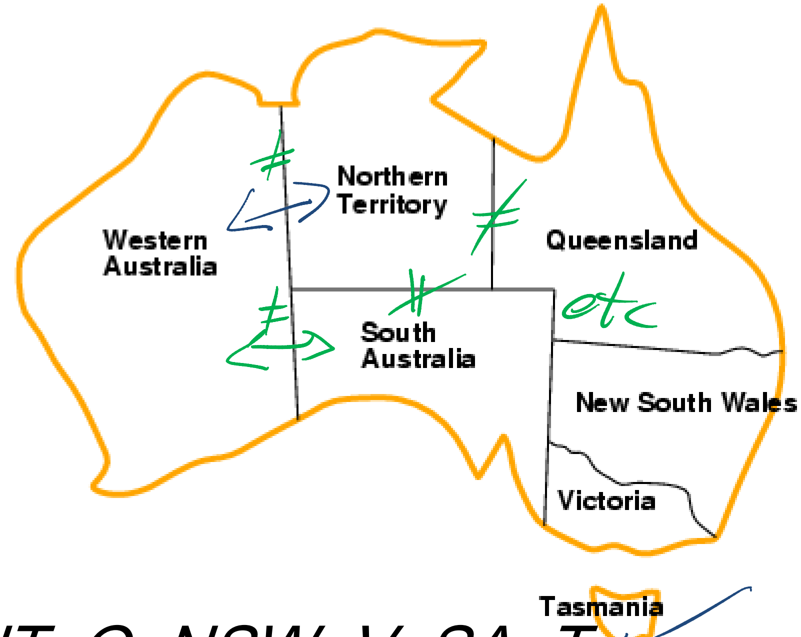
- a set of variables
- a domain for each variable
- a set of constraints

possible world

Definition (model / solution)

A model of a CSP is an assignment of values to variables 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 . . . .

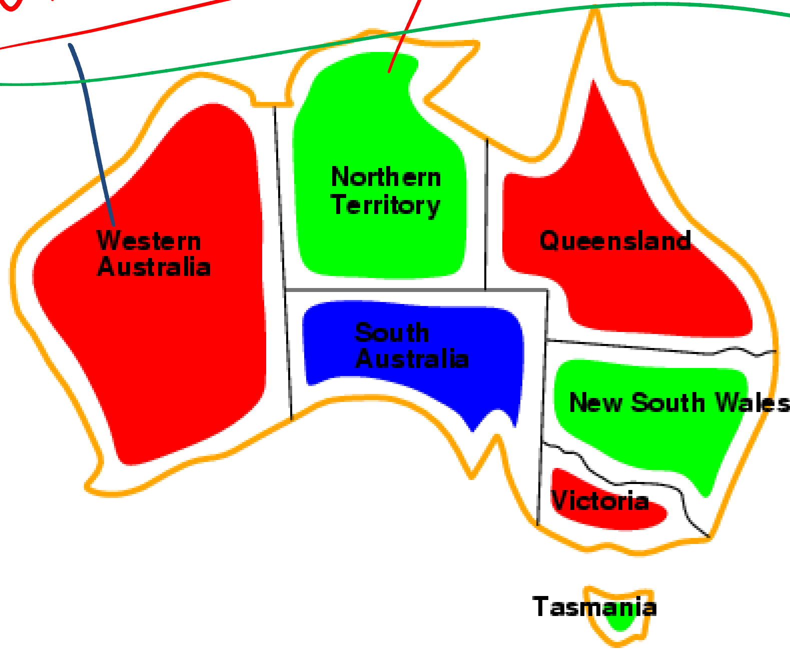
(WA,NT) in {(red,green),(red,blue),(green,red), (green,blue),(blue,red),(blue,green)}

Example: Map-Coloring

UNIQUE ?

WA = red NT = green

with these two it becomes unique



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

useful to avoid waiting time on B

B. find a model

our focus

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~~

not in this course

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)

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 3
  - List vs. function format. 3

Verify whether a possible world satisfies a set of constraints (i.e., whether it is a model, a solution)

Extra slide (may be used here?)

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

Goal state: 9×9 grid completely filled so that
- each column,
- each row, and
- each of the nine 3×3 boxes
- contains the digits from 1 to 9, only *one* time each

A possible **start state** (partially completed grid)

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