AI

Search Problem

The trade-off usually between:

- Speed (Time complexity)
- Memory (Space complexity)
- 'Optimality' (Optimal)

Four characteristics:

- Time complexity: number of nodes generated
- Space complexity: maximum number of nodes in memory
- Optimality: does it always find a least-cost solution?
- completeness: does it always find a solution if one exists?

b: maximum branching factor of the search tree

d: depth of the least-cost solution

m: maximum depth of the state space (may be ∞)

Uninformed search strategies

Breadth-first search

- Complete: Yes
- Time: $O(b^{d+1})$
- Space: $O(b^{d+1})$
- Optimal: Yes
- **Space** is the bigger problem (More than time)

Depth-first search

- Complete: No
- Time: $O(b^m)$, terrible if m is much larger than d
- Space: O(bm), linear space
- Optimal: No

Iterative deepening search

- Complete: Yes
- Time: $O(b^d)$
- Space: O(bd) linear space
- Optimal: Yes

Heuristic Search

Best first search:

Idea: Use an **evaluation function** $f(n)$ for each node

special case: Greedy Best First Search, A* Search

Greedy best-first search

- Evaluation function $f(n)$ = $h(n)$ (**h**euristic) = estimate of cost from n to goal
- Complete: No
- Time: $O(b^m)$
- Space: $O(b^m)$
- Optimal: No

A* search

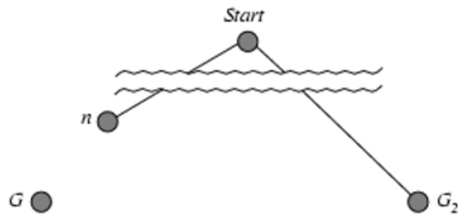
- Idea: avoid expending paths that are already expensive
- Evaluation function $f(n) = g(n) + h(n)$
  - $g(n)$: cost so far (to reach n)
  - $h(n)$: estimated cost from n to goal
  - $f(n)$: estimated total cost of path through *n* to goal
- **Admissible heuristics** for A*:

- A heuristic *h(n)* is **admissible** if for every node *n*, $h(n) \le h^*(n)$, where $h^*(n)$ is the **true** cost to reach the goal state from *n*.
- An admissible heuristic never overestimates the cost to reach the goal
- **Theorem:** If *h(n)* is admissible, A* using TREE-SEARCH is optimal

* Complete: Yes
* Time: Exponential (in length of optimal solution)
* Space: Keeps all (expanded) nodes in memory
* Optimal: Yes

Optimality of A^* (proof)

- Suppose some suboptimal goal state G_2 has been generated and is in the fringe. Let n be an unexpanded node in the fringe such that n is on a shortest path to an optimal goal G with true cost C^*.



$f(n) = g(n)+h(n)$
(from $h(x) \le g(x)$ and $g(G) = C^*$)
$f(n) \le C^*$

$f(G_2) = g(G_2)+h(G_2)$
(from $h(x) \le g(x)$ and G_2 is a goal state)
$f(G_2)=g(G_2)$
(from suboptimality of G_2)
$f(G_2)>C^*$

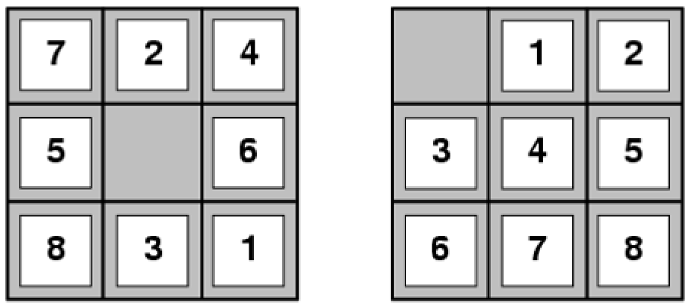
$\therefore$ $f(n) \le f(G_2)$, and A^* will never select G_2 for expansion

Admissible heuristics

E.g., for the 8-puzzle:

- $h_1(n)$ = number of misplaced tiles
- $h_2(n)$ = total Manhattan distance

(i.e., no. of squares from desired location of each tile)



Start State Goal State

- $h_1(S)$ = ? 8
- $h_2(S)$ = ? 3+1+2+2+2+3+3+2 = 18

Dominance

- If $h_2(n) \geq h_1(n)$ for all n (both admissible)
- then h_2 dominates h_1
- h_2 is better for search

- Typical search costs (average number of nodes expanded = effective branching factor):

- $d=12$ IDS = 3,644,035 nodes
  $A^*(h_1)$ = 227 nodes
  $A^*(h_2)$ = 73 nodes
- $d=24$ IDS = too many nodes
  $A^*(h_1)$ = 39,135 nodes
  $A^*(h_2)$ = 1,641 nodes

Constraint satisfaction problem (CSP)

- **state** is defined by **variables** X_i with **values** from **domain** D_i
- **goal test** is a set of **constraints** specifying allowable combinations of values for subsets of variables

Which variable should be assigned next?

*Most constrained variable （选之后的点用这个）

- choose the variable with the fewest legal values,
- as known as minimum remaining values (MRV) heuristic
- Because these are the variables that are most likely to prune the search tree

Most constraining variable (如果是选第一个点，用这个)

- Most constraining variable = the variable with the most constraints on remaining variables
- Because the variable involved in the most constraints, so it is most likely to cause prune the search tree

Least constraining value （选涂什么颜色）

- Given a variable, choose the least constraining value:
  - the one that rules out the fewest values in the neighbouring variables – to leave other variables as open as possible

Summary

- CSPs are a special kind of problem:
  - states defined by values of a fixed set of variables
  - goal test defined by constraints on variable values
- Backtracking = depth-first search with one variable assigned per node
- Variable ordering and value selection heuristics help significantly
- Forward checking prevents assignments that guarantee later failure
- Constraint propagation (e.g., arc consistency) does additional work to constrain values and detect inconsistencies
- Iterative min-conflicts is usually effective in practice

Logic

(Not important)

The central component of a knowledge-based agent is their **knowledge base**, or KB.

The agent uses logical reasoning to make decisions.

The agent draws conclusions from the available information.

Propositional Logic

(Just know the concept)

language

To define a knowledge base and define sentences, we need a language

The language of propositional logic is built from

true			
false			
$\wedge$	and	conjunction	(& or .)
$\vee$	or	disjunction	(| or +)
$\neg$	not	negation	($\sim$)
$\Rightarrow$	if . . . then	implication	($\rightarrow$)
$\Leftrightarrow$	if and only if	equivalence	($\leftrightarrow$)

Formulas of propositional logic are defined as follows:

- Each propositional variable $P, Q, R, \ldots$ is a formula.
- **T** and **F** are formulas
- If ϕ is a formula, $\neg\phi$ is a formula
- If ϕ and ψ are formulas so are

$$\phi \wedge \psi, \quad \phi \vee \psi, \quad \phi \Rightarrow \psi, \quad \phi \Leftrightarrow \psi.$$

Propositional variables and constants are called **atomic formulas**. The remaining cases are called **compound (or complex) formulas.**

Axiomatisation

(Important)

Propositional logic is defined by a set of axioms and rules.

Propositional logic has the following axiomatisation:

- $A \implies (B \implies A)$
- $(A \implies (B \implies C)) \implies ((A \implies B) \implies (A \implies C))$
- $(\neg A \implies \neg B) \implies (B \implies A)$
- From A and $A \implies B$, infer B (**Modus Ponens**)
  - which means if A and $A \implies B$ are **true**, then we could infer B is also **true**.
  - $$\boxed{\frac{A \Rightarrow B, A}{B}} \qquad \text{Or, } \boxed{A \Rightarrow B, A \vdash B}$$

    If the antecedent of an implication (a conditional claim) is true, then the consequent must also be true.

Modus Ponens Example

A⇒ B: “If today is Tuesday, then I will go to AI lecture.”
A: “Today is Tuesday.”
B: “I will go to AI lecture.”

A⇒ B: "If it is raining, then there are clouds in the sky."
A: "It is raining."
B: "There are clouds in the sky."

A⇒ B: “If Yankees win today's game, they will be champions.”
A: “Yankees win today's game.”
B: Yankees are champions.

A ⇒ B: “If the weather is good, we can go to the beach.”
A: The weather is good.
B: We can go to the beach.

Proof

Given a KB, a **proof** is a finite sequence of formulas

$$\phi_1, \ldots, \phi_n$$

where each ϕ_i is

- a formula from KB, or
- an instance of an axiom (A1-A3), or
- derived from ϕ_j, ϕ_k, with $j, k < i$ by using modus ponens.

Given a KB and a formula ϕ , we say that ϕ is a **logical consequence** of KB,denoted

$$KB \vdash \phi$$

if there exists a proof of ϕ from KB.

Example: how to prove that

$$\phi \Rightarrow \phi$$

is a theorem.

1. $(\phi \Rightarrow ((\psi \Rightarrow \phi) \Rightarrow \phi)) \Rightarrow ((\phi \Rightarrow (\psi \Rightarrow \phi)) \Rightarrow (\phi \Rightarrow \phi))$
   Instance of A2
2. $\phi \Rightarrow ((\psi \Rightarrow \phi) \Rightarrow \phi)$
   Instance of A1
3. $(\phi \Rightarrow (\psi \Rightarrow \phi)) \Rightarrow (\phi \Rightarrow \phi)$
   From (1) and (2) with modus ponens
4. $\phi \Rightarrow (\psi \Rightarrow \phi)$
   Instance of A1
5. $\phi \Rightarrow \phi$
   From (3) and (4) with modus ponens.

Example

From p ⇒ q, (¬r ∨ q) ⇒ (s ∨ p), q, prove s ∨ q.

1. p ⇒ q	[Given]
2. (¬ r ∨ q) ⇒ (s ∨ p)	[Given]
3. q	[Given]
4. s ∨ q	[3, ∨ introduction]

Exercise

Show r from p ⇒ (q ⇒ r) and p ∧ q using the rules we have been given so far. That is, prove

p ⇒ (q ⇒ r), p ∧ q ⊢ r.

1 p ⇒ (q ⇒ r)	[Given]
2.p ∧ q	[Given]
3. q	[2, ∧-Introduction]
4. p	[2, ∧-Introduction]
5. q ⇒ r	[1,4, modus ponens]
6. r	[3, modus ponens]

Let *Prop.* be a set of propositional variables. A valuation v is a mapping :

$$v : Prop \to \{0, 1\}$$

A valuation assigns to each propositional variable either 1 (true) or 0 (false)

Negations:

$$v(\neg\phi) = \begin{cases} 1 & v(\phi) = 0 \\ 0 & \text{otherwise} \end{cases}$$

$\neg\phi$ is true IFF ϕ is false.

negation ¬

p	¬p
T	F
F	T

Conjunctions:

$$v(\phi \wedge \psi) = \begin{cases} 1 & v(\phi) = v(\psi) = 1 \\ 0 & \text{otherwise} \end{cases}$$

$\phi \wedge \psi$ is true IFF both ϕ and ψ are true

Conjunction ∧

p	q	p ∧ q
T	T	T
T	F	F
F	T	F
F	F	F

Disjunction:

$$v(\phi \vee \psi) = \begin{cases} 0 & v(\phi) = v(\psi) = 0 \\ 1 & \text{otherwise} \end{cases}$$

$\phi \vee \psi$ is true IFF at least one formula in $\{\phi, \psi\}$ is true.

Disjunction ∨

p	q	p ∨ q
T	T	T
T	F	T
F	T	T
F	F	F

Implications:

$$v(\phi \Rightarrow \psi) = \begin{cases} 0 & v(\phi) = 1,\ v(\psi) = 0 \\ 1 & \text{otherwise} \end{cases}$$

$\phi \Rightarrow \psi$ is true IFF either ϕ is false or both ϕ and ψ are true.

Implication ⇒ (If...then...)

p	**q**	**p ⇒ q**
T	T	T
T	F	F
F	T	T
F	F	T

p ⇒ q:

converse: q ⇒ p

contrapositive: ¬ q ⇒ ¬ p

inverse: ¬ p ⇒ ¬ q

Double Implication:

$$v(\phi \Leftrightarrow \psi) = \begin{cases} 1 & v(\phi) = v(\psi) \\ 0 & \text{otherwise} \end{cases}$$

$\phi \Leftrightarrow \psi$ is true IFF ϕ and ψ have the same truth-value.

Double Implication ⇔ (if and only if...)

p	**q**	**p ⇔ q**
T	T	T
T	F	F
F	T	F
F	F	T

Truth Table

A truth-table for a formula is just a representation of all the possible assignments of values and the corresponding outputs.

ϕ	ψ	$\neg\phi$	$\phi \wedge \psi$	$\phi \vee \psi$	$\phi \Rightarrow \psi$	$\phi \Leftrightarrow \psi$
1	1	0	1	1	1	1
1	0	0	0	1	0	0
0	1	1	0	1	1	0
0	0	1	0	0	1	1

Example

p
(my breakfast is) eggs.
q
(my breakfast is) cereal.
r
(my breakfast is) toast.

p	q	r	$p \vee q$	$r \wedge (p \vee q)$
T	T	T	T	T
T	T	F	T	F
T	F	T	T	T
T	F	F	T	F
F	T	T	T	T
F	T	F	T	F
F	F	T	F	F
F	F	F	F	F

The statement 'my breakfast is toast and either eggs or cereal' in symbolic form as r ∧ (p ∨ q)

Valuation

We can check whether a formula is true under a given valuation by looking at its truth-table.

- A formula ϕ is called **satisfiable** if there exists a valuation v such that $v(\phi) = 1$.
  - "a sentence is satisfiable if it is **True in some** model"
- A formula ϕ is called a **tautology** if for all possible valuations v, $v(\phi) = 1$.
  - a formula is said to be tautology if and only if it is **true under every interpretation**.
- A formula ϕ is called a **contradiction** if for all possible valuations v, $v(\phi) = 0$.
  - a sentence is contradiction if it is **false in all** models
- Two formulas are **logically equivalent** if they have the **same truth-table**.
- Given a knowledge base KB, we denote by $Mod(KB)$, the set of all valuations that make all the formulas in KB true.

Examples:

The following truth-table shows that $\phi \Rightarrow \psi$ and $\neg\phi \vee \psi$ are logically equivalent:

ϕ	ψ	$\neg\phi$	$\phi \Rightarrow \psi$	$\neg\phi \vee \psi$
1	1	0	1	1
1	0	0	0	0
0	1	1	1	1
0	0	1	1	1

The following truth-table shows that axiom A1 is a tautology: (easy could prove axioms A2 and A3 are also tautologies)

ϕ	ψ	$\psi \Rightarrow \phi$	$\phi \Rightarrow (\psi \Rightarrow \phi)$
1	1	1	1
1	0	1	1
0	1	0	1
0	0	1	1

Semantic consequence

Given a knowledge base KB and a formula ϕ , we say that ϕ is **semantic consequence** of KB, or that KB **entails** ϕ denoted

$$KB| = \phi$$

if every valuation (model) that makes all the formulas in KB true also makes ϕ true.

Which means:

$$Mod(KB) \subseteq Mod(\phi)$$

the set of models that make all the formulas in KB true is a subset of the set of valuations that make ϕ true.

|- is about deductive inference and proof. It is a syntactical notion.

|= is about relationship between models. It is a semantical notion.

Propositional logic is **sound**:

- If $KB| - \phi$ then $KB| = \phi$
- This means that logical inference preserves truth, i.e. from true premises we can only derive true conclusions.

Propositional logic is also **complete**:

- If $KB| = \phi$ then $KB| - \phi$
- Everything we prove from true premises is true, and every truth can be proven.

The simples algorithm enumerates all the models of KB and ϕ and checks if $Mod(KB) \subseteq Mod(\phi)$.

If KB and ϕ are built from n propositional variables, this means we have to check **2^n models**.

Equivalent Formulae

Examples of inference rules and theorems (which can be derived from A1-A3 and modus ponens)

And-elimination:

$$\text{From } \phi \wedge \psi \text{, derive } \phi.$$

Double negation:

$$\neg\neg\phi \Leftrightarrow \phi$$

De Morgan's Laws:

$$\neg(\phi \wedge \psi) \Leftrightarrow (\neg\phi \vee \neg\psi)$$

$$\neg(\phi \vee \psi) \Leftrightarrow (\neg\phi \wedge \neg\psi)$$

Distributivity Laws:

$$(\phi \wedge (\psi \vee \gamma)) \Leftrightarrow ((\phi \wedge \psi) \vee (\phi \wedge \gamma))$$

$$(\phi \vee (\psi \wedge \gamma)) \Leftrightarrow ((\phi \vee \psi) \wedge (\phi \vee \gamma))$$

Two formulae A and B are equivalent, written $A \equiv B$ if and only if A and B have the **same truth values** for **every interpretation**. (A 为 true, B 也为 true；A为 false，B也为 false) (这里的 $\equiv$ 和课件里面的$\Leftrightarrow$ 同义)

$\wedge$ - elimination

$$\frac{A \wedge B}{A} \quad \text{or} \quad \frac{A \wedge B}{B} \quad \text{Or,} \quad \begin{array}{l} A \wedge B \vdash B \\ A \wedge B \vdash A \end{array}$$

From a conjunction you can infer any of the conjuncts

$\wedge$ - introduction

$$\frac{A, B}{A \wedge B} \qquad A, B \vdash A \wedge B$$

If A holds (true), and B holds, then A $\wedge$ B must also hold.

$\vee$ - introduction

$$\frac{A}{A \vee B} \quad \text{or} \quad \frac{A}{B \vee A} \quad \text{Or} \quad A \vdash A \vee B$$

If A holds (or are provable or true) then A$\vee$B must hold.

$$\frac{A \vee B, \neg B \vee C}{A \vee C}$$

It says that if you know “A or B”, and you know “not be B or C”, then you're allowed to conclude “A or C”.

Contraposition

$(A \Rightarrow B = \neg B \Rightarrow \neg A)$

Associative laws

$(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$

$(A \vee B) \vee C \equiv A \vee (B \vee C)$

Commutative laws

$A \wedge B \equiv B \wedge A$

$A \vee B \equiv B \vee A$

Distributive laws

$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$

$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$

Complement laws

$A \wedge \neg A \equiv F$

$A \vee \neg A \equiv T$

$\neg (\neg A) \equiv A$

de Morgan's laws

$\neg (A \wedge B) \equiv \neg A \vee \neg B$

$\neg (A \vee B) \equiv \neg A \wedge \neg B$

laws for $\Rightarrow$ and $\Leftrightarrow$:

$A \Rightarrow B \equiv \neg A \vee B$

$A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (B \Rightarrow A)$

Example

Prove the following equivalence

$\neg(\neg(P \wedge Q) \vee P) \equiv F$

$\neg(A \wedge B) \equiv \neg A \vee \neg B$

$\neg(\neg(P \wedge Q) \vee P)$	[given]
$\equiv \neg((\neg P \vee \neg Q) \vee P)$	[de Morgan's laws]
$\equiv \neg((\neg Q \vee \neg P) \vee P)$	[Commutative laws]
$\equiv \neg(\neg Q \vee (\neg P \vee P))$	[Associative laws]
$\equiv \neg(\neg Q \vee T)$	[Complement laws]
$\equiv \neg T$	[Complement laws]
$\equiv F$	[Complement laws]

(课件中的example)

Example:

R1: $\neg P_{1,1}$, R2: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$, R3: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$, R4: $\neg B_{1,1}$, R5: $B_{2,1}$

KB = {R1,R2,R3,R4,R5}

prove KB|= $\neg P_{1,2}$, which also means prove KB|- = $\neg P_{1,2}$

We prove that $\neg P_{1,2}$ is a consequence of KB syntactically.

1. $((B_{1,1}) \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow (B_{1,1}))$
   Equivalent to $((B_{1,1}) \Leftrightarrow (P_{1,2} \vee P_{2,1}))$
2. $((P_{1,2} \vee P_{2,1}) \Rightarrow (B_{1,1}))$
   From (1) by And-elimination
3. $(\neg(B_{1,1}) \Rightarrow \neg(P_{1,2} \vee P_{2,1}))$
   From (2) and by modus ponens, A3 and double negation.
4. $\neg(P_{1,2} \vee P_{2,1})$
   From $\neg B_{1,1}$ and modus ponens.
5. $\neg P_{1,2} \wedge \neg P_{2,1}$
   From (4), by De Morgan's laws.
6. $\neg P_{1,2}$
   by And-elimination

Conjunctive Normal Form

A formula is in **Conjunctive Normal Form** if it is of the form

$A_1 \wedge A_2 \wedge \ldots A_k$

where each A_i is **a disjunction of propositions or their negations**.

Example

$(p \vee q) \wedge r \wedge (\neg p \vee \neg r \vee s)$ is in CNF.

$\neg(p \vee q) \wedge r \wedge (\neg p \vee \neg r \vee s)$ is not in CNF.

$(p \vee q) \wedge r \wedge (p \Rightarrow (\neg r \vee s))$ is not in CNF.

Translation into CNF

To translate into CNF, first translate into NNF. Apply **distribution laws** or **commutativity laws** until in the correct form.

Example

$A \Rightarrow B \equiv \neg A \vee B$

Translate $(\neg(p \vee \neg q) \vee r) \Rightarrow p$ into CNF.

We first translate into NNF and obtain

$((p \vee \neg q) \wedge \neg r) \vee p.$

$\neg(A \vee B) \equiv \neg A \wedge \neg B$

$((p \vee \neg q) \wedge \neg r) \vee p$

$((p \vee \neg q) \vee p) \wedge (\neg r \vee p)$

$\neg(A \wedge B) \equiv \neg A \vee \neg B$

$(p \vee \neg q \vee p) \wedge (\neg r \vee p)$

Note the latter could be transformed into

$(p \vee \neg q) \wedge (\neg r \vee p)$

We show how to rewrite $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$ in CNF:

1. $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$
   By definition of $\Leftrightarrow$
2. $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$
   By definition of $\vee$
3. $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$
   By De Morgan laws and double negation
4. $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (B_{1,1} \vee \neg P_{1,2}) \wedge (B_{1,1} \vee \neg P_{2,1})$
   By distributivity

Resolution

Resolution is a proof method for classical **propositional** and **first-order logic**.

Given a formula ϕ, resolution will decide whether the formula is **contradiction** or not.

Propositional resolution works only on expressions in **clausal form**

A **literal** is either an atomic sentence or a negation of an atomic sentence, such as p or ¬p

A **clausal sentence** is either a literal or a disjunction of literals.

- p
- ¬ p
- p ∨ q

Resolution method involves:

- translation to a normal form (CNF);
- At each step, a new clause is derived from two clauses you already have
- Proof steps all use the same rule **resolution rule**
- repeat until false is derived or no new formulae can be derived.

resolution rule

If A or p is true, and B is true or p is false, Then either A or B is true.

$$\boxed{\begin{array}{c} A \vee p \\ B \vee \neg p \\ \hline A \vee B \end{array}}$$

A ∨ B is called the **resolvent**

A ∨ p and B ∨¬ p are called **parents of the resolvent**.

p and ¬ p are called complementary literals.

(Note in the above A or B can be empty)

The resolution algorithm shows that

$$KB| = \phi$$

by showing that (KB∧¬ϕ) is a contradiction and so is unsatisfiable, i.e. there are no models that make (KB∧¬ϕ) true.

The algorithm starts by converting (KB∧¬ϕ) into a CNF.

Then, the resolution rule is applied to the resulting clauses.

Example: {} means a contradiction

Given

$P \vee Q$

$P \Rightarrow R$

$Q \Rightarrow R$

Prove R

1.	$P \vee Q$	
2.	$\neg P \vee R$	
3.	$\neg Q \vee R$	
4.	$\neg R$	negated conclusion
5.	$Q \vee R$	1, 2
6.	$\neg P$	2, 4
7.	$\neg Q$	3, 4
8.	R	5, 7
9.	{ }	4, 8

Using resolution rule, prove $\neg(p \Rightarrow (q \Rightarrow p))$

Step 1: translate to CNF:

$\neg(\neg p \vee (\neg q \vee p))$	I
$\neg\neg p \wedge \neg(\neg q \vee p))$	N
$p \wedge (\neg\neg q \wedge \neg p)$	D
$p \wedge q \wedge \neg p$	
{p}	Operators out
{q}	
{-p}	

Step 2: apply the resolution rule

1. {p}	Premise
2. {q}	Premise
3. {-p}	Premise
4. {}	1,3

Example:

- We want to prove that

$$((B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}) \models \neg P_{1,2}.$$

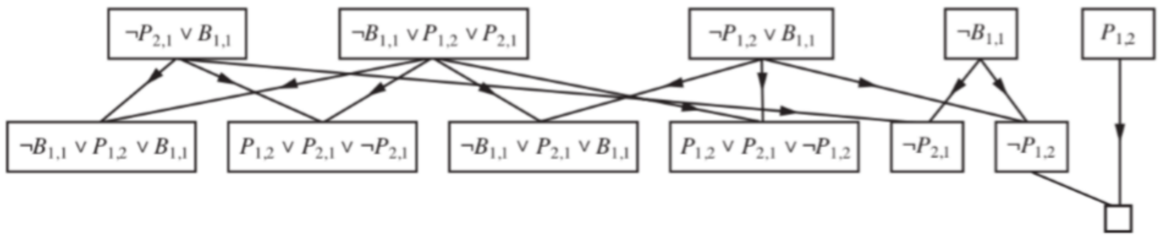
- So, we show that

$$((B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}) \wedge P_{1,2}$$

  is a contradiction (recall that $P_{1,2}$ is equivalent to $\neg\neg P_{1,2}$).

- We convert the above formula in CNF (see page 50):
  1. $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1}) \wedge \neg B_{1,1} \wedge P_{1,2}$
  2. $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1}) \wedge \neg B_{1,1} \wedge P_{1,2}$
  3. $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1}) \wedge \neg B_{1,1} \wedge P_{1,2}$
- So, we obtain the following clauses:

$$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \qquad (\neg P_{1,2} \vee B_{1,1}) \qquad (\neg P_{2,1} \vee B_{1,1}) \qquad \neg B_{1,1} \qquad P_{1,2}$$



- The clauses we obtained are in the first row above.
- The second row shows clauses obtained by resolving pairs in the first row.
- When $P_{1,2}$ is resolved with $\neg P_{1,2}$ we obtain the empty clause.
- This means that

$$((B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}) \wedge P_{1,2}$$

  is a contradiction.

- So, we have proved

$$((B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}) \models \neg P_{1,2}.$$

Deduction Theorem

Validity is connected to inference via the **Deduction Theorem**:

KB |= a if and only if KB ⇒ a

The Deduction Theorem tells us that **a** is a logical consequence of KB if and only if KB ⇒ a is a tautology.

KB |=a if and only if ¬(KB ∧ ¬a)

This means that **a** is a logical consequence of KB if and only if KB∧¬a is a contradiction.

Probabilistic

Boolean algebras of set

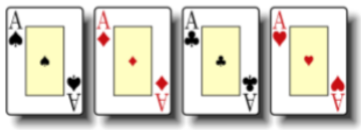
Let W be s non-empty set and let 2^W be the powerset of W, i.e. the set of all subsets of W. A **Boolean algebra of sets** $\mathcal{F}$ is a subset of 2^W that is

- closed under intersection, i.e. for all $A, B \in \mathcal{F}, \quad A \cap B \in \mathcal{F}$
- closed under union, i.e. for all $A, B \in \mathcal{F}, \quad A \cup B \in \mathcal{F}$
- closed under complementation, i.e. for all $A \in \mathcal{F}, \quad A^c \in \mathcal{F}$

Events:

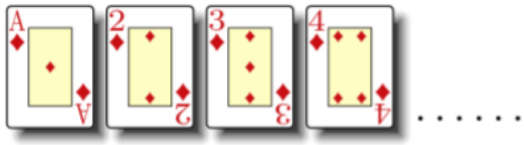
- Let W be a set of possible worlds and let A,B⊆W be events.
- A∪B is the event that occurs if at least one of A and B occurs.
- A∩B is the event that occurs if both A and B occur.
- A^c is the even that occurs if A does not occur.
- W is the certain event, i.e. the event that always occurs.

- Pick a card from a standard deck of 52 cards.
- The sample space W is the set of all 52 cards.
- Consider the following four events:
  - A: card is an ace.

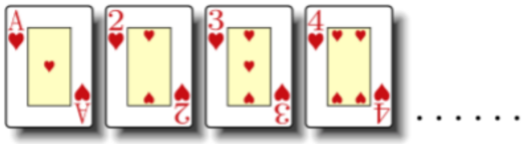


  - B: card has a black suit.

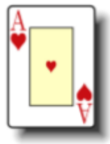
- D: card is a diamond.



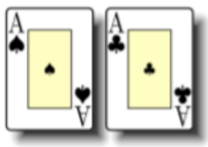
- H: card is a heart:



- $A \cap H$:



- $A \cap B$:



Finitely Additive Probability Measures

Let W be a finite set and let $\mathcal{F}$ be a Boolean algebra of sets. A **finitely additive probability measure** over $\mathcal{F}$ is a function

$$\mu : \mathcal{F} \to [0, 1]$$

such that, for all A, B $\in \mathcal{F}$

- $\mu(W) = 1$
- $\mu(A \cup B) = \mu(A) + \mu(B) - \mu(A \cap B)$
- $\mu(\emptyset) = 0$
- $\mu(A^c) = 1 - \mu(A)$
- if $A \subseteq B$, then $\mu(A) \leq \mu(B)$

Definition

Given a (finite) set W, a probability distribution over W is a function

$$\pi : W \to [0, 1]$$

such that

$$\sum_{x \in W} \pi(x) = 1.$$

- For every probability distribution over a finite set it is possible to define a corresponding probability measure.
- Given a finite set W along with a Boolean algebra $\mathcal{F}$ and a probability distribution π over W, for all $A \in \mathcal{F}$, the function

$$\mu(A) = \sum_{x \in A} \pi(x)$$

is a probability measure over $\mathcal{F}$.

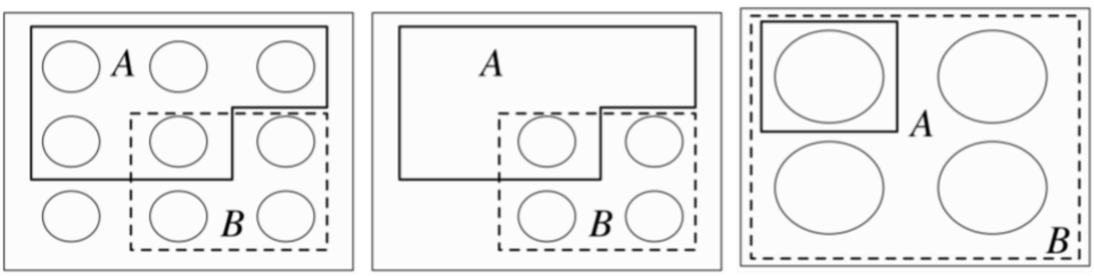
Conditional Probability

If A and B are events, with $\mu(B) > 0$, the conditional probability of A given B, denoted by $\mu(A|B)$ is given by

$$\mu(A|B) = \frac{\mu(A \cap B)}{\mu(B)}$$

- A is the event we want to update and B is the evidence.
- $\mu(A)$ is the **prior** probability of A.
- $\mu(A|B)$ is the **posterior** probability of A.

($\mu(A|B)$ 等于A和B同时发生的部分的占总体的概率除以B占总体的概率)



- Take a finite set of possible worlds and an event A.
- We learn that B occurred.
- We can obtain $\mu(A \mid B)$ by eliminating the worlds in B^c and renormalising the probability of A over B.

Suppose I have 2 cats.

What is the probability that both are female, given that you know that at least one is a girl? What is the probability that both are female, given that you know that the younger is a girl?

$$\mu(\textit{both girls} \mid \textit{at least one girl}) = \frac{\mu(\textit{both girls} \cap \textit{at least one girl})}{\mu(\textit{at least one girl})} = \frac{1/4}{3/4} = \frac{1}{3}$$

$$\mu(\textit{both girls} \mid \textit{younger is a girl}) = \frac{\mu(\textit{both girls} \cap \textit{younger is a girl})}{\mu(\textit{younger is a girl})} = \frac{1/4}{1/2} = \frac{1}{2}$$

Independent

Events A and B are **independent** if

$$\mu(A \cap B) = \mu(A) \cdot \mu(B).$$

If μ(A) > 0 and μ(B) > 0, then this is equivalent to

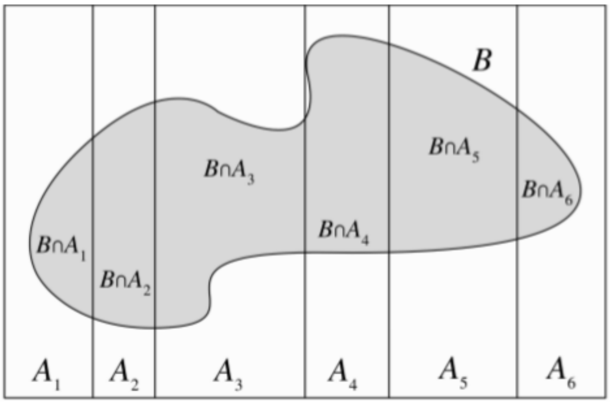
$$\mu(A|B) = \mu(A)$$
$$\mu(B|A) = \mu(B)$$

Independence is completely different from **disjointness**.

- If A and B are disjoint, then $\mu(A \cap B) = 0$, so they are independent only if $\mu(A) = \mu(B) = 0$.
- **Disjoin**: A occurs tells us that B definitely did not occur, so A clearly conveys information about B, meaning the two events are not independent

Bayes' Rule

- For any events A,B with positive probabilities,
  $\mu(A \cap B) = \mu(B) \cdot \mu(A|B) = \mu(A) \cdot \mu(B|A)$
- For any events A1,...,An with positive probabilities,
  $\mu(\bigcap_{i=1}^{n} A_i) = \mu(A_1) \cdot \mu(A_2|A_1) \cdot \mu(A_3|A_1 \cap A_2) \cdots \mu(A_n|\bigcap_{j=1}^{n-1} A_j)$
- Bayes' Rule
  $\mu(A|B) = \frac{\mu(B|A) \cdot \mu(A)}{\mu(B)}$
- Let $A_1,\ldots,A_n$ be a partition of the set of possible worlds, with $\mu(A_i) > 0$ for all i. Then
  $\mu(B) = \sum_{i=1}^{n} \mu(B|A_i) \cdot \mu(A_i)$



Since the A_i form a partition, then

$$B = (B \cap A_1) \cup (B \cup A_2) \cup \cdots \cup (B \cap A_n).$$

Then, since the pieces are disjoint

$$\mu(B) = \mu(B \cap A_1) + \mu(B \cup A_2) + \cdots + \mu(B \cap A_n).$$

Finally

$$\mu(B) = \mu(B \mid A_1) \cdot \mu(A_1) + \cdots + \mu(B \mid A_n) \cdot \mu(A_n).$$

MINI-Max