AI Tools, Techniques and Applications

Subject Code: 17IT2504A

UNIT I

Dr. Sangeetha. Y
Asso.Professor

Textbooks:

1. Stuart J. Russell and Peter Norvig, Artificial Intelligence
   A Modern Approach
2. Tom Markiewicz& Josh Zheng,Getting started with
   Artificial Intelligence, Published by O’Reilly Media,2017

References:

1. AurélienGéron, Hands on Machine Learning with
Scikit-Learn and TensorFlow [Concepts, Tools, and Techniques to Build Intelligent Systems], Published by O’Reilly Media,2017

Intelligence

Intelligence is the ability to learn and understand, to solve problems and to make decisions.

Artificial Intelligence is a branch of *Science* which deals with helping machines to find solutions to complex problems in a more human-like fashion.

For this it generally involves borrowing characteristics from human intelligence, and applying them as algorithms in a computer friendly way.

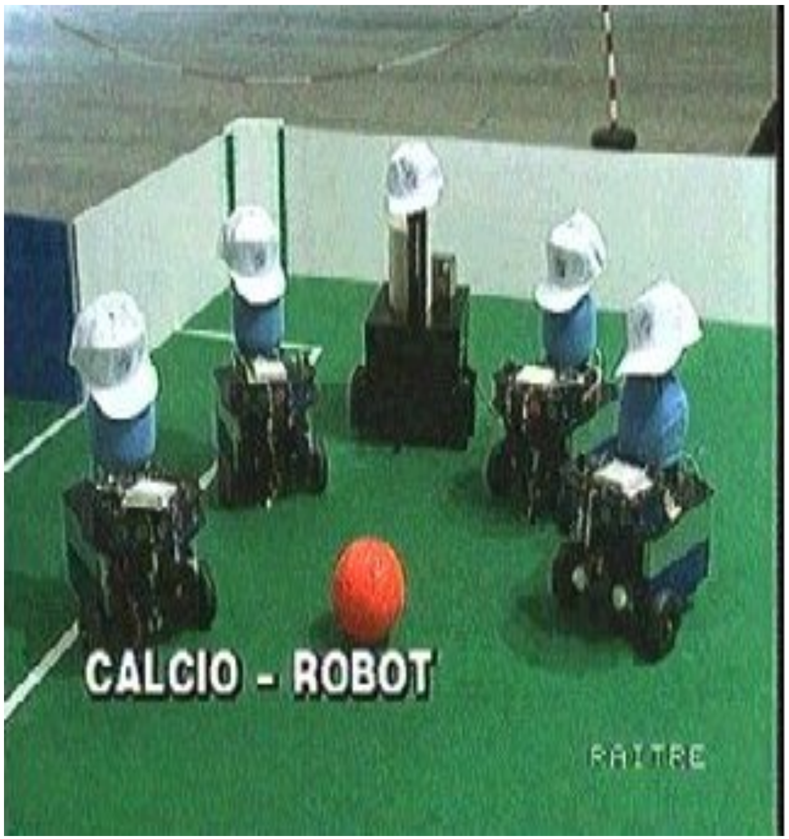
Applications of AI

- There are movies that have a robot in some way in the plot, but these are really either pretend robots or industrial automation devices.
- An automatic machine sweeper
- An automatic car for a child to play with
- A machine removing mines in a war field
- In space
- In military , and many more..

- Personal service robots
  - House cleaning
  - Lawn mowing
  - Assistance to the elderly and handicapped
  - Office assistants
  - Security services

RoboCup

RoboCup is an annual international robotics competition proposed



- Robots must cooperate in…
  - Strategy acquisition
  - Real-time reasoning
  - Multi-agent collaboration
  - Competition against another team of robots

AI Survey …….

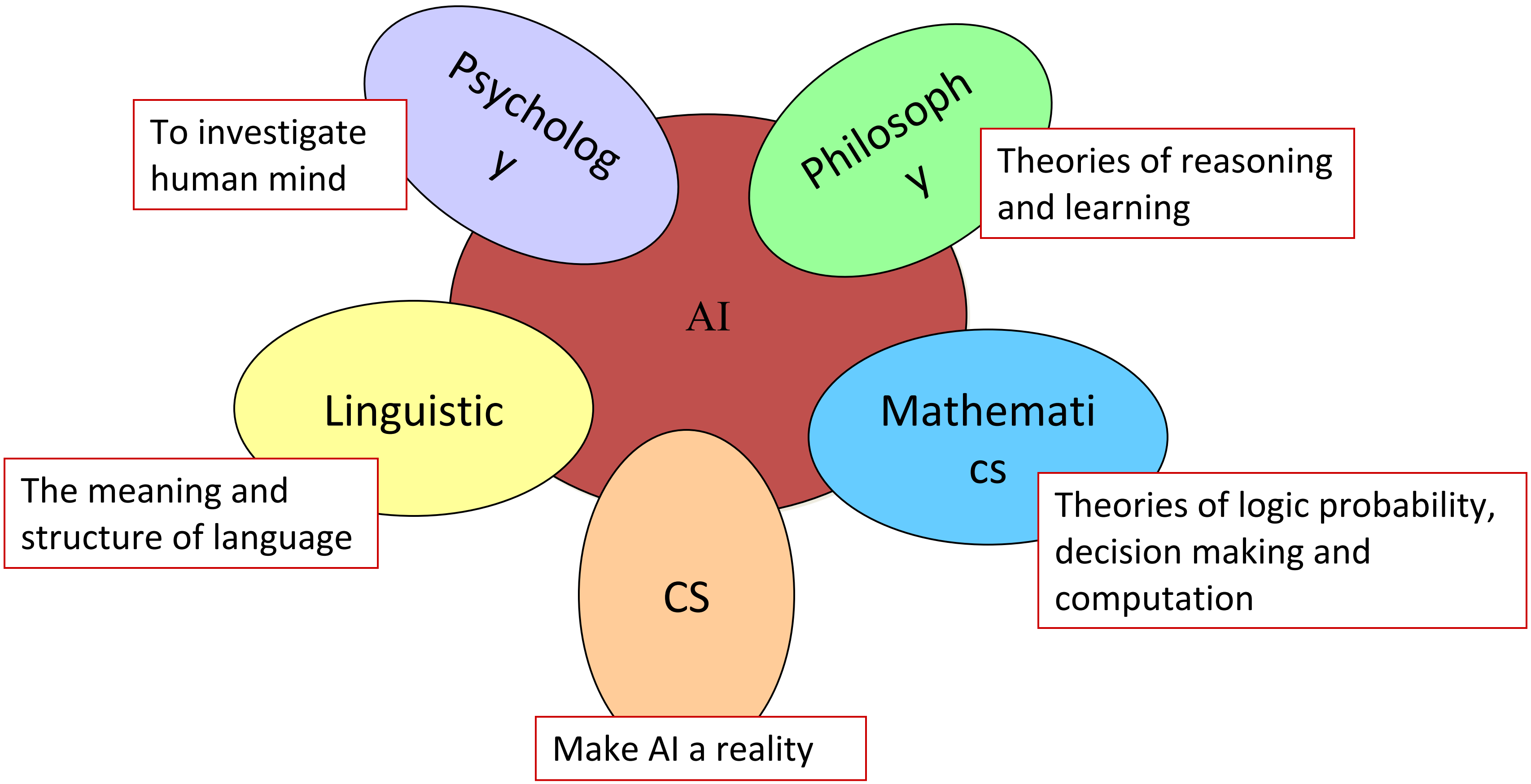
- Artificial Intelligence can be classified under two general motives. Some people think that increasing use of Artificial Intelligence would cut jobs from workers, but on the other hand, AI would increase safety in the workplace. It would do dangerous tasks so humans do not have to while speeding up production lines with increased accuracy.
- AI can also be used to aid in the private sector with intelligent financial and electronic help applications. Artificial Intelligence is becoming more and more advanced and will be used a lot more as its usefulness increases.

Main use of AI

- Speed
- Can work in hazardous/dangerous temperature
- Can do repetitive tasks
- Can do work with accuracy

AI Foundations?

AI inherited many ideas, viewpoints and techniques from other disciplines.

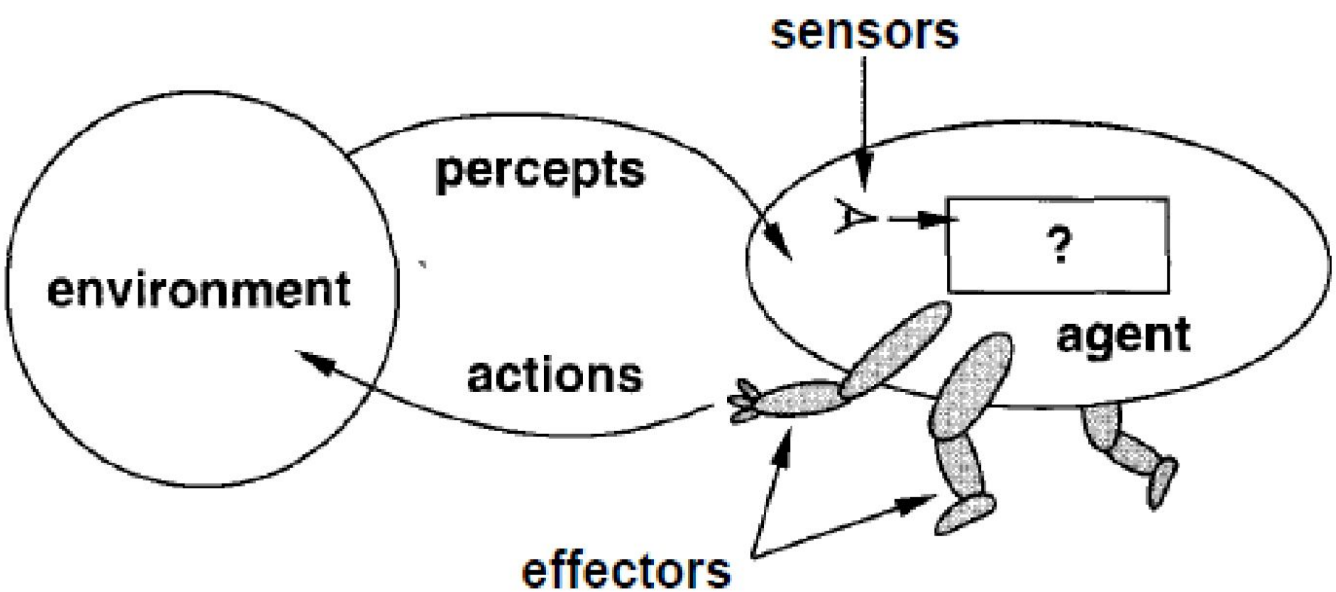


INTELLIGENT AGENTS: Agents and Environments

- **An agent :** Perceiving its environment through sensors
and acting upon that environment through actuators.
Agents include humans, robots, softbots, thermostats, etc.
The agent function maps from percept histories to actions:
f : P * → A.
- **A human agent** has eyes, ears, and other organs for sensors, and hands, legs, mouth, and other body parts for effectors.
- **For robots:**

cameras
infrared range finders — Sensors

various motors — effectors



Agents interact with environments through sensors and effectors.

agent: robot vacuum cleaner

- **Environment:** floors of your apartment
- **Sensors:**
  - ❑ dirt sensor: detects when floor in front of robot is dirty
  - ❑ bump sensor: detects when it has bumped into something
  - ❑ power sensor: measures amount of power in battery
  - ❑ bag sensor: amount of space remaining in dirt bag
- **effectors:**
  - ❑ motorized wheels
  - ❑ suction motor
  - ❑ plug into wall? empty dirt bag?
  - ❑ percepts: “Floor is dirty”
  - ❑ actions: “Forward, 0.5 ft/sec”

Constraint satisfaction problems (CSPs)

- Constraint satisfaction problems (CSPs) are mathematical problems defined as a set of objects whose state must satisfy a number of constraints or limitations.

Constraint satisfaction problems

- What is a CSP?
  - Finite set of variables $X_1, X_2, \ldots, X_n$
    - Where each variable X_i is a Nonempty domain of possible values $D_1, D_2, \ldots D_d$
  - Finite set of constraints $C_1, C_2, \ldots, C_m$
    - Each constraint C_i limits the values that variables can take, e.g., $X_1 \neq X_2$
- A state is defined as an assignment of values to some or all variables.
- Consistent assignment: assignment does not violate the constraints.

Constraint satisfaction problems

- An assignment is complete when every variable is assigned a value.
- A solution to a CSP is a complete assignment that satisfies all constraints.
- Applications:
    - Eight queens puzzle
    - Map coloring problem
    - Sudoku etc…..

Sudoku as a Constraint Satisfaction Problem (CSP)

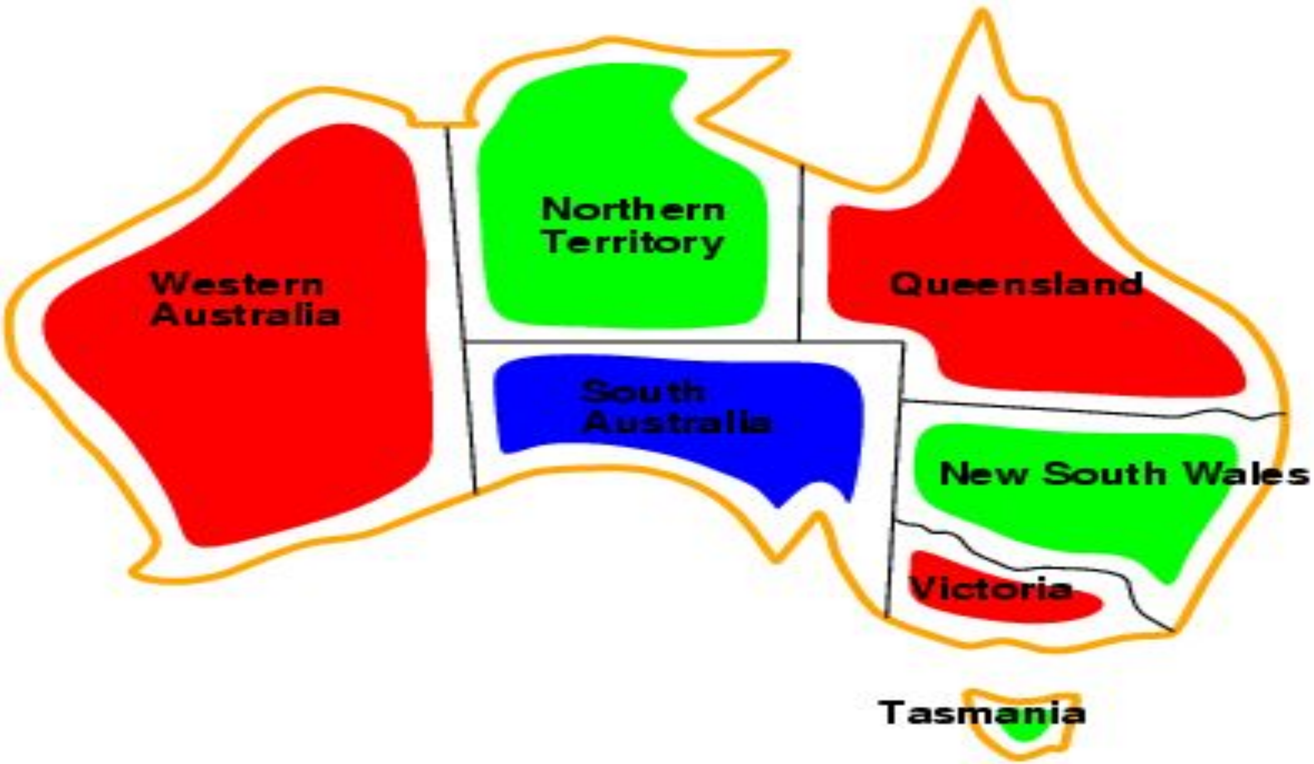
	1	2	3	4	5	6	7	8	9
A		6		1		4		5	
B			8	3		5	6		
C	2								1
D	8			4		7			6
E			6				3		
F	7			9		1			4
G	5								2
H			7	2		6	9		
I		4		5		8		7	

- Variables: 81 variables
  - A1, A2, A3, …, I7, I8, I9
  - Letters index rows, top to bottom
  - Digits index columns, left to right
- Domains: The nine positive digits
  - A1 $\in$ {1, 2, 3, 4, 5, 6, 7, 8, 9}
  - Etc.
- Constraints: 27 *Alldiff* constraints
  - *Alldiff*(A1, A2, A3, A4, A5, A6, A7, A8, A9)
  - Etc.

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** **So (WA,NT) must be in {(red,green),(red,blue),(green,red), …}**

Example: Map-Coloring

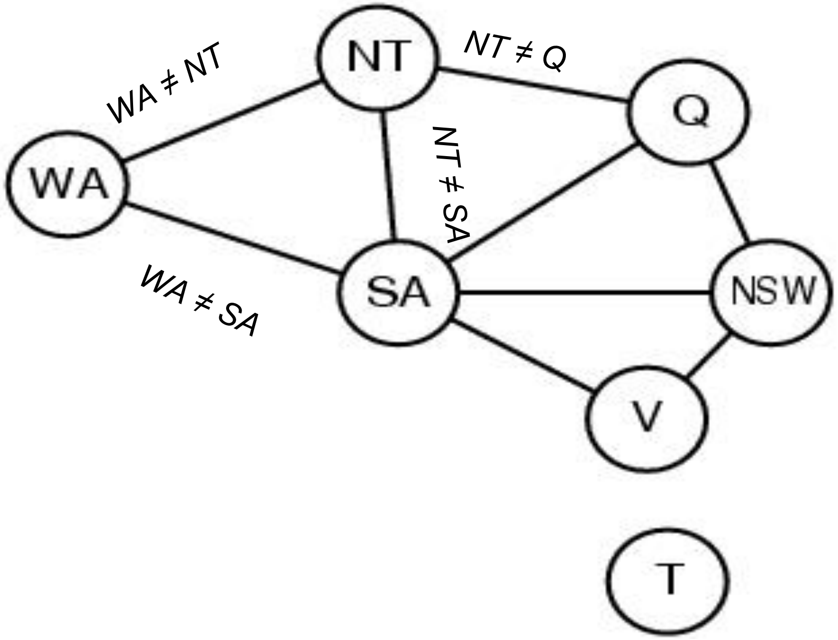


Solutions are complete and consistent assignments,

– e.g., WA = red, NT = green, Q = red, NSW = green, V = red, SA = blue, T = green

Constraint graph

- Constraint graph:
  - nodes are variables
  - arcs are constraints



Varieties of constraints

- Unary constraints
  - e.g., SA ≠ green
- Binary constraints
  - e.g., SA ≠ WA
- Higher-order constraints
  - e.g., crypt-arithmetic column constraints
- Preference (soft constraints)
  - e.g. red is better than green

Cryptarithmetic Problem

- The goal here is to assign each letter a digit from 0 to 9 so that the arithmetic works out correctly.

Example: Cryptarithmetic

C_4	C_3	C_2	C_1	$\leftarrow$	Carry
	S	**E**	**N**	**D**	
+	**M**	**O**	**R**	**E**	
M	**O**	**N**	**E**	**Y**	

- Variables: $S\ E\ N\ D\ M\ O\ R\ E\ N\ \ Y,$
- Domain: $\{0,1,2,3,4,5,6,7,8,9\}$
- Constraints: No two letters have the same value
- Initial Problem State

  S = ? ; E = ? ;N = ? ; D = ? ; M = ? ;O = ? ; R = ? ;Y = ?

Carries :

$C_4 = ?$; $C_3 = ?$; $C_2 = ?$; $C_1 = ?$

$$
\begin{array}{cccccc}
C_4 & C_3 & C_2 & C_1 & \leftarrow & \text{Carry} \\
 & S & E & N & D & \\
+ & M & O & R & E & \\
\hline
M & O & N & E & Y & \\
\hline
\end{array}
$$

M has to be non zero digit, so the value of C4 =1

Constraint equations:

$Y = D + E \quad C_1 \longrightarrow$

$E = N + R + C_1 \quad C_2 \longrightarrow$

$N = E + O + C_2 \quad C_3 \longrightarrow$

$O = S + M + C_3 \longrightarrow C_4$

$M = C_4$

S = 9 ; E = 5 ; N = 6 ; D = 7 ;
M = 1 ; O = 0 ; R = 8 ;Y = 2

Backtracking search for CSPs

- The term backtracking search is used for a depth-first search that chooses values for BACKTRACKING SEARCH one variable at a time and backtracks when a variable has no legal values left to assign

- **General purpose methods that address the following:**
  - Which variable is assigned next
  - What are the implications of the current variable assignments
  - When a path fails

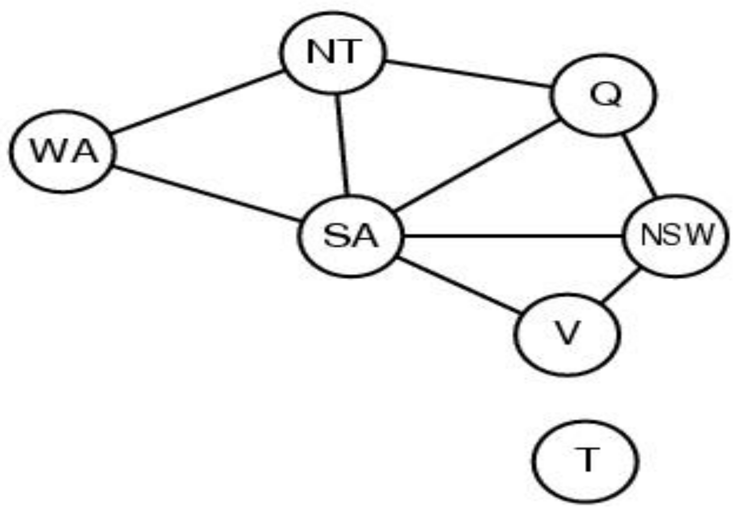
Variable and value ordering

- For Ex:
  - After assigning WA=red and NT=green there is only one possible value for SA i.e blue.
  - After SA is assigned, Values for Q,NSW, V are all forced
  - Choosing the variable with fewest legal moves is called MRV heuristic or MCV or fail-first heuristic.

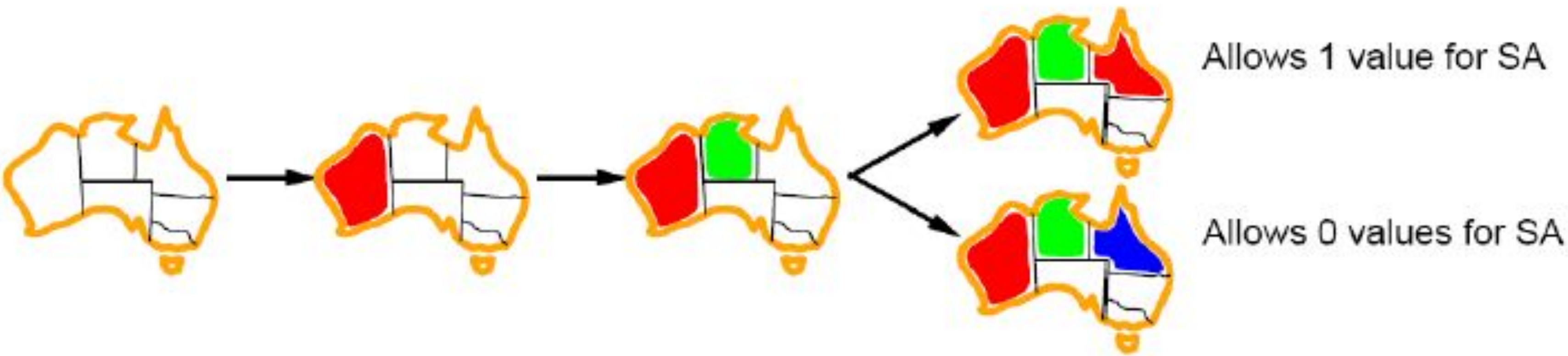

Northern
Territory
Western
Australia
Queensland
South
Australia
New South Wales
Victoria
Tasmania

Variable and value ordering

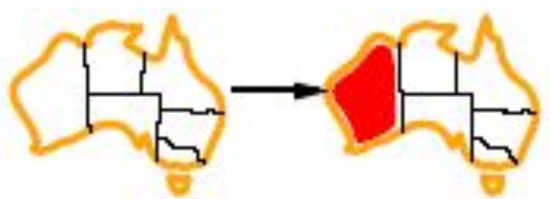
- Degree heuristic: Degree heuristic can be useful as a tie breaker after MRV
- MRV heuristic is a more powerful guide, but the degree heuristic can be useful as a tie breaker

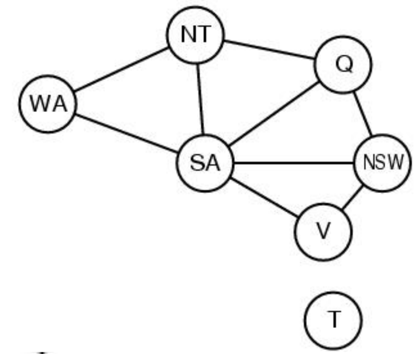


LCV: Least constraining value



Forward checking





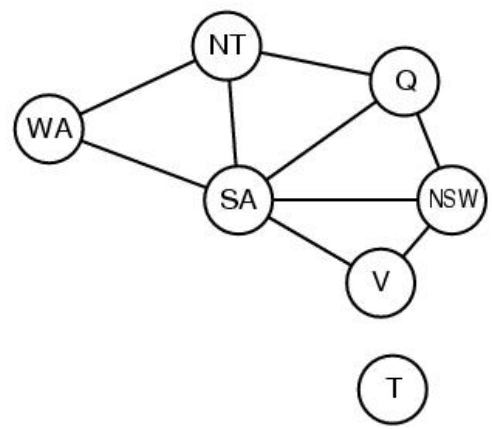
- **Assign *{WA=red}***

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

Forward checking

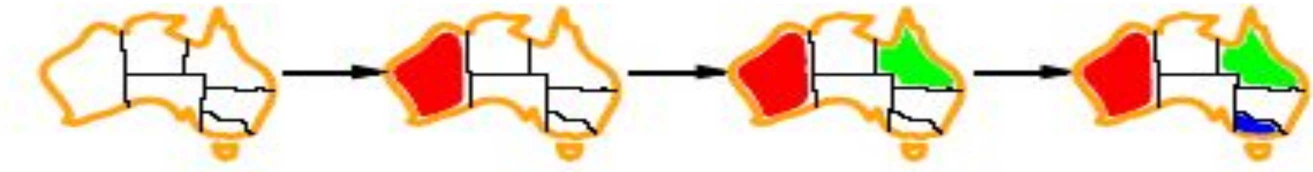
- **Assign *{Q=green}***

- **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***

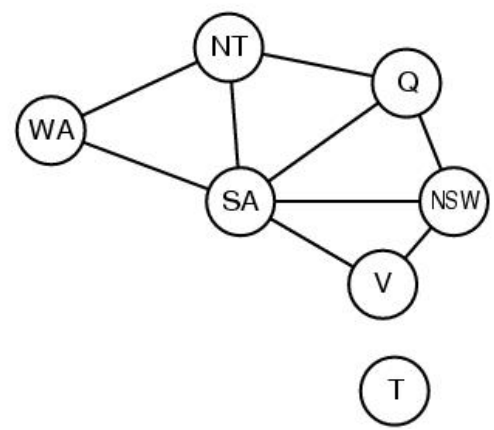


- ***MRV heuristic* would automatically select NT or SA next**

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.

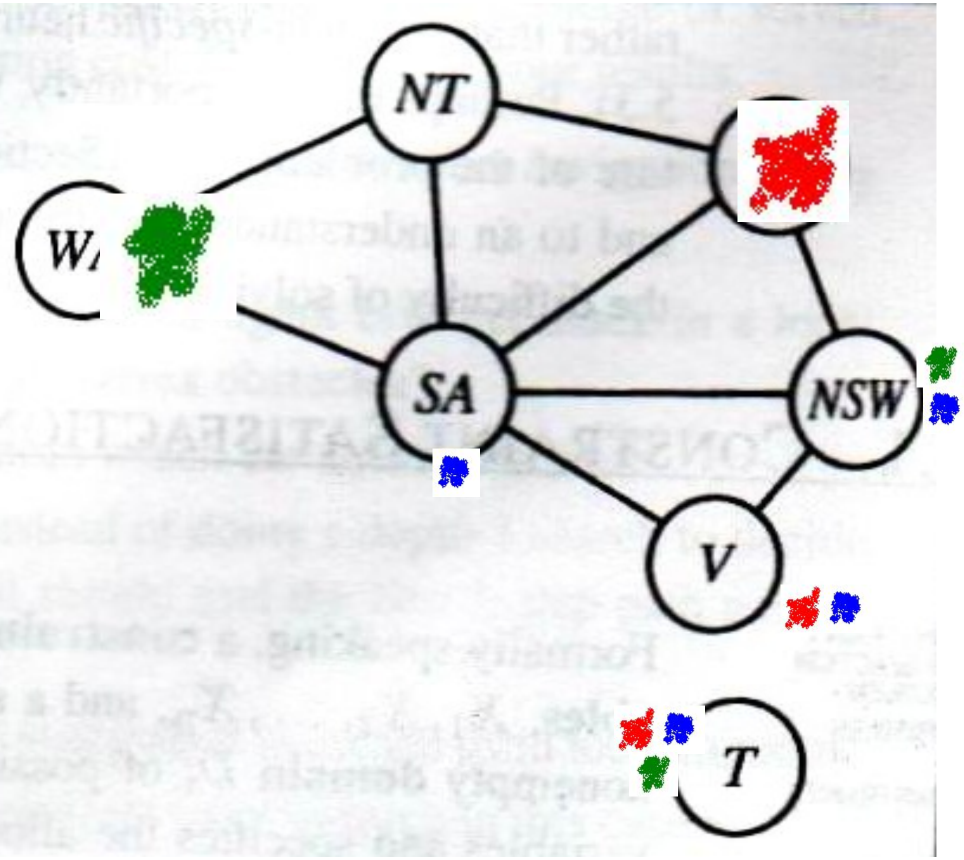


Constraint propagation

- There are some inconsistencies with forward checking
- Constraint propagation is stronger than forward checking by considering arc consistency
- the arc is consistent if, for *every* value x of SA, there is *some* value y of NSW that is consistent with x.
- For Ex: SA - {blue} and NSW = {red, blue}

Arc Consistency

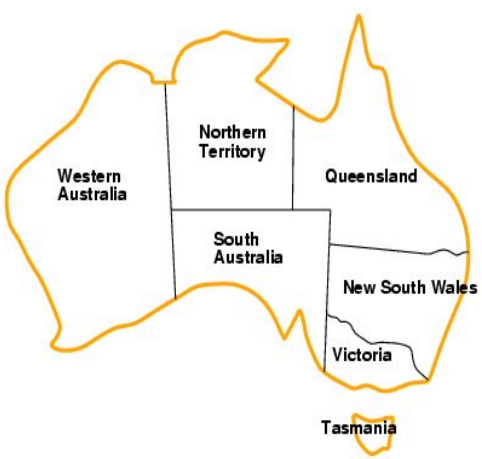
- A→B is consistent if for each remaining value in domain of   A, there may be a consistent value in domain of B.
  - Consistent:
    - SA→NSW, NSW→V,…
  - Not Consistent:
    - NSW→SA, NT→SA,…

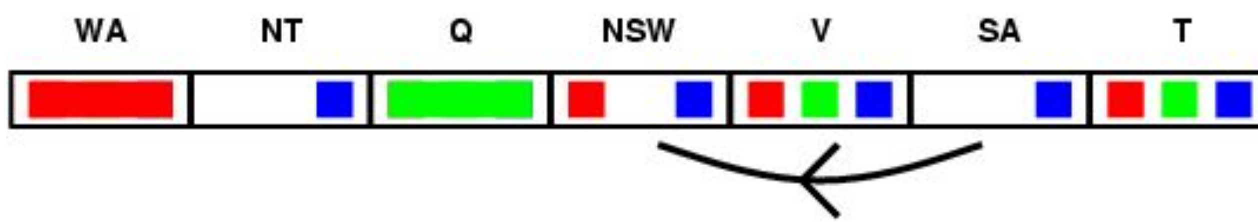


Arc consistency

- Simplest form of propagation makes each arc consistent
- *X* → *Y* is consistent iff

  for every value *x* of *X* there is some allowed *y*



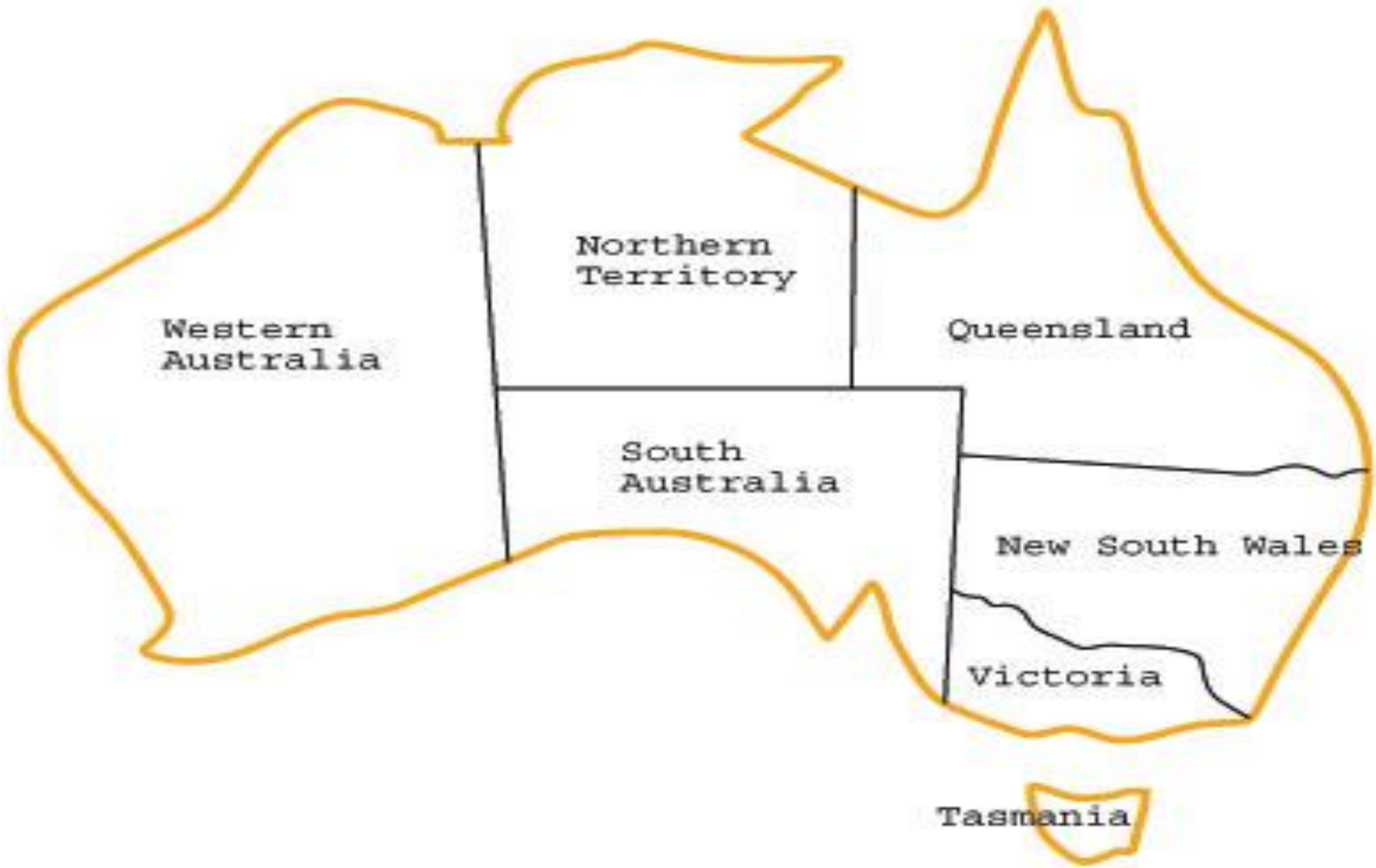


Intelligent Backtracking

- The BACKTRACKING-SEARCH algorithm has a very simple policy for what to do when a branch of the search fails: back up to the preceding variable and try a different value for it.
- This is called chronological backtracking, because the *most recent decision* point is revisited.
- There are much better ways
- One way to get rid of the problem is using **intelligent backtracking** algorithms
- **Backjumping** (**BJ**) is different from BT in the following:

BJ vs. BT

We want to color each area in the map with a different color



BJ vs. BT

- Let’s consider what BT does in the map coloring problem
  - Assume that variables are assigned in the order *Q, NSW, V, T, SA, WA, NT*
  - Assume that we have reached the partial assignment

    Q = red, NSW = green, V = blue, T = red

  - When we try to give a value to the next variable *SA*, we find out that all possible values violate constraints
    - Dead end!
  - BT will backtrack to try a new value for variable *T*!
    - Not a good idea!

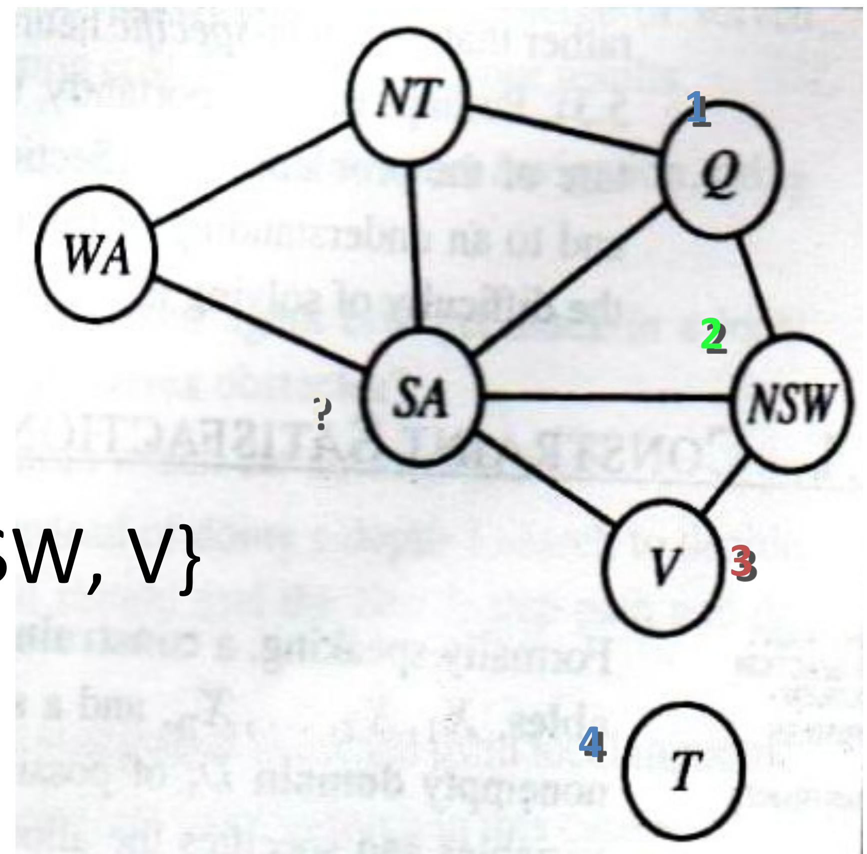
BJ vs. BT

- BJ has a smarter approach to backtracking
  - A more intelligent approach to backtracking is to go all the way back to one of the set of variables that *caused the **failure***
  - The set of these variables is called a **conflict set**
  - The conflict set for SA is {Q, NSW, V}

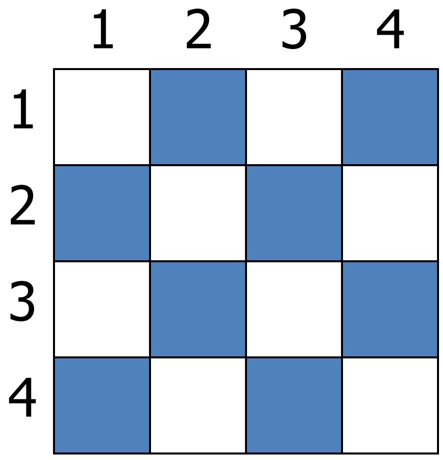
Back Jumping

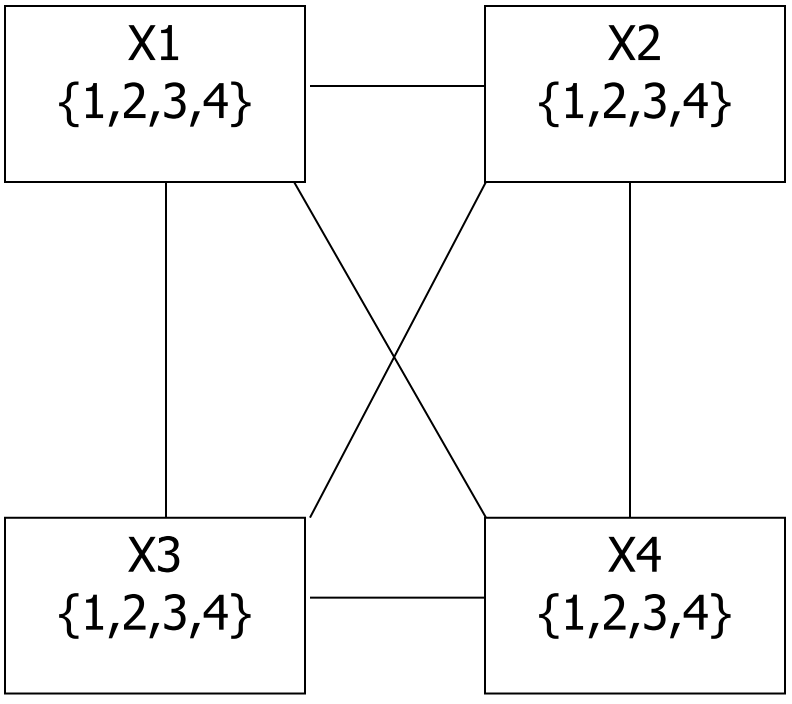
1. Q Red
2. NSW Green
3. V Blue
4. T Red
5. SA ?

- Conflict Set: {Q,NSW, V}

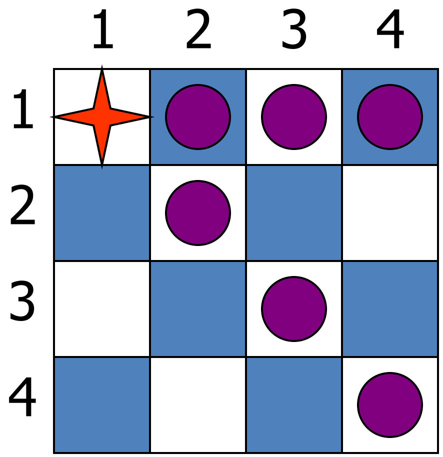


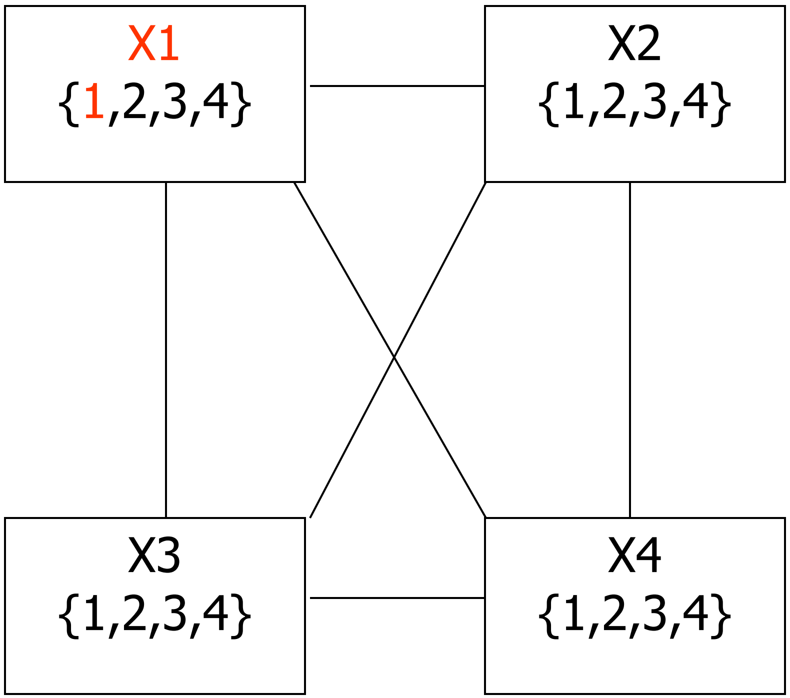
Example: 4-Queens Problem



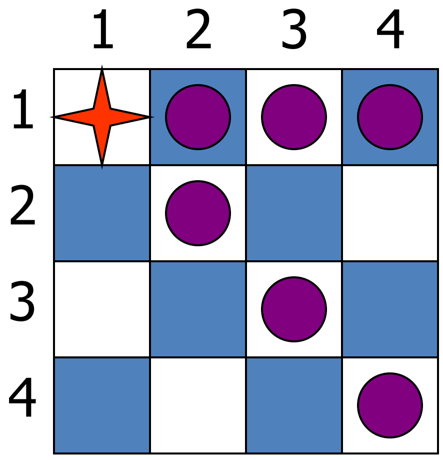


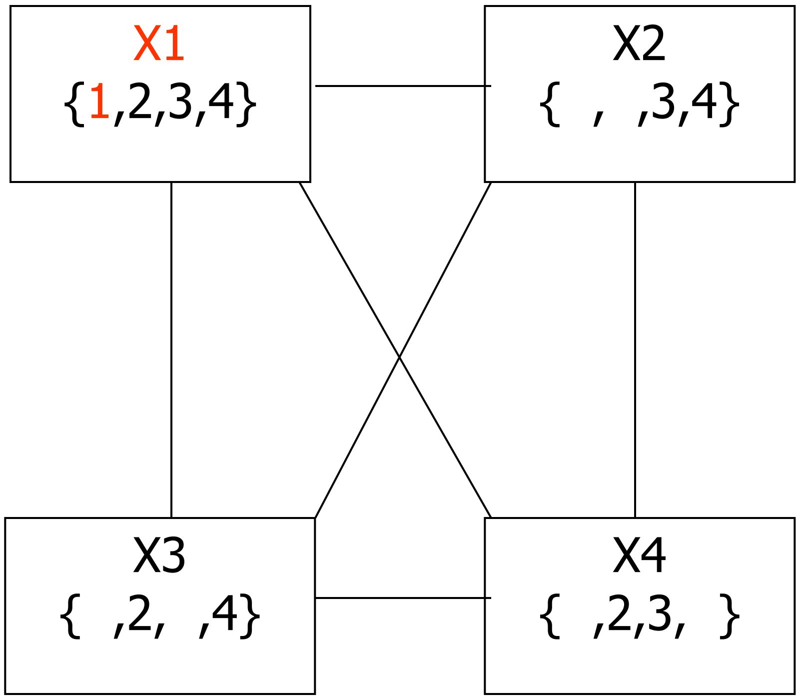
Example: 4-Queens Problem



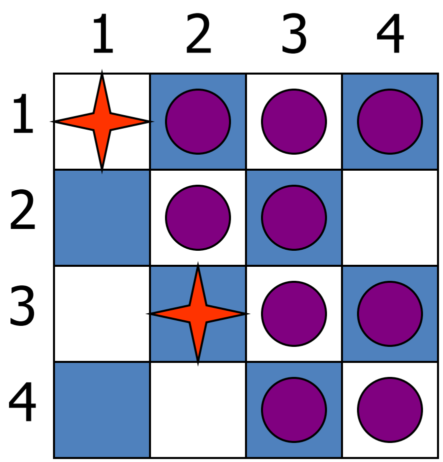


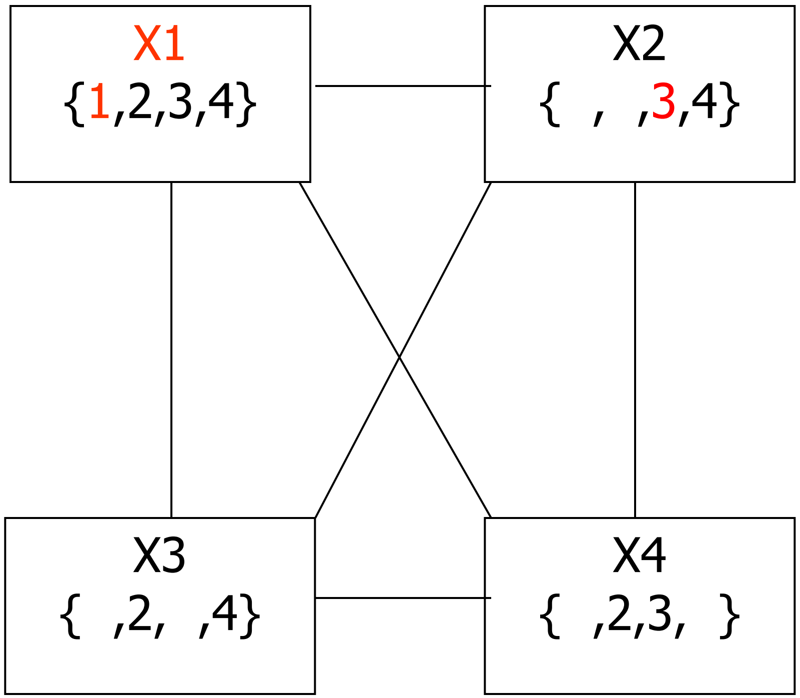
Example: 4-Queens Problem



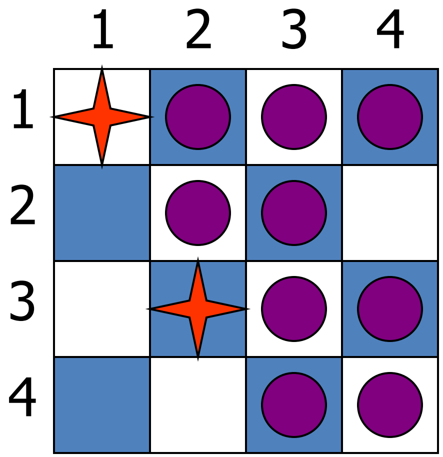


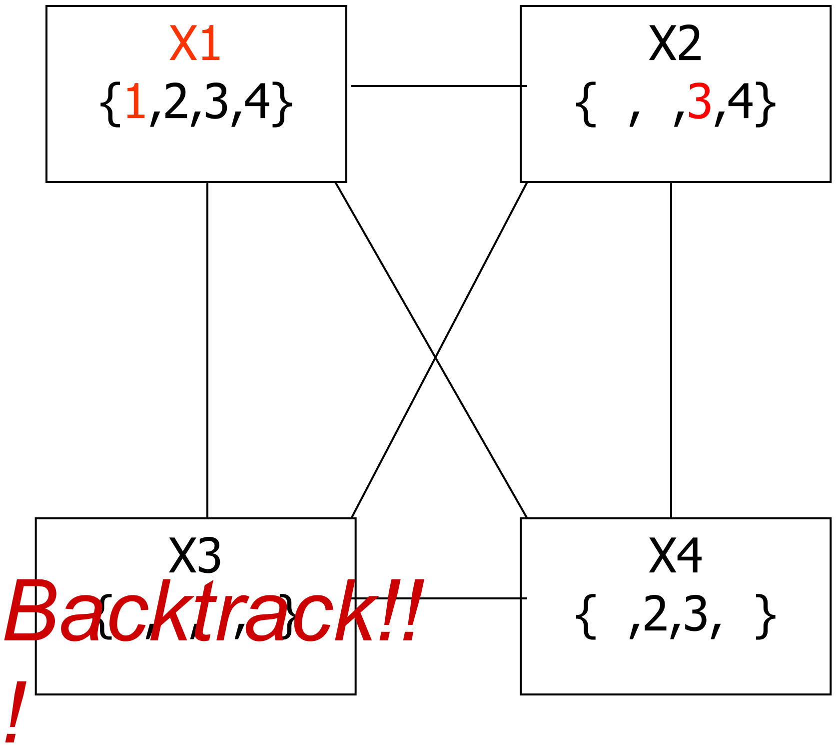
Example: 4-Queens Problem



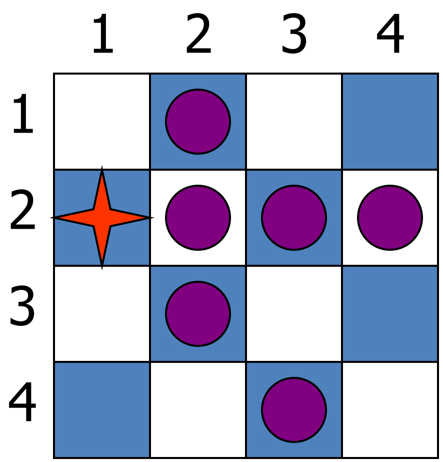


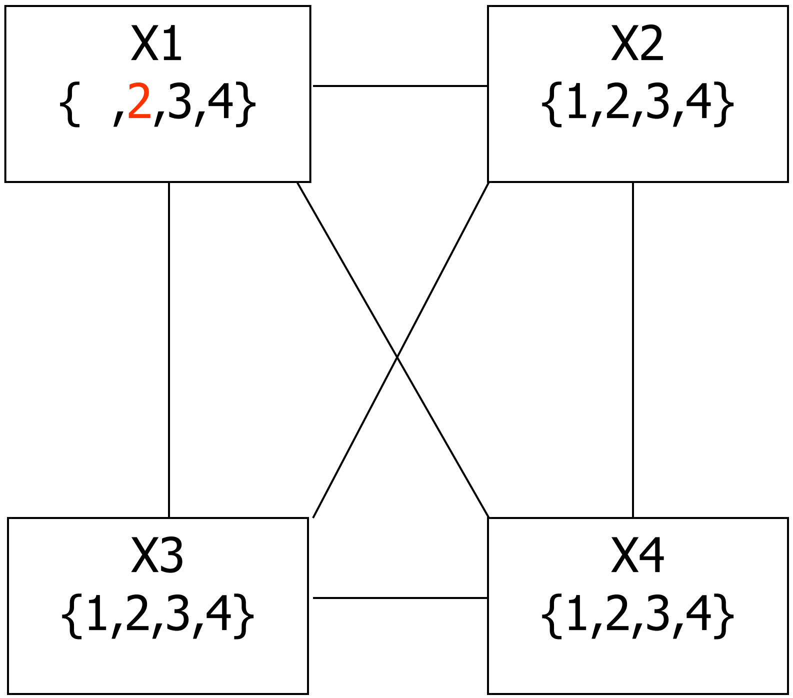
Example: 4-Queens Problem



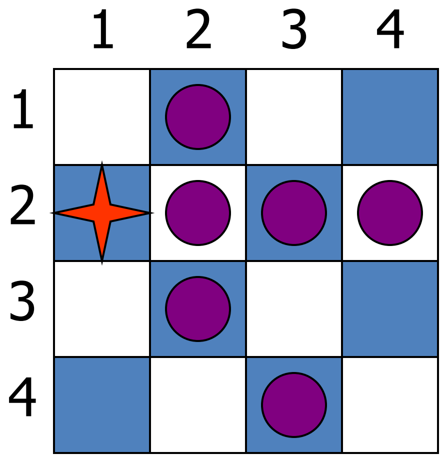


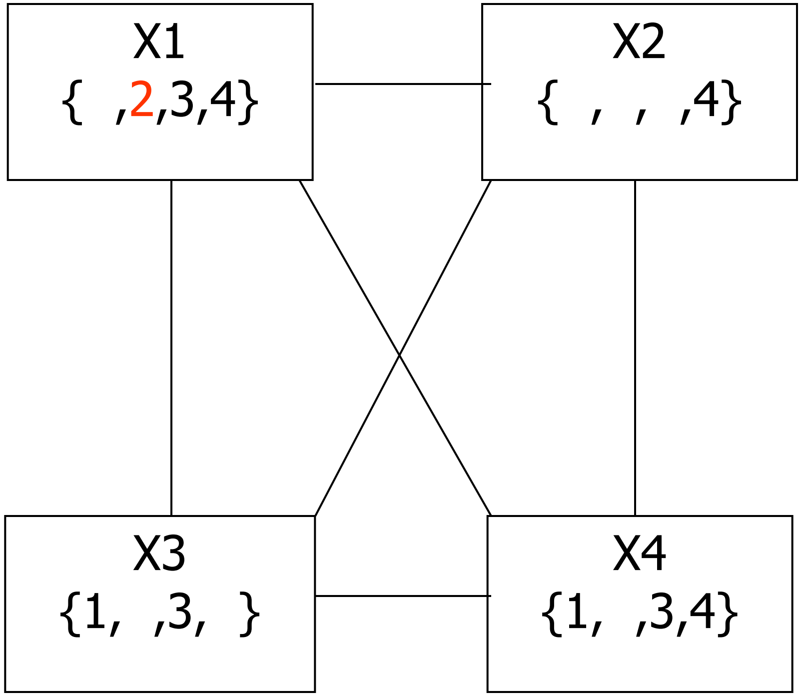
Example: 4-Queens Problem



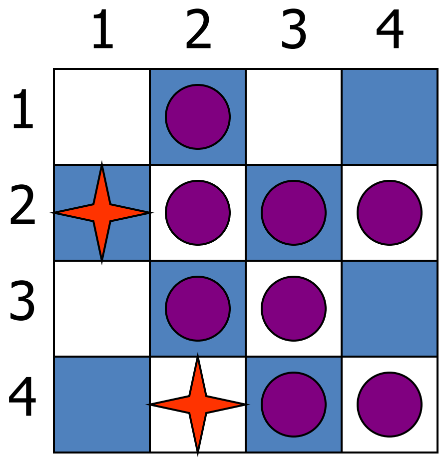


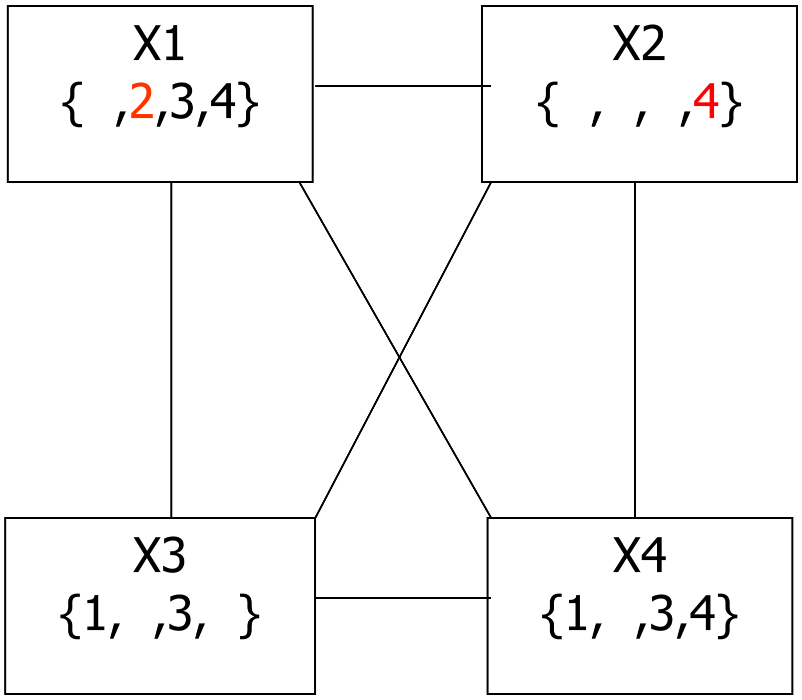
Example: 4-Queens Problem



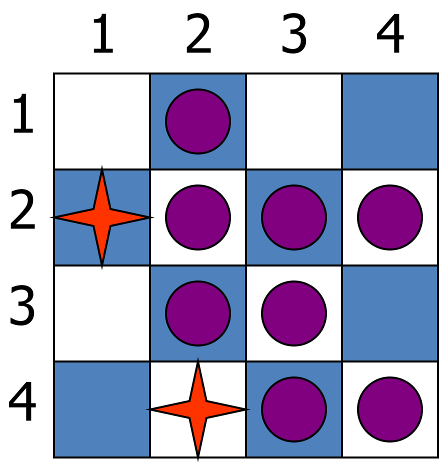


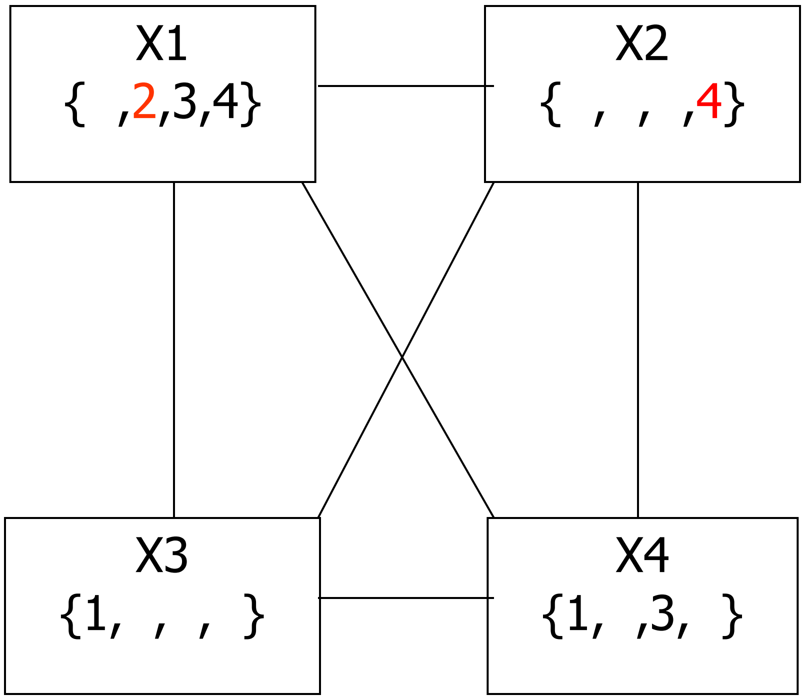
Example: 4-Queens Problem



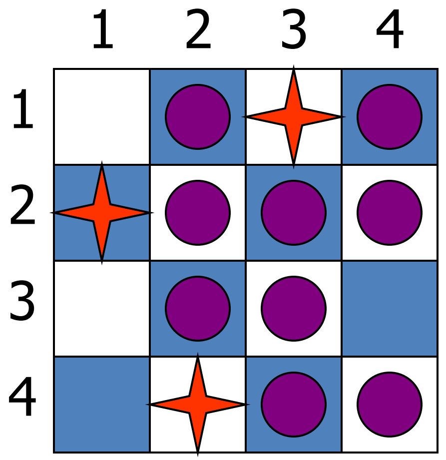


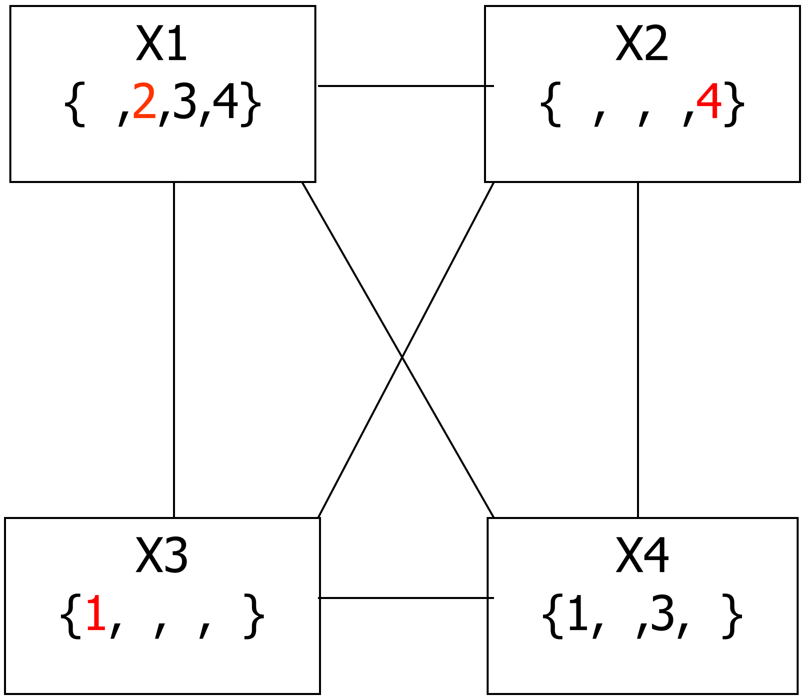
Example: 4-Queens Problem



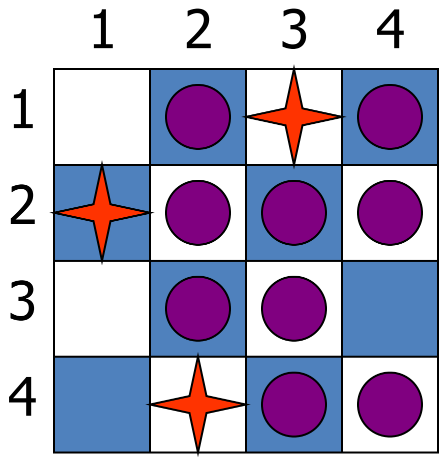


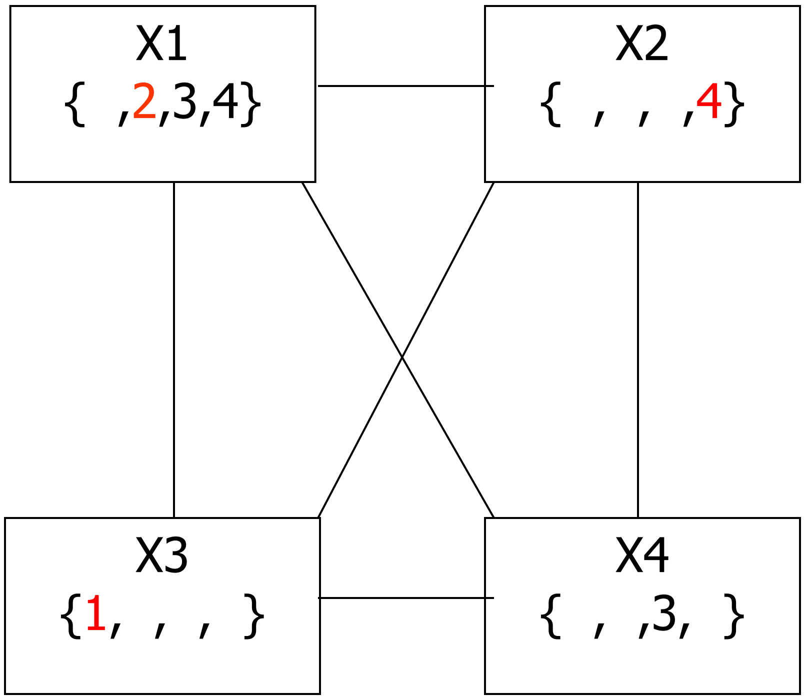
Example: 4-Queens Problem



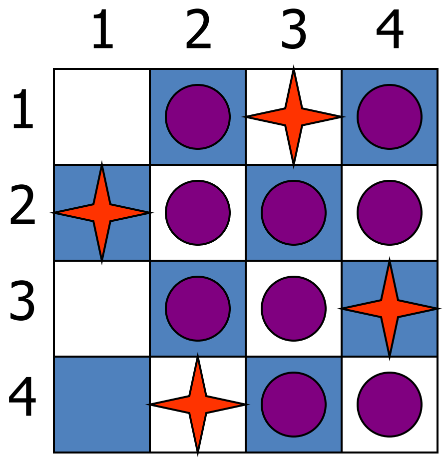


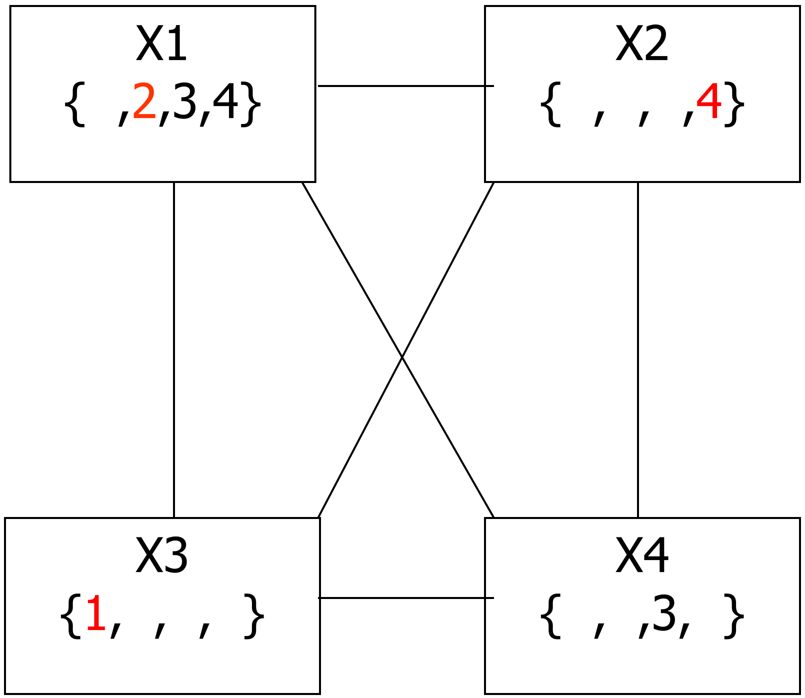
Example: 4-Queens Problem





Example: 4-Queens Problem





Knowledge Based Agents

Humans are best at understanding, reasoning, and interpreting knowledge. Human knows things, which is knowledge and as per their knowledge they perform various actions in the real world.

But how machines do all these things comes under knowledge representation and reasoning.

Logic - Introduction

Sentences in natural language	Sentence in Logic
Socrates is a man	man(socrates)
Plato is a man	man(plato)
All men are mortal	$\forall X$ (man(X) → mortal(X))
Conclusions	
Socrates is mortal: mortal(Socrates)	
Plato is mortal: mortal(Plato)	

Ensure that all actions performed by computer are justifiable ("rational")

Deduction

In AI we need to create new facts from the existing facts. In propositional logic, the process is called deduction. Given two presumably *true* sentences, we can deduce a new *true* sentence. For example:

Either he is at home or at the office	Premise 1:
He is not at home	Premise 2:
Therefore, he is at the office	Conclusion

If we use H for "he is at home", O for "he is at office" and the symbol |– for the "therefore", then we can show the above argument as:

$$\{H \vee O, \ \neg H\} \ \vdash \ O$$

Knowledge-based agents

- Knowledge and reasoning are also important for artificial agents because they enable successful behaviours that would be very hard to achieve.
- KBA can combine general knowledge with current percept's to infer hidden aspects of the current state prior to selecting actions

Knowledge-based agents

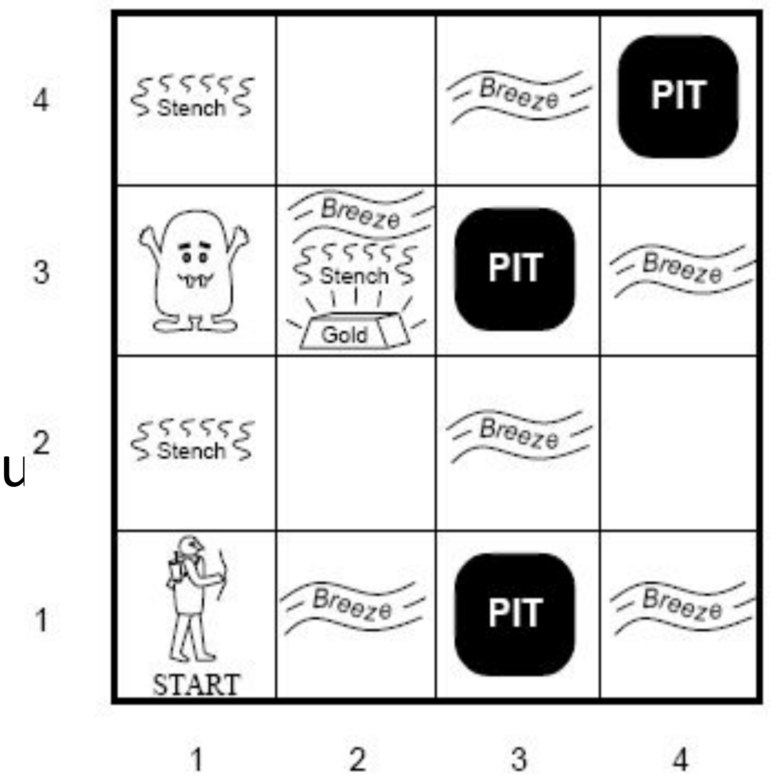
- KB = knowledge base
  - A set of sentences or facts
  - e.g., a set of statements in a logic language

- Inference
  - Deriving new sentences from old
  - e.g., using a set of logical statements to infer new ones

- Knowledge-based agents can benefit from knowledge expressed in very general forms, combining and recombining information to suit myriad purposes.
- **knowledge base**, or KB
- knowledge base is a set of **sentences**.
- Each sentence is expressed in a language called a **knowledge representation language**

Example: The WUMPUS WORLD

- Environment
  - –Cave of 4×4
  - –Agent enters in [1,1]
  - –16 rooms
  - –Wumpus: A deadly beast who kills anyone entering his room.
  - –Pits: Bottomless pits that will trap you forever.
  - –Gold



Agent in a Wumpus world: Percepts

- The agent perceives
  - a stench in the square containing the wumpus and in the adjacent squares (not diagonally)
  - a breeze in the squares adjacent to a pit
  - a glitter in the square where the gold is
  - a bump, if it walks into a wall
  - a woeful scream everywhere in the cave, if the wumpus is killed
- The percepts will be given as a five-symbol list:
  - If there is a stench, and a breeze, but no glitter, no bump, and no scream, the percept is

    [Stench, Breeze, None, None, None]
- The agent can not perceive its own location.

The actions of the agent in Wumpus game are:

- **go forward**
- **turn right** 90 degrees
- **turn left** 90 degrees
- **grab** means pick up an object that is in the same square as the agent
- **shoot** means fire an arrow in a straight line in the direction the agent is looking.
  - The arrow continues until it either hits and kills the wumpus or hits the wall.
  - The agent has only one arrow.
  - Only the first shot has any effect.
- **climb** is used to leave the cave.
  - Only effective in start field.
- **die**, if the agent enters a square with a pit or a live wumpus.
  - (No take-backs!)

The agent's goal

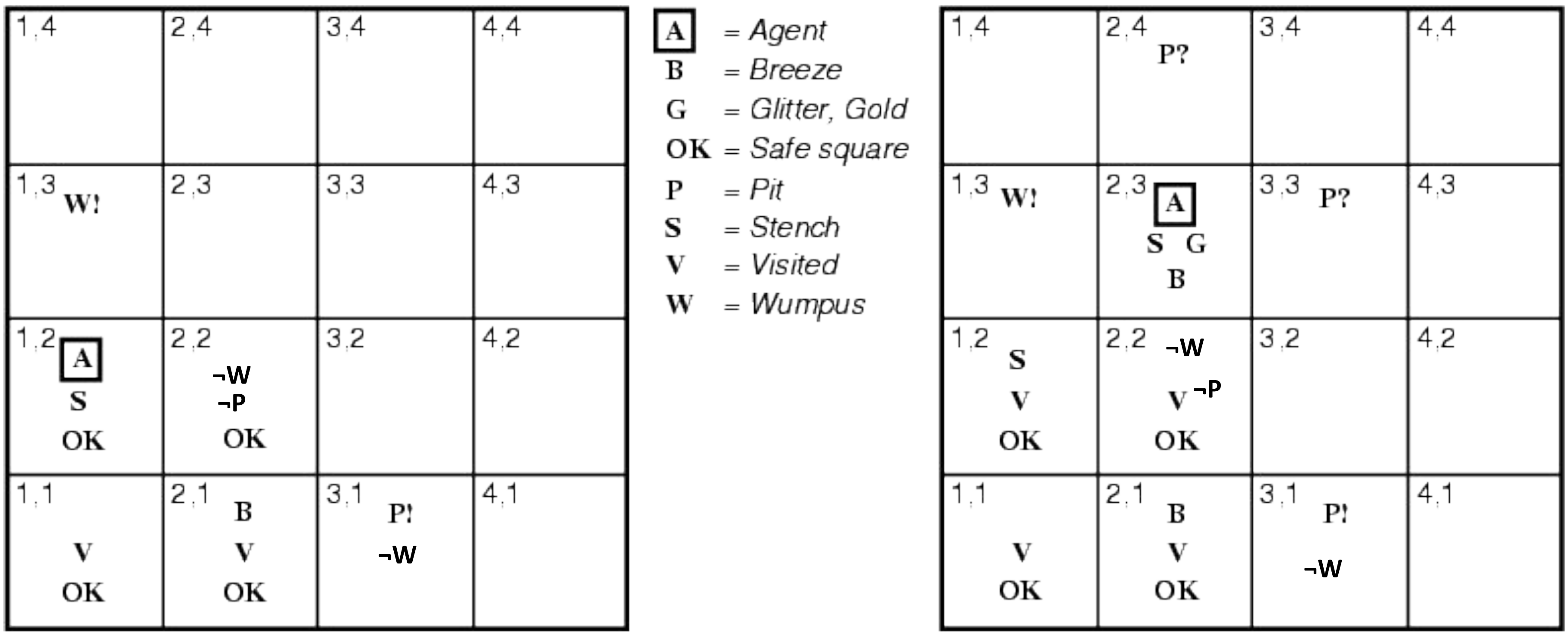
The agent's goal is to find the gold and bring it back to the start as quickly as possible, without getting killed.

- 1000 points reward for climbing out of the cave with the gold
- 1 point deducted for every action taken
- 10000 points penalty for getting killed

The Wumpus agent's first step

(a) (b)

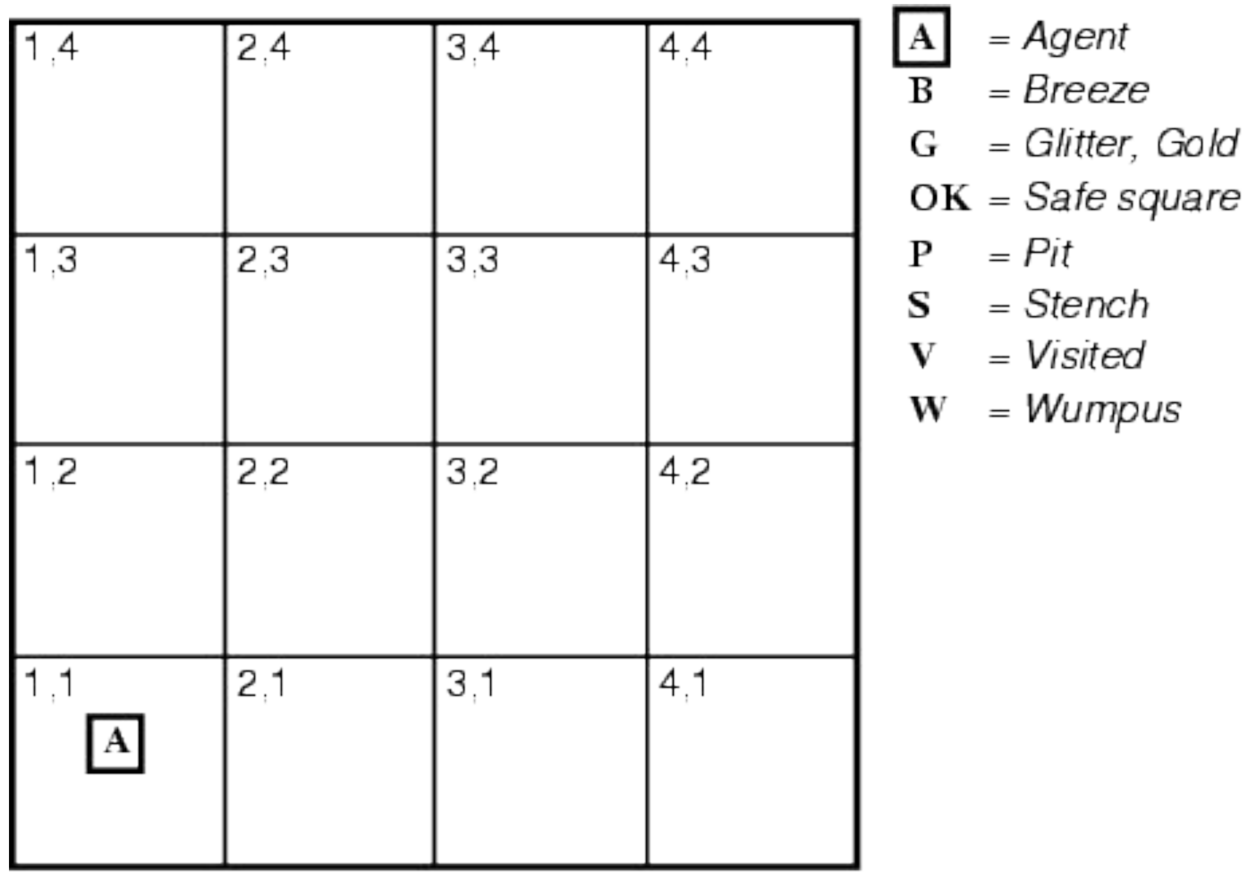
Later



1,4
2,4
3,4
4,4
1,3 W!
2,3
3,3
4,3
1,2 A S OK
2,2 ¬W ¬P OK
3,2
4,2
1,1 V OK
2,1 B V OK
3,1 P! ¬W
4,1
A = Agent
B = Breeze
G = Glitter, Gold
OK = Safe square
P = Pit
S = Stench
V = Visited
W = Wumpus
1,4
2,4 P?
3,4
4,4
1,3 W!
2,3 A S G B
3,3 P?
4,3
1,2 S V OK
2,2 ¬W V ¬P OK
3,2
4,2
1,1 V OK
2,1 B V OK
3,1 P! ¬W
4,1

(a) (b)

Let's Play!



1,4
2,4
3,4
4,4
1,3
2,3
3,3
4,3
1,2
2,2
3,2
4,2
1,1
A
2,1
3,1
4,1
A = Agent
B = Breeze
G = Glitter, Gold
OK = Safe square
P = Pit
S = Stench
V = Visited
W = Wumpus

Representation, reasoning, and logic

- The object of *knowledge representation* is to express knowledge in a **computer-tractable form**, so that agents can perform well.
- A knowledge representation language is defined by:
  - its **syntax,** which defines all possible sequences of symbols that constitute sentences of the language.
  - Syntax – what expressions are legal (well-formed)
  - Examples: Sentences in a book, bit patterns in computer memory.
  - its **semantics**, which determines the facts in the world to which the sentences refer.
  - Each sentence makes a claim about the world.
  - An agent is said to believe a sentence about the world.
  - Semantics – what legal expressions mean

LOGIC

- Logics are formal languages for representing information such that conclusions can be drawn
- Syntax defines the sentences in the language
- Ex:
  - X + Y = 4 - Sentence
  - X2Y+ is not a sentence
- Sentences in KB are real world scenarios.
- Semantics define the meaning of sentences;
  i.e., truth of a sentence in a world

Inference Procedures: Rules for deriving new sentences from existing sentences. Operates on the syntax; to be useful must respect what holds in the world

PROPOSITIONAL LOGIC: A VERY SIMPLE LOGIC

Syntax:

- The syntax of propositional logic defines the allowable sentences.
- Each sentence is represented by a propositional symbol.
- We use Uppercase letters for symbols:

  P, Q, R ..
- We can write W1,3 for wumpus in [1,3] state

- Proposition – two types (True/False)
- Complex sentences are constructed from simpler sentences using Logical Connectives.
- Five Connectives:

˜ $\neg$ (not)
˜ $\wedge$ (and)
˜ $\vee$ (or)
˜ $\Rightarrow$ (implies)
˜ $\Leftrightarrow$ (if and only if)

- Formal grammar of propositional logic

$$
\begin{aligned}
\mathit{Sentence} &\rightarrow \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} &\rightarrow \mathbf{True} \mid \mathbf{False} \mid \mathit{Symbol} \\
\mathit{Symbol} &\rightarrow \mathbf{P} \mid \mathbf{Q} \mid \mathbf{R} \mid \ldots \\
\mathit{ComplexSentence} &\rightarrow \neg\, \mathit{Sentence} \\
&\mid \ (\, \mathit{Sentence} \wedge \mathit{Sentence}\,) \\
&\mid \ (\, \mathit{Sentence} \vee \mathit{Sentence}\,) \\
&\mid \ (\, \mathit{Sentence} \Rightarrow \mathit{Sentence}\,) \\
&\mid \ (\, \mathit{Sentence} \Leftrightarrow \mathit{Sentence}\,)
\end{aligned}
$$

Propositional logic: Syntax

- Propositional logic is the simplest logic – illustrates basic ideas
- The proposition symbols S1, S2 etc are sentences
  - If S is a sentence, ¬S is a sentence (negation)
  - If S1 and S2 are sentences, S1 ∧ S2 is a sentence (conjunction)
  - If S1 and S2 are sentences, S1 ∨ S2 is a sentence (disjunction)
  - If S1 and S2 are sentences, S1 ⇒ S2 is a sentence (implication)
  - If S1 and S2 are sentences, S1 ⇔ S2 is a sentence (biconditional)

Propositional logic: Semantics

Semantics: The rules for whether a sentence is true or false

E.g. P1,2 P2,2 P3,1

false true false

With these symbols, 8 possible models, can be enumerated automatically.

Rules for evaluating truth with respect to a model *m*:

¬S is true iff S is false

S1 ∧ S2 is true iff S1 is true and S2 is true

S1 ∨ S2 is true iff S1 is true or S2 is true

S1 ⇒ S2 is true iff S1 is false or S2 is true

i.e., is false iff S1 is true and S2 is false

S1 ⇔ S2 is true iff S1 ⇒ S2 is true and S2 ⇒ S1 is true

Simple recursive process evaluates an arbitrary sentence, e.g.,

P	Q	$\neg P$	$P \land Q$	$P \lor Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
$false$	$false$	$true$	$false$	$false$	$true$	$true$
$false$	$true$	$true$	$false$	$true$	$true$	$false$
$true$	$false$	$false$	$false$	$true$	$false$	$false$
$true$	$true$	$false$	$true$	$true$	$true$	$true$

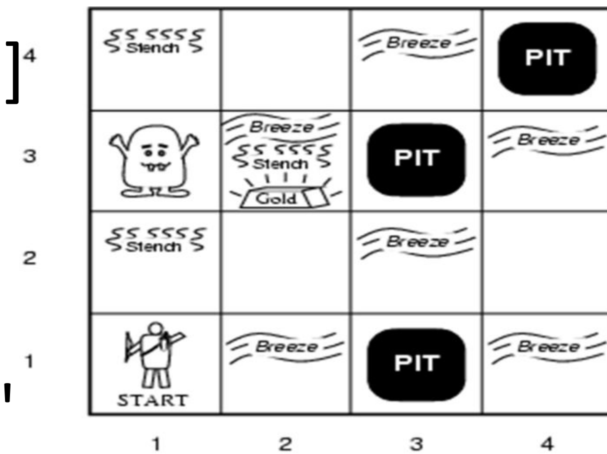
Inference rules in propositional logic

- $(A \wedge B) \equiv (B \wedge A)$ $\wedge$ is commutative
- $(A \vee B) \equiv (B \vee A)$ $\vee$ is commutative
- $((A \wedge B) \wedge C) \equiv (A \wedge (B \wedge C))$ $\wedge$ is associative
- $((A \vee B) \vee C) \equiv (A \vee (B \vee C))$ $\vee$ is associative
- $\neg(\neg A) \equiv A$ Double-negation elimination
- $(A \Rightarrow B) \equiv (\neg B \Rightarrow \neg A)$ Contraposition
- $(A \Rightarrow B) \equiv (\neg A \vee B)$ Implication elimination
- $(A \Leftrightarrow B) \equiv ((A \Rightarrow B) \wedge (B \Rightarrow A))$ Biconditional elimination
- $\neg(A \wedge B) \equiv (\neg A \vee \neg B)$ ”De Morgan”
- $\neg(A \vee B) \equiv (\neg A \wedge \neg B)$ ”De Morgan”
- $(A \wedge (B \vee C)) \equiv ((A \wedge B) \vee (A \wedge C))$ Distributivity of $\wedge$ over $\vee$
- $(A \vee (B \wedge C)) \equiv ((A \vee B) \wedge (A \vee C))$ Distributivity of $\vee$ over $\wedge$

A simple knowledge base

Wumpus world sentences

Let $P_{i,j}$ be true if there is a pit in [i, j].
Let $B_{i,j}$ be true if there is a breeze in [i, j]



$\neg P_{1,1}$
$\neg B_{1,1}$
$B_{2,1}$

- "Pits cause breezes in adjacent squares"
  - $B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})$
  - $B_{2,1} \Leftrightarrow (P_{1,1} \lor P_{2,2} \lor P_{3,1})$

Propositional Logic - PL

- PL deals with
  - the validity, satisfiability and unsatisfiability of a formula
  - derivation of a new formula using equivalence laws.
- Each row of a truth table for a given formula is called its **interpretation** under which a formula can be true or false.
- A formula α is called **tautology** if and only
  - if α is true for all interpretations.
- A formula α is also called **valid** if and only if
  - it is a **tautology.**

Proof in Wumpus KB

$B_{11} \Leftrightarrow (P_{12} \vee P_{21})$	Rule of the game
$B_{11} \Rightarrow (P_{12} \vee P_{21}) \wedge (P_{12} \vee P_{21}) \Rightarrow B_{11}$	Biconditional elimination
$(P_{12} \vee P_{21}) \Rightarrow B_{11}$	And elimination
$\neg B_{11} \Rightarrow \neg(P_{12} \vee P_{21})$	Contraposition
$\neg B_{11} \Rightarrow \neg P_{12} \wedge \neg P_{21}$	"De Morgan"

Thus, we have proven, in four steps, that no breeze in (1,1) means there can be no pit in either (1,2) or (2,1)

Some Propositional Rules for the wumpus world:

(R1) $\neg S_{11} \Rightarrow \neg W_{11} \wedge \neg W_{12} \wedge \neg W_{21}$

(R2) $\neg S_{21} \Rightarrow \neg W_{11} \wedge \neg W_{21} \wedge \neg W_{22} \wedge \neg W_{31}$

(R3) $\neg S_{12} \Rightarrow \neg W_{11} \wedge \neg W_{12} \wedge \neg W_{22} \wedge \neg W_{13}$

(R4) $S_{12} \Rightarrow W_{13} \vee W_{12} \vee W_{22} \vee W_{11}$

Representation of Knowledgebase for Wumpus world:

Following is the Simple KB for wumpus world when an agent moves from room [1, 1], to room [2,1]:

$\neg W_{11}$	$\neg S_{11}$	$\neg P_{11}$	$\neg B_{11}$	$\neg G_{11}$	V_{11}	OK_{11}
$\neg W_{12}$	----	$\neg P_{12}$	-----	----	$\neg V_{12}$	OK_{12}
$\neg W_{21}$	$\neg S_{21}$	$\neg P_{21}$	B_{21}	$\neg G_{21}$	V_{21}	OK_{21}

First-order logic

- We saw how propositional logic can create intelligent behavior
- But propositional logic is a poor representation for complex environments
- First-order logic is a more expressive and powerful representation

What don't we like about propositional logic?

- Lacks expressive power to describe the environment concisely
  - Separate rules for every square/square relationship in Wumpus world

Syntax of FOL: Basic elements

- Constants KingJohn, 2, NUS,...
- Predicates Brother, >,...
- Functions Sqrt, LeftLegOf,...
- Variables x, y, a, b,...
- Connectives $\neg$, $\Rightarrow$, $\wedge$, $\vee$, $\Leftrightarrow$
- Equality =
- Quantifiers $\forall$, $\exists$

Quantifiers

- A way to express properties of entire collections of objects
- FOL contains two standard quantifiers
  - Universal**(for all, everyone, everything)**
  - Existential**(for some, at least one).**

Universal Quantification: Syntax

Definition: The symbol $\forall$ is called the universal quantifier. The universal quantification of P(x) is the statement
"P(x) for all values x in the universe", which is written in logical notation as:
$\forall x P(x)$ or sometimes $\forall x \in D, P(x)$.

Example

"Everyone studying in Koblenz is smart:

$$\forall \underbrace{x}_{\text{variables}} \underbrace{(StudiesAt(x, Koblenz) \Rightarrow Smart(x))}_{\text{sentence}}$$

Universal quantification contd..

- Given some propositional function P(x)
- And values in the universe $x_1 .. x_n$
- The universal quantification $\forall x\, P(x)$ implies:

$$P(x_1) \wedge P(x_2) \wedge \ldots \wedge P(x_n)$$

Existential Quantification: Syntax

Definition: The symbol $\exists$ is called the existential quantifier and represents the phrase "there exists" or "for some".

The existential quantification of P(x) is the statement "P(x) for some values x in the universe", or equivalently, "There exists a value for x such that

P(x) is true", which is written $\exists x P(x)$.

Existential quantification

- Represented by backwards E: $\exists$
  - It means "there exists"
  - Let P(x) = x+1 > x
- We can state the following:
  - $\exists x\, P(x)$
  - Meaning:"there exists (a value of) x such that P(x) is true"
  - "for at least one value of x, x+1>x is true"

Existential quantification contd…

- Given some propositional function P(x)
- And values in the universe $x_1 .. x_n$
- The existential quantification $\exists x\, P(x)$ implies:

$$P(x_1) \vee P(x_2) \vee \ldots \vee P(x_n)$$

Example

Definition for the Breezy predicate:

If a square is breezy, some adjacent square must contain a pit

$\forall y\ Breezy(y) \Rightarrow \exists x\ Pit(x) \wedge Adjacent(x, y)$

If a square is not breezy, no adjacent pit contains a pit

$\forall y\ \neg Breezy(y) \Rightarrow \neg \exists x\ Pit(x) \wedge Adjacent(x, y)$

Examples

- Some Examples of FOL using quantifier:
- **1. All birds fly.**
  In this question the predicate is "**fly(bird)**."
  And since there are all birds who fly so it will be represented as follows.
  $\forall$x bird(x) →fly(x).
- **2. Every man respects his parent.**
  In this question, the predicate is "**respect(x, y)," where x=man, and y= parent**.
  Since there is every man so will use $\forall$, and it will be represented as follows:
  $\forall$x man(x) → respects (x, parent).
- **3. Some boys play cricket.**
  In this question, the predicate is "**play(x, y)**," where x= boys, and y= game. Since there are some boys so we will use $\exists$**, and it will be represented as**:
  $\exists$x boys(x) → play(x, cricket).
- **4. Not all students like both Mathematics and Science.**
  In this question, the predicate is "**like(x, y)," where x= student, and y= subject**.
  Since there are not all students, so we will use $\forall$ **with negation, so** following representation for this:
  ¬$\forall$ (x) [student(x) → like(x, Mathematics) $\wedge$ like(x, Science)].
- **5. Only one student failed in Mathematics.**
  In this question, the predicate is "**failed(x, y)," where x= student, and y= subject**.
  Since there is only one student who failed in Mathematics, so we will use following representation for this:
  $\exists$(x) [student(x) → failed (x, Mathematics) $\wedge$ $\forall$ (y) [¬(x==y) $\wedge$ student(y) → ¬failed (x, Mathematics)].

Inferences in First Order Logic

- We know how to do inference in propositional logic, but not in first-order logic.

- We use simple inference rules to convert quantified statements to propositional sentences

Inference rules

- **Universal elimination:**
  - $\forall$ x Likes(x, IceCream) with the substitution {x / Einstein} gives us Likes(Einstein, IceCream)
  - The substitution has to be done by a ground term
- **Existential elimination:**
  - From $\exists$ x Likes(x, IceCream) we infer Likes(Man, IceCream) as long as Man does not appear elsewhere in the Knowledge base
- **Existential introduction:**
  - From Likes(Monalisa, IceCream) we can infer

    $\exists$ x Likes(x, IceCream)

Uncertain and probabilistic reasoning - Basic Probability Notation

ACTING UNDER UNCERTAINTY

- When an agent knows enough facts about its environment, the logical approach enables it to derive plans that are guaranteed to work.
- *Agents almost never have access to the whole truth about their environment. Agents must, therefore, act under* ***uncertainty-*** **the state of being unsure of something**
- wumpus agent - unable to discover which of two squares contains a pit

Uncertainty

Let action A_t = leave for airport t minutes before flight

Will A_t get me there on time?

Problems:

partial observability (road state, other drivers' plans, etc.)

1. noisy sensors (traffic reports)
2. uncertainty in action outcomes (flat tire, etc.)
3. immense complexity of modeling and predicting traffic

"A_{25} will get me there on time if there's no accident on the bridge and it doesn't rain and my tires remain intact etc "

(A_{1440} might reasonably be said to get me there on time but I'd have to stay overnight in the airport …)

- The world is not a well-defined place.
- There is uncertainty in the facts we know:
  - What’s the temperature? Imprecise measures
  - Is trump a good president? Imprecise definitions
  - Where is the pit? Imprecise knowledge

Sources of Uncertainty

- **Uncertain data**
  - missing data, unreliable, ambiguous, imprecise representation, inconsistent, subjective, derived from defaults, noisy…
- **Uncertain knowledge representation**
  - restricted model of the real system
  - limited expressiveness of the representation mechanism
- **inference process**
  - Derived result is formally correct, but wrong in the real world
  - New conclusions are not well-founded (eg, inductive reasoning)
  - Incomplete, default reasoning methods

Using FOL for (Medical) Diagnosis

$\forall p\ Symptom(p, Toothache) \Rightarrow Disease(p, Cavity)$

Not correct...

$\forall p\ Symptom(p, Toothache) \Rightarrow Disease(p, Cavity) \lor Disease(p, GumDisease) \lor Disease(p, WisdomTooth)$

Not complete...

$\forall p\ Disease(p, Cavity) \Rightarrow Symptom(p, Toothache)$

Not correct...

The main tool for dealing with degrees of belief is **probability theory** , which assigns to each sentence a numerical degree of belief between 0 and 1.

Handling Uncertain Knowledge

Problems using first-order logic for diagnosis:

Laziness:
Too much work to make complete rules.
Too much work to use them

Theoretical ignorance:
Complete theories are rare

Practical ignorance:
We can't run all tests anyway

Probability can be used to *summarize* the laziness and ignorance !

Handling uncertain knowledge

- The sentence itself is *in fact* either **true** or **false**.
- A degree of belief **is different from a degree of truth**.
- A probability of 0.8 does not mean “80% true”, but rather an 80% degree of belief that something is true.

Probability

- A well-known and well-understood framework for dealing with uncertainty
- Has a clear semantics
- Provides principled answers for:
  - Combining evidence
  - Predictive and diagnostic reasoning
  - Incorporation of new evidence
- Can be learned from data

Probability

Compare the following:

1) First-order logic:
   "The patient has a cavity"
2) Probabilistic:
   "The probability that the patient has a cavity is 0.8"

1) Is either valid or not, depending on the state of the world
2) Validity depends on the agents perception history, the evidence

Representing Uncertain Knowledge: Probability

- Probabilities provide us with a way of assigning **degrees of belief** in a sentence.
- Probability is a way of **summarizing the uncertainty** regarding a situation.
- The exact probability assigned to a sentence depends on existing **evidence**: the knowledge available up to date.
- Probabilities can change when **more evidence** is acquired.

Probability

Ontological commitments: Facts hold or do not hold in the world.

Epistemological commitments: A probabilistic agent has a **degree of belief** in a particular sentence. Degrees of belief range from 0 (for sentences that are certainly false) to 1 (for sentences that are certainly true).

Conditional Probability

The Posterior prob. (conditional prob.) after obtaining evidence:

Notation:

P(A|B) means:

"The probability of A given that all we know is B". Example:

P(Sunny | Summer) = 0.65

Is defined as:

$$\boxed{P(A \mid B) = \frac{P(A \wedge B)}{P(B)}}$$

Can be rewritten as the product rule:

$P(A \wedge B) = P(A|B)P(B) = P(B|A)P(A)$

"For A and B to be true,

B has to be true, and A has to be true given B"

Bayes' Rule

The left side of the product rule is symmetric w.r.t B and A:

$$P(A \wedge B) = P(A)P(B \mid A)$$
$$P(A \wedge B) = P(B)P(A \mid B)$$

Equating the two right-hand sides yields Bayes' rule:

$$P(B \mid A) = \frac{P(A \mid B)P(B)}{P(A)}$$

Bayes' Rule

Useful for assessing diagnostic probability from causal probability:

$$P(\text{Cause}|\text{Effect}) = \frac{P(\text{Effect}|\text{Cause})P(\text{Cause})}{P(\text{Effect})}$$

Example: Medical Diagnosis

- You go to the doctor complaining about the symptom of having a fever (*evidence*).
- The doctor knows that bird flu causes a fever 95% of the time.
- The doctor knows that if a person is selected randomly from the population, there is a 10^{-7} chance of the person having bird flu.
- In general, 1 in 100 people in the population suffer from fever.
- What is the probability that you have bird flu (*hypothesis*)?

$$P(H|e) = \frac{P(e|H)P(H)}{P(e)} = \frac{0.95 \times 10^{-7}}{0.01} = 0.95 \times 10^{-5}$$

Bayes' Rule

In distribution form

$$\mathbf{P}(Y|X) = \frac{\mathbf{P}(X|Y)\,\mathbf{P}(Y)}{\mathbf{P}(X)} = \alpha\mathbf{P}(X|Y)\,\mathbf{P}(Y)$$