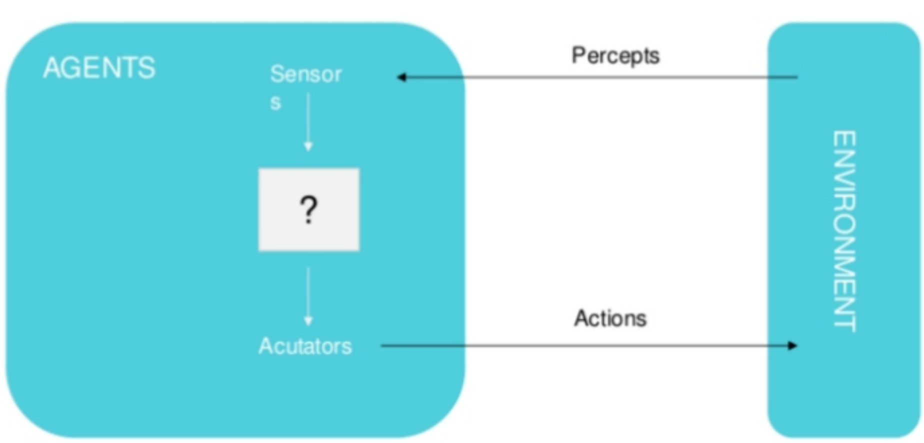
Week 2 - Intelligent Agents and Search Algorithms

- Agents = Anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators
  - ↳ Human Agents
  - ↳ Robotic Agents
  - ↳ Software Agents

- An intelligent agent is a system that :
  - Perceives its environment
  - Reasons to interpret perceptions, draw inferences, solve problems, and determine actions
  - Acts
  - Rational : Select an action that is expected to maximize its performance measure, based on evidence



- Before design an intelligent agents, harus specify task environment / PEAS :
  - Performance Measure : How desirable was an agent's action
  - Environment : What is the environment which agent going to move
  - Actuators : What are the agent's abilities to change environment
  - Sensors : How is the agent going to perceive environment

o) Contd :

Agent Type	Performance Measure	Environment	Actuators (Actions)	Sensors (Percepts)
Taxi driver	Safe, fast, legal, comfort, max. profits	Roads, traffic, pedestrians, customers	Steering, accelerators, break, signal, display	Cameras, sonar, speedometer, GPS, engine sensors, odometer, fee meter
Medical diagnosis system	Healthy patient, min. cost, lawsuits	Patient, hospital, staff	Display, questions, tests, diagnoses, treatments	Keyboard entry symptoms, patient answers, medical history
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display categorization of scene	Color pixel arrays
Part-picking robot	Percentage parts in correct bins	Conveyor belt with parts, bins	Jointed arm and hand	Camera, joint angle, sensors

o) Fully VS Partially Observable

- If an agent's sensors give it access to complete state of the environment at each point in time
- A fully observable AI environment has access to all required information to complete target task
- Partially observable environment is one in which the agent can never see the entire state of environment

o) Single VS Multi Agent

- One of the most obvious dimensions to classify and AI environment is based on the number of agents involved.
- When there is only one agent in a defined environment, it is named the Single-Agent System / SAS. Only act only with its environment.
- If there is more than 1 agent and interact with each other and their environment, the system called Multi-Agent System

Deterministic VS Stochastic

- If the next state of the environment is completely determined by current state and action executed by agent
- Dikasih current state of environment dan aksi agen AI, the AI bisa tahu next state of environment
- In a stochastic environment, there is always some level of randomness and uncertainty

Static VS Dynamic

- Static AI environments rely on data-knowledge source that don't change frequently over time.
- Dynamic AI environments such as vision AI systems in drones deal with data sources that change quite frequently

Sequential VS Episodic

- In episodic environment, agent's experience dibagi menjadi atomic episodes. Tiap episode, agent menerima persepsi dan perform single action. Next episode tidak depend on action dari previous episode.
- Sequential : Current decision could affect all future decisions, require memory of past actions to determine next best action.
- Nah, agent performs independent tasks in episodic, whereas non-episodic all agent's action are related

Discrete VS Continuous

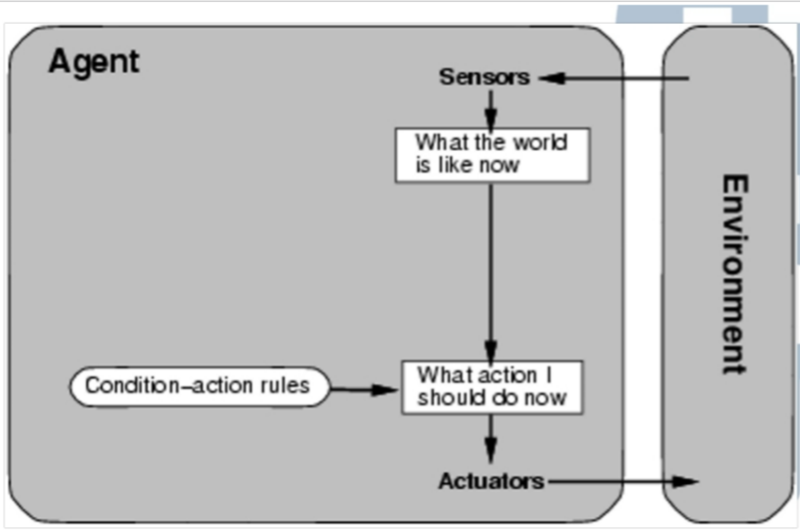
- Discrete : Finite set of posibilities can drive final outcome of the task
- Continuous : Rely on unknown and rapidly changing data sources

o)

Task Env.	Observable	Deterministic/Stochastic	Episodic/Sequential	Static/Dynamic	Discrete/Continuous	Agents
Crossword puzzle	fully	D	S	S	D	S
Taxi driving	partial	S	S	D	C	M
Medical diagnosis	partial	S	S	D	C	S
Image analysis	fully	D	E	S	C	S
Part-picking robot	partial	S	E	D	C	S

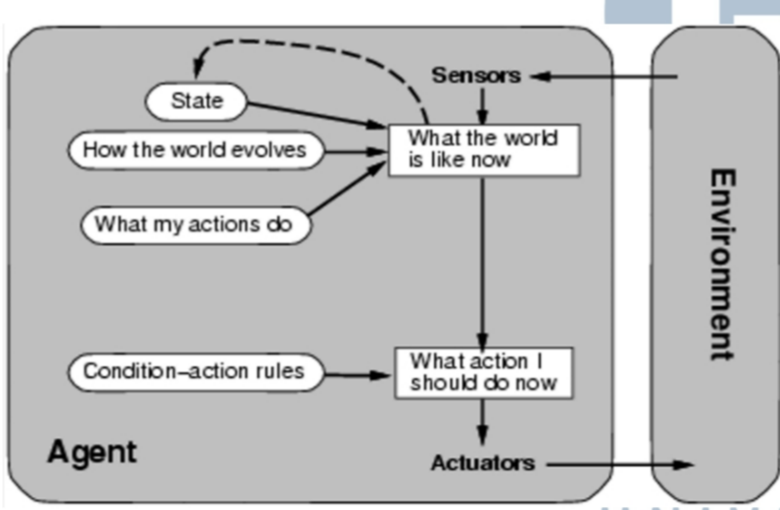
o) Agent ada 5 macam berdasarkan degree of perceived intelligence and capacity :

1. Simple Reflex Agents



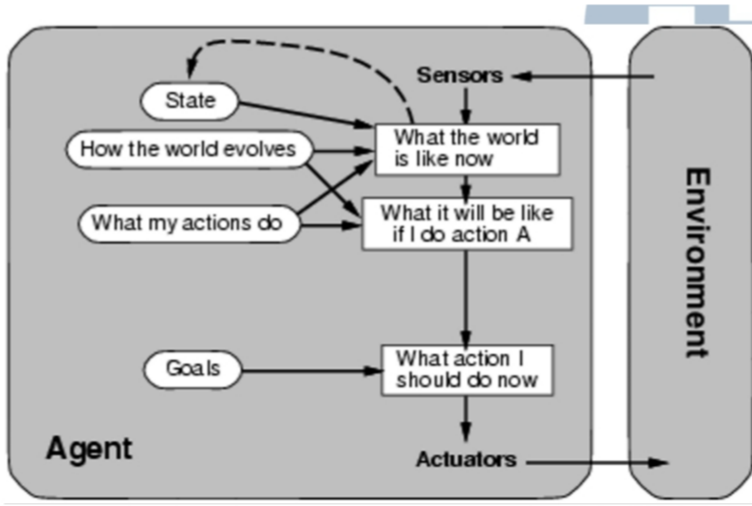
- Select action based on the agents current perception and not based on the past perceptions.
- Rules are triggered (if-then rules).
- Car in front is braking, then braking.

2. Model - Based Reflex Agents



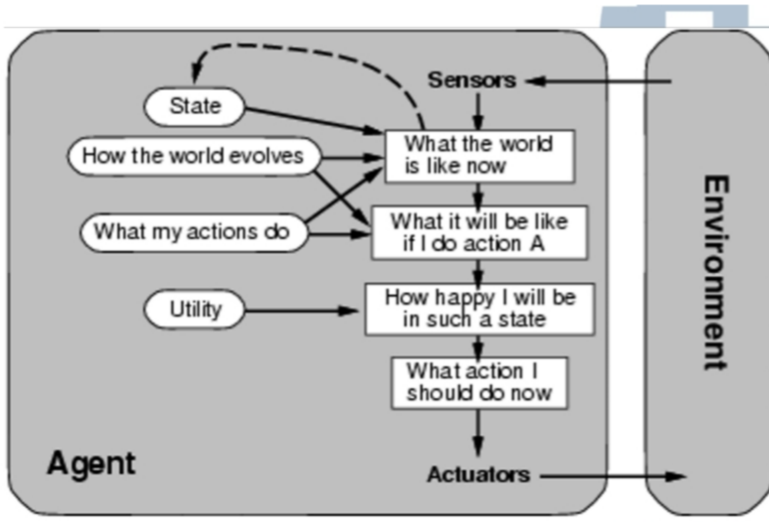
- Partially observable problems: the agent doesn't know what will happen if it will take some action against the current percept agent keeps track of the part of the world it can't see now (internal state updates).
- Based on the input of the sensors and knowledge of the world.
- It is able to store its sensor history/action that has been done.
- Rules are triggered.
- Smarts are shared. Still, the coder can have a big influence.

3. Goal-Based Reflex Agents



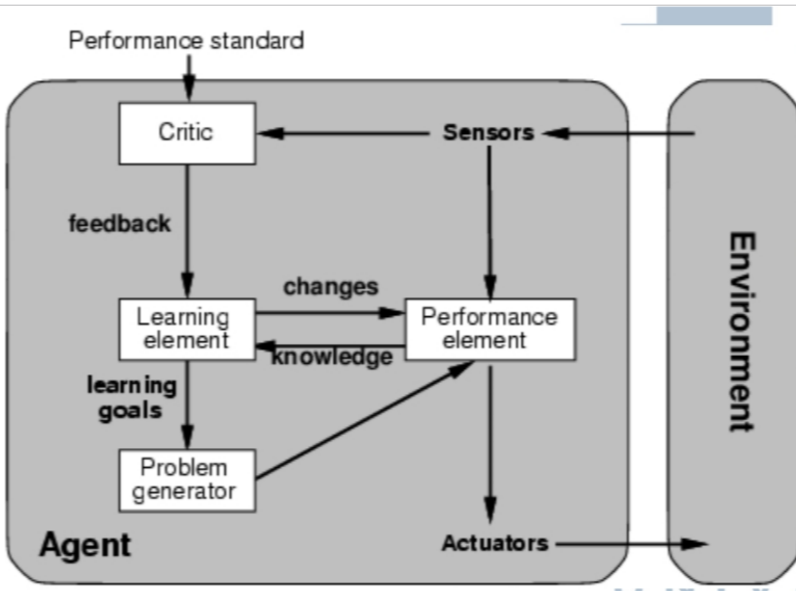
- Goals are evaluated by the system.
- Based on the input of the sensors and knowledge of the world.
- Goals are evaluated according to sensors and state.
- Rules are triggered.
- Smarts are shared.
- Choose one action that is closest to the goal.

4. Utility-Based Reflex Agents



- Goals are evaluated according to a utility function.
- Based on the input of the sensors and knowledge of the world.
- Rules are triggered.
- Smarts are shared. Less on the coder.
- Goals are evaluated according to sensors and state.

5. Learning Agents



- The goal is to improve performance.
- **Performance element**: choose the action.
- **Learning element**: improvement of basic knowledge (rewards/penalties).
- **Critic**: score exam/evaluation component.
- **Problem generator**: responsible for suggesting actions that will lead to new and informative experiences.

∘) Search Algorithms

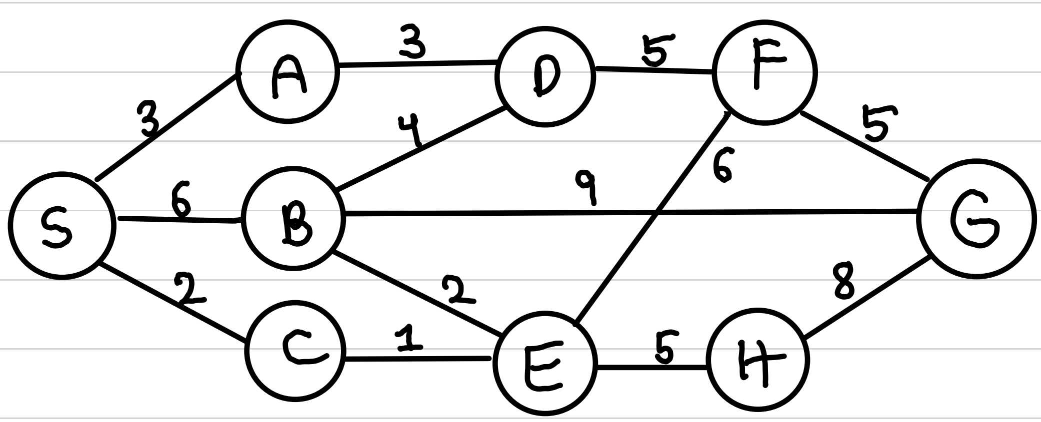
1. Uninformed Search Strategy

- BFS : Expand shallowest node | FIFO/Queue
- DFS : Expand deepest node | LIFO/Stack
- UCS : Expand least cost node | Bandingkan semua kemungkinannya

2. Informed Search Strategy

- Greedy Best-first Search
- A* Search

∘)



S ke G?

BFS : S, C, B, A, E, G

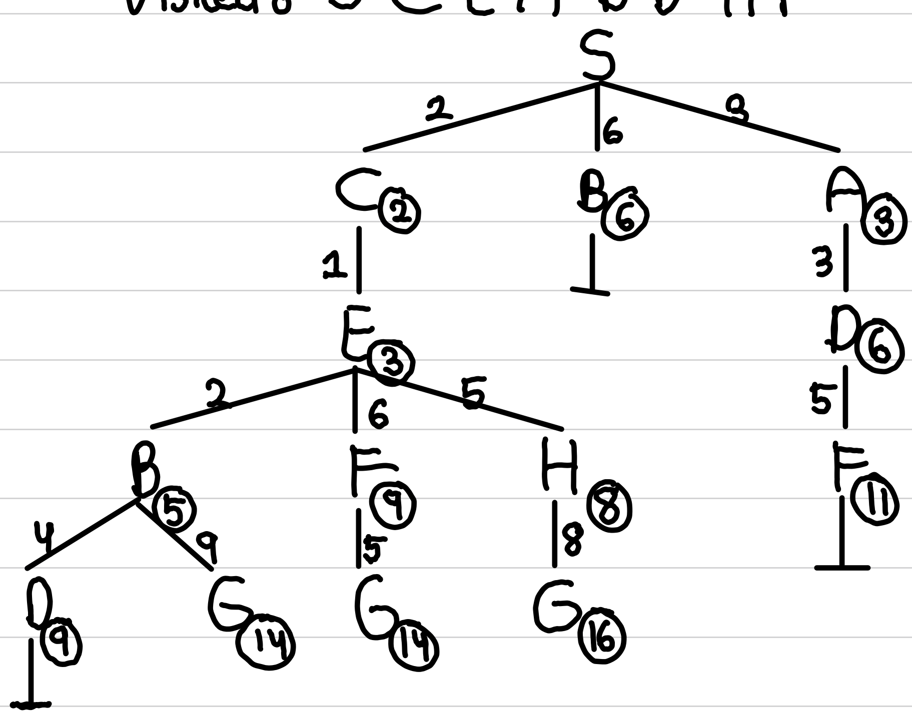
Q : ~~S~~ ~~C~~ ~~B~~ ~~A~~ ~~E~~ ~~G~~ D H F

DFS : S, A, D, F, G

S : ~~G~~ E ~~F~~ ~~D~~ ~~A~~ B C ~~S~~

UCS : S, C, E, B, G V S, C, E, F, G

Visited : S C E A B D H F



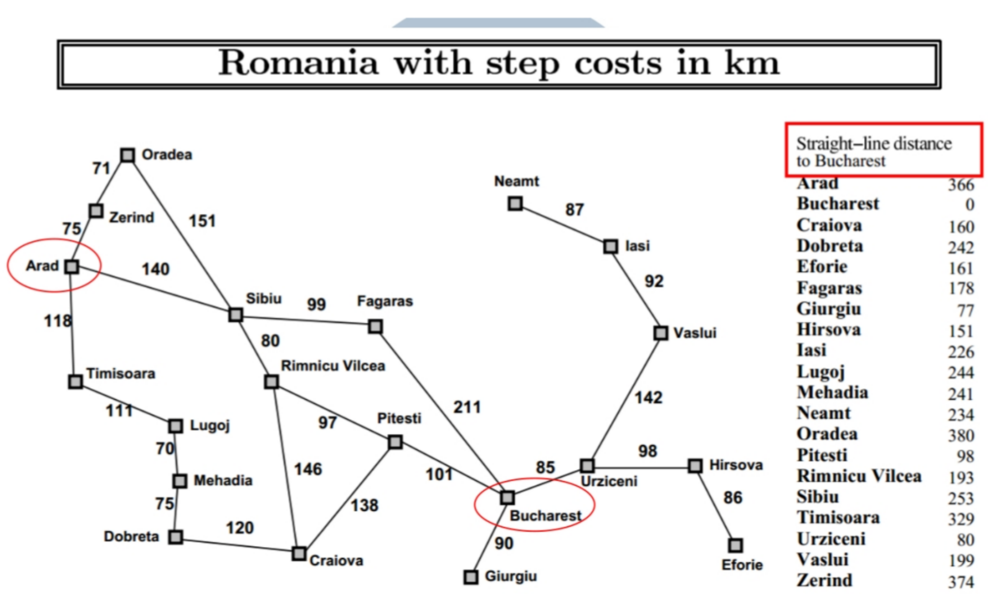
Best-first Search

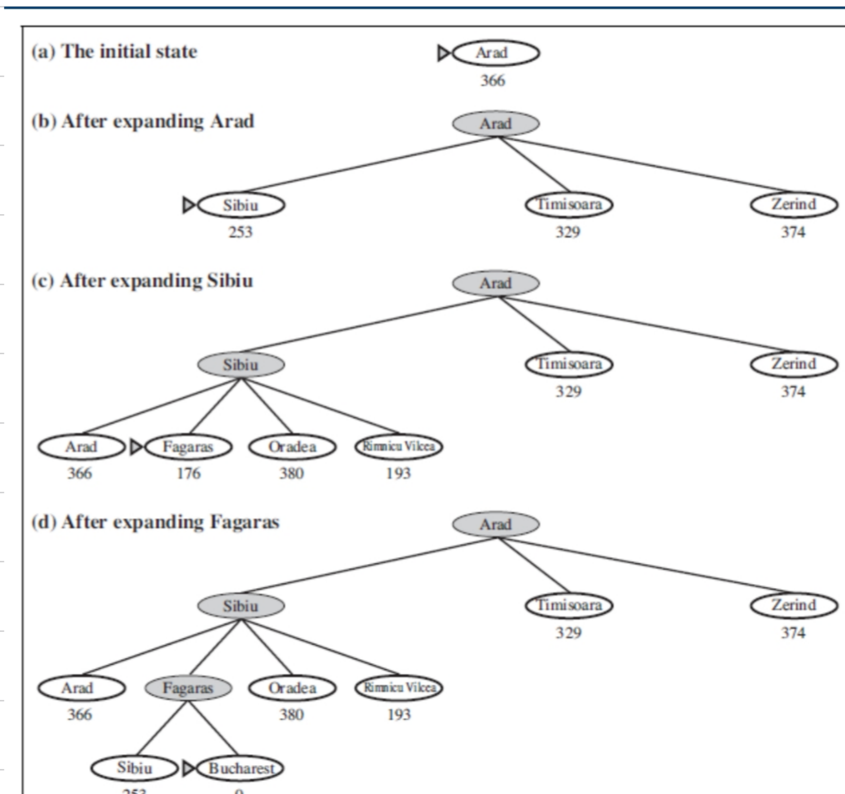
- A node is selected for expansion based on an evaluation function, $f(n)$, estimate of desirability expand most desirable unexpanded node.
- Implementation : Order the nodes in queue in decreasing order of desirability

Greedy Best-first Search I

- Heuristic evaluation function, $h(n)$ = Estimate of cost from n to goal. e.g. : $h_{SLD}(n)$ = Straight line distance from n to Bucharest
- Expand node that appears to be closest to goal

Greedy Best-first Search II





Smw alternatif diitung costnya (cari yg plg optimal)

↓

Max/Min

- Complete: No – e.g. Iasi to Fagaras (infinite loop).
  - Iasi -> Neamt -> Iasi -> Neamt
- Time: $O(b^m)$ – assume b is constant and m is the max. depth of the search space.
- Space: $O(b^m)$
- Optimality: No
  - Greedy cost: Arad – Sibiu – Fagaras – Bucharest = 450 km
  - Optimal cost: Arad – Sibiu – Rimnicu Vilcea – Pitesti – Bucharest = 418 km

o) A* Search I

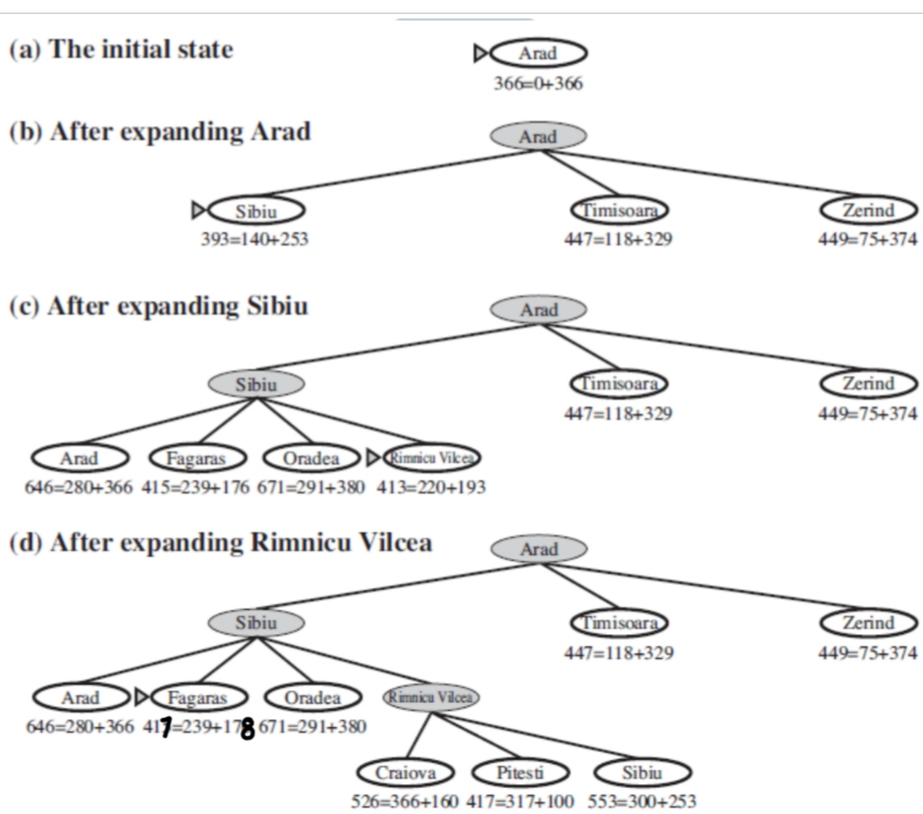
↳ Most widely known form of Best-first search

↳ Avoid expanding paths yg udh mahal

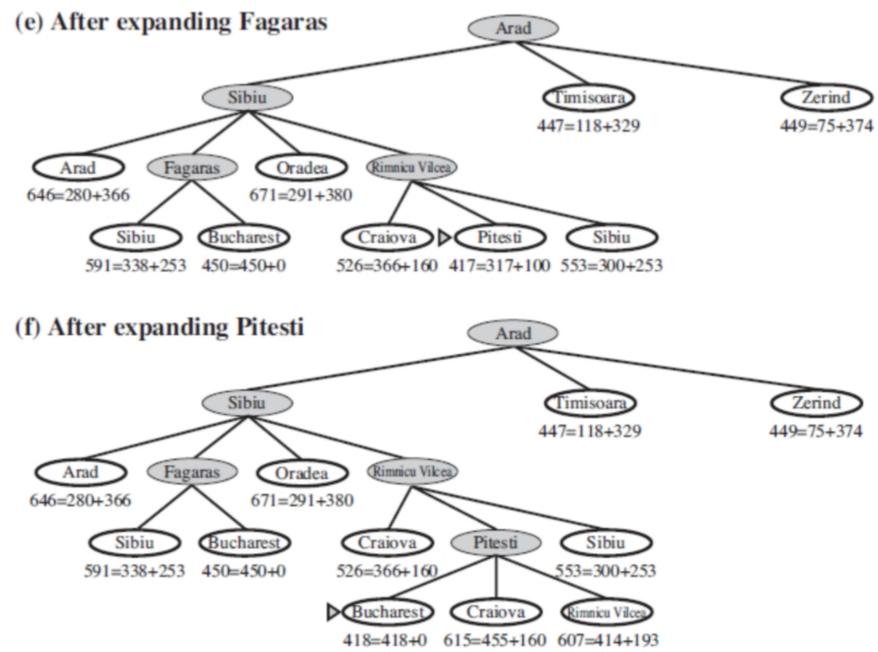
↳ $f(n) = g(n) + h(n)$

- $g(n)$ = 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

o) A* Search II



c) A* Search III



c) A* Search IV

- Complete: Yes
- Time: exponential – impractical for many large-scale problems.
- Space: exponential – impractical for many large-scale problems. Keeps all nodes in memory.
- Optimality: Yes

Week 3 – Models of Knowledge Representation and Propositional Logic

- Solving mslh dgn AI dpt represent & reason knowledge about world utk dpt meaningful answer

- Knowledge Representation (KR) is study of how knowledge & facts about world can be represented and what kinds of reasoning can be done with that knowledge.

-

Knowledge-based agents
Knowledge base
Inference engine

- **Knowledge base (KB)**: a set of sentences that describe the world and its behavior in some formal (representational) language.
- **Inference engine**: a set of procedures that use the representational language to infer new facts from known ones or answer a variety of KB queries. Inferences typically require search.

- Key aspects of knowledge representation language :
  - ↳ Syntax : Describes how sentences are formed
  - ↳ Semantics : Describes meaning of sentences yg refer to real world

- How knowledge be represented ?
  - ↳ Symbolic : Logic → Propositional logic, fuzzy logic, FOL, ontology, Prolog
  - ↳ Non-symbolic : Neural network, genetic algorithm

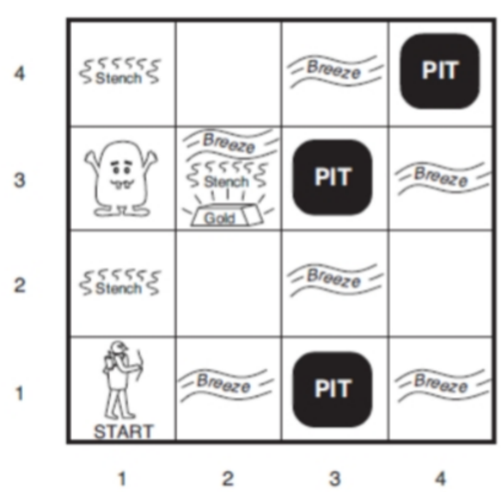
- Logic = Language for expressing reasoning steps
  - ↳ Syntax : Atomic symbols of logical language and its rules
  - ↳ Semantics : Define truth of sentence

- An expression in some language :
  - True / False
  - Negation makes sense
  - Believe or not
  - Negation believe or not

- Proporsi ada nilai kebenaran

- Valid / Tautology : Smw true
  Satisfability : Sebagian true
  Unsatisfability / Contradiction : Smw false

- Wumpus world :



- Adjacent rooms are connected (horizontally or vertically)
- Lurking in the cave is the Wumpus
- Agent can smell the Wumpus (**stench**)
- Agent feels a **breeze** if pit nearby
- Agent can shoot ONE arrow at (and kill) the Wumpus (**scream**)
- Some rooms contain pits that will trap agent
- One room contains a pot of gold (**glitter**)
- When agent walks into a wall, it will perceive a **bump**

- PEAS:
  - **Performance measure:** +1000 for walk out w/ gold; -1000 for dying; -1 for each action; -10 for arrow
  - **Environment:** a 4x4 grid. Agent starts at [1,1]; gold and pits randomly distributed.
  - **Actuators:** agent can move forward, backward, left, or right.
  - **Sensors:** [Stench, Breeze, Glitter, Bump, Scream]

- $P_{x,y}$ true kalau ada pit di $[x,y]$
  $W_{x,y}$ true kalau ada wumpus di $[x,y]$
  $B_{x,y}$ true kalau ada breeze di $[x,y]$
  $S_{x,y}$ true kalau ada stench di $[x,y]$

- Sentence : Suffice to derive $\neg P_{1,2}$ (Mau buktikan)

  - No pit in $[1,1]$
    R_1 : $\neg P_{1,1}$

- A square is breezy if and only if there is a pit di tetangga
  $R_2 : B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
  $R_3 : B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$

- After visit [1,1], [1,2], [2,1]. The breeze percepts for the first 2 square visited in the specific world
  $R_4 : \neg B_{1,1}$
  $R_5 : B_{2,1}$

- $KB = R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5$

- Apply biconditional elimination to R_2
  $R_6 : (B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

- And-elimination to R_6
  $R_7 : ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

- Logical equivalence for contra-positives gives
  $R_8 : (\neg B_{1,1} \Rightarrow \neg (P_{1,2} \vee P_{2,1}))$

- Modus ponens with R_8 and precept R_4
  $R_9 : \neg (P_{1,2} \vee P_{2,1})$

- De Morgan
  $R_{10} : \neg P_{1,2} \wedge \neg P_{2,1}$ → Neither [1,2] nor [2,1] contains pit

c)

$(\alpha \wedge \beta)$	$\equiv$	$(\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta)$	$\equiv$	$(\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma)$	$\equiv$	$(\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma)$	$\equiv$	$(\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg\alpha)$	$\equiv$	α	double-negation elimination
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta)$	$\equiv$	$((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta)$	$\equiv$	$(\neg\alpha \vee \neg\beta)$	De Morgan
$\neg(\alpha \vee \beta)$	$\equiv$	$(\neg\alpha \wedge \neg\beta)$	De Morgan
$(\alpha \wedge (\beta \vee \gamma))$	$\equiv$	$((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma))$	$\equiv$	$((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

- Inference and Proofs: Applied to derive a proof lead to goal
  - Modus ponens: $\dfrac{\alpha \Rightarrow \beta \quad \alpha}{\beta}$
  - AND-elimination: $\dfrac{\alpha \land \beta}{\beta}$
  - Logical equivalence: $\dfrac{\alpha \Leftrightarrow \beta}{(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)}$ and $\dfrac{(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)}{\alpha \Leftrightarrow \beta}$
  - Proof by resolution: $\dfrac{A \lor B, \ \neg A \lor C}{B \lor C}$

Horn Clauses

- A disjunction of literals of which at most one is positive
  - ↳ e.g.: $(\neg P_{1,2} \lor \neg P_{2,1} \lor B_{1,1})$, $(\neg P_{1,2} \lor \neg P_{2,1} \lor \neg B_{1,1})$

Definite Clauses

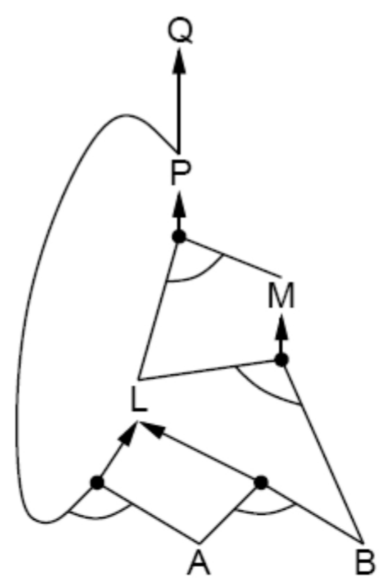
- Restricted inference algorithm
- A disjunction of literals of which exactly one is positive
  - ↳ e.g.: $(\neg P_{1,2} \lor \neg P_{2,1} \lor B_{1,1})$
- Every definite clause can be re-written as implikasi
  - ↳ $(\neg P_{1,2} \lor \neg P_{2,1} \lor B_{1,1}) \equiv (P_{1,2} \land P_{2,1}) \Rightarrow B_{1,1}$

- Inference with horn clause can be done through forward-chaining and backward chaining.

- **Forward Chaining**
  - For each new piece of data, generate all new facts, until the desired fact is generated.
  - Data-directed reasoning - reasoning in which the focus of attention starts with the known data.
  - E.g. decision support system, medical diagnosis system.
- **Backward Chaining**
  - To prove the goal, find a clause that contains the goal as its head, and prove the body recursively.
  - Backtrack when you chose the wrong clause
  - Goal-directed reasoning.
  - E.g. Computer forensic investigation.

- Fire any rule whose premises are satisfied in the KB.
- Add its conclusion to the KB until the query is found.
- Ex.: Given A and B are true, prove that Q is true.

$$
\begin{aligned}
&P \Rightarrow Q \\
&L \wedge M \Rightarrow P \\
&B \wedge L \Rightarrow M \\
&A \wedge P \Rightarrow L \\
&A \wedge B \Rightarrow L \\
&A \\
&B
\end{aligned}
$$



Week 4 - First Order Logic

- FOL = Logical system for reasoning about properties of objects
- PL = Extension of PL/Propositional Logic
- predicates = Describe properties of objects
  functions = Map objects to one another
  quantifiers = Allow us to reason about multiple objects
- FOL VS PL
  - PL no signify/express generalization, specialization, or pattern seperti quantifier di FOL
  - PL convert a complete sentence to symbol and makes it logical, sedangkan FOL ad relasi, fungsi, konstanta
- FOL ada :
  - Constants → IF 307, John, apple
  - Relasi / Predikat → is_Man(John), like(tio, children)
  - Function → nigga_why( ), friend_of( )
- To reason about objects, FOL pke predikat yg menghslkn proposisi, either T/F.
- Equality = Part of FOL (Kontol = Penis)
  ↳ Equality only applied to objects. To state proposisi equal, pakai ↔
- Function always return a single value and evaluate to objects not propositions
-

	... operate on ...	... and produce
Connectives (↔, ∧, etc.) ...	propositions	a proposition
Predicates (=, etc.) ...	objects	a proposition
Functions ...	objects	an object

a)

$$
\begin{array}{rcl}
\textit{Sentence} & \rightarrow & \textit{AtomicSentence} \mid \textit{ComplexSentence} \\
\textit{AtomicSentence} & \rightarrow & \textit{Predicate} \mid \textit{Predicate}(\textit{Term}, \ldots) \mid \textit{Term} = \textit{Term} \\
\textit{ComplexSentence} & \rightarrow & (\textit{Sentence}) \mid [\textit{Sentence}] \\
& \mid & \neg\, \textit{Sentence} \\
& \mid & \textit{Sentence} \wedge \textit{Sentence} \\
& \mid & \textit{Sentence} \vee \textit{Sentence} \\
& \mid & \textit{Sentence} \Rightarrow \textit{Sentence} \\
& \mid & \textit{Sentence} \Leftrightarrow \textit{Sentence} \\
& \mid & \textit{Quantifier Variable}, \ldots\ \textit{Sentence} \\
\\
\textit{Term} & \rightarrow & \textit{Function}(\textit{Term}, \ldots) \\
& \mid & \textit{Constant} \\
& \mid & \textit{Variable} \\
\\
\textit{Quantifier} & \rightarrow & \forall \mid \exists \\
\textit{Constant} & \rightarrow & A \mid X_1 \mid \textit{John} \mid \cdots \\
\textit{Variable} & \rightarrow & a \mid x \mid s \mid \cdots \\
\textit{Predicate} & \rightarrow & \textit{True} \mid \textit{False} \mid \textit{After} \mid \textit{Loves} \mid \textit{Raining} \mid \cdots \\
\textit{Function} & \rightarrow & \textit{Mother} \mid \textit{LeftLeg} \mid \cdots \\
\\
\text{OPERATOR PRECEDENCE} & : & \neg, =, \wedge, \vee, \Rightarrow, \Leftrightarrow
\end{array}
$$

a) Syntax FOL :

Constant Symbols: *KingJohn*, 2, ...

Predicate symbols: *Brother*, >, ...

Function Symbols: *Sqrt, LeftLegOf, ...*

Variable symbols: *x, y, a, b, ...*

Connectives: $\wedge$, $\vee$, $\neg$, $\Rightarrow$

Equality: =

Quantifiers: $\forall$, $\exists$

Punctuation: ( ) ,

Term: *function(term$_1$, ..., term$_n$)* | *constant* | *variable*

LeftLeg(John)

Atomic sentences: *predicate(term$_1$, ..., term$_n$)* | *term$_1$ = term$_2$*

Brother(Richard, John); Married(Father(Richard), Mother(John))

Complex sentences: *Brother(John, Richard)* $\wedge$ *Brother (Richard, John)* or

$\neg$*King(Richard)* $\Rightarrow$ *King(John)*

Universal Quantifiers: $\forall x\ King(x) \Rightarrow Person(x)$ → Utk form rules ($\Rightarrow$)

- The expression is true for every possible value of the variable.
- "For all...", "All···"

Existential Quantifiers: $\exists x\ Crown(x) \wedge OnHead(x, John)$ → To specify list of propertes about an individual ($\wedge$)

- The expression is true for at least one value of the variable.
- "There exists···", "There is at least one···", "For some···"

Equality: $Father(John) = Henry; \exists x,y\ Brother(x, Richard) \wedge Brother(y, Richard) \wedge \neg(x = y)$

- To state facts about a given function.
- To signify that two terms refer to the same object / not.

o)

x = {,,}

All purple mushrooms are poisonous

- $\forall x\ Purple(x) \wedge Mushroom(x) \Rightarrow Poisonous(x)$

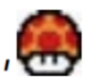
Universal: all three implications are true

-  is purple and a mushroom ⇒ is poisonous
-  is purple and a mushroom ⇒ is poisonous
-  is purple and a mushroom ⇒ is poisonous

Sesuai

∀

o)

x = {,,}

All purple mushrooms are poisonous

- $\forall x\ Purple(x) \wedge Mushroom(x) \wedge Poisonous(x)$

Universal: all three statements are true

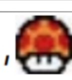
-  is purple and a mushroom and is poisonous
-  is purple and a mushroom and is poisonous
-  is purple and a mushroom and is poisonous

"Everything is a poisonous purple mushroom"

Salah

∀

o)

x = {,,}

Some purple mushrooms are poisonous

- $\exists x\ Purple(x) \wedge Mushroom(x) \Rightarrow Poisonous(x)$

Existential: at least one implication is true

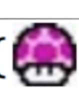
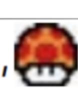
-  is purple and a mushroom ⇒ is poisonous
-  is purple and a mushroom ⇒ is poisonous
-  is purple and a mushroom ⇒ is poisonous

But all those where the premise is false are true

Salah

∃

o)

x = {,,}

Some purple mushrooms are poisonous

- $\exists x\ Purple(x) \wedge Mushroom(x) \wedge Poisonous(x)$

Existential: at least one statement is true

-  is purple and a mushroom and is poisonous
-  is purple and a mushroom and is poisonous
-  is purple and a mushroom and is poisonous

The first statement is true, the other two are false

- The existential quantifier is satisfied
- The situation is consistent with our original meaning

∃

o) $\forall$ use $\Rightarrow$ only
$\exists$ use $\wedge$ only

o) Everyone likes some kind of food
$\forall y\ \exists x,\ Food(x) \wedge Likes(y, x)$
sesuai urutan dikalimat

There is a kind of food that everyone likes
$\exists x\ \forall y,\ Food(x) \wedge Likes(y, x)$

Someone likes all kinds of food
$\exists y\ \forall x,\ Food(x) \wedge Likes(y, x)$

Every food has someone who likes it
$\forall x\ \exists y,\ Food(x) \wedge Likes(y, x)$

o) $\forall x \forall y \equiv \forall y \forall x$
$\exists x \exists y \equiv \exists y \exists x$
$\exists x \forall y \not\equiv \forall y \exists x$

$\exists x\ \forall y\ Loves(x, y)$
"There is a person who loves everyone in the world."
$\forall y\ \exists x\ Loves(x, y)$
"Everyone in the world is loved by at least one person."

o) $\forall x\ \neg P \equiv \neg \exists x\ P$
$\neg \forall x\ P \equiv \exists x\ \neg P$
$\forall x\ P \equiv \neg \exists x\ \neg P$
$\exists x\ P \equiv \neg \forall x\ \neg P$

Not everyone like McDonalds
$\neg(\forall x,\ Likes(x, McDonalds))$
$\exists x\ \neg Likes(x, McDonalds)$

No one likes McDonalds
$\neg(\exists x,\ Likes(x, McDonalds))$
$\forall x\ \neg Likes(x, McDonalds)$

o) Brothers are siblings < kandung / tiri
$\forall x,y\ \ Brother(x,y) \Rightarrow Sibling(x,y)$

Kalau Brothers are kandung pakai $\Leftrightarrow$

No two adjacent countries have the same map color
$\forall x,y\ \ Country(x) \wedge Country(y) \wedge Border(x,y) \Rightarrow \neg(Color(x) = Color(y)) \wedge \neg(x = y)$

One's mother is one's female parent
$\forall x,y\ \ Mother(x,y) \Leftrightarrow (Female(x) \wedge Parent(x,y))$

A grandparent is a parent of one's parent
$\forall g,c\ Grandparent(g,c) \Leftrightarrow \exists p\ Parent(g,p) \wedge Parent(p,c)$
Keduanya harus terpenuhi

◦ Universal Instantiations / Elimination

KB:
- ∀x King(x) ∧ Greedy(x) ⇒ Evil(x)
- King(John)
- Greedy(John)
- Brother(Richard, John)

Intinya dipisah

UI using {x/John} and {x/Richard}
- King(John) ∧ Greedy(John) ⇒ Evil(John)
- King(Richard) ∧ Greedy(Richard) ⇒ Evil(Richard)

discard the Universally quantified sentence. We can get the KB to be propositions.

◦ Build an inference engine :

1. And-introduction / AI

P,Q

P∧Q

2. Universal Elimination

3. Generalized Modus Ponens

Bob is a buffalo	1. *Buffalo*(*Bob*)
Pat is a pig	2. *Pig*(*Pat*)
Buffaloes outrun pigs	3. ∀*x*, *y* *Buffalo*(*x*) ∧ *Pig*(*y*) ⇒ *Faster*(*x*, *y*)
Bob outruns Pat	
AI 1 & 2	4. *Buffalo*(*Bob*) ∧ *Pig*(*Pat*)
UE 3, {*x*/*Bob*, *y*/*Pat*}	5. *Buffalo*(*Bob*) ∧ *Pig*(*Pat*) ⇒ *Faster*(*Bob*, *Pat*)
MP 4 & 5	6. *Faster*(*Bob*, *Pat*)

$$\frac{p_1', \; p_2', \; \ldots, \; p_n', \; (p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)}{q\sigma} \qquad \text{where } p_i'\sigma = p_i\sigma \text{ for all } i$$

$$\begin{aligned} \text{E.g. } p_1' &= \text{Faster(Bob,Pat)} \\ p_2' &= \text{Faster(Pat,Steve)} \\ p_1 \wedge p_2 \;\Rightarrow\; q &= Faster(x,y) \wedge Faster(y,z) \;\Rightarrow\; Faster(x,z) \\ \sigma &= \{x/Bob, y/Pat, z/Steve\} \\ q\sigma &= Faster(Bob, Steve) \end{aligned}$$

- 2 mekanisme utk apply GMP adalah forward / backward chaining

Week 5 - Bayesian Theorem

- Bayes Rule menentukan gmn gabungin data & pengetahuan sebelumnya / Prior

- Probabilitas adalah ukuran kepercayaan agen dalam proposisi - subjective probability / Degree of uncertainty dari agen cerdas

- Keyakinan agen bergantung pada prior asumsi dan apa yang diamati oleh agen dari semua informasi yang ada. Hal ini memberikan probabilitas Prior.

- Probability = 0 → True
  Probability = 1 → False

- Random variables adalah elemen dasar probabilitas (Dimulai huruf kapital)

- $P(A = \text{True}) = P(A) =$ Unconditional Probabilitas

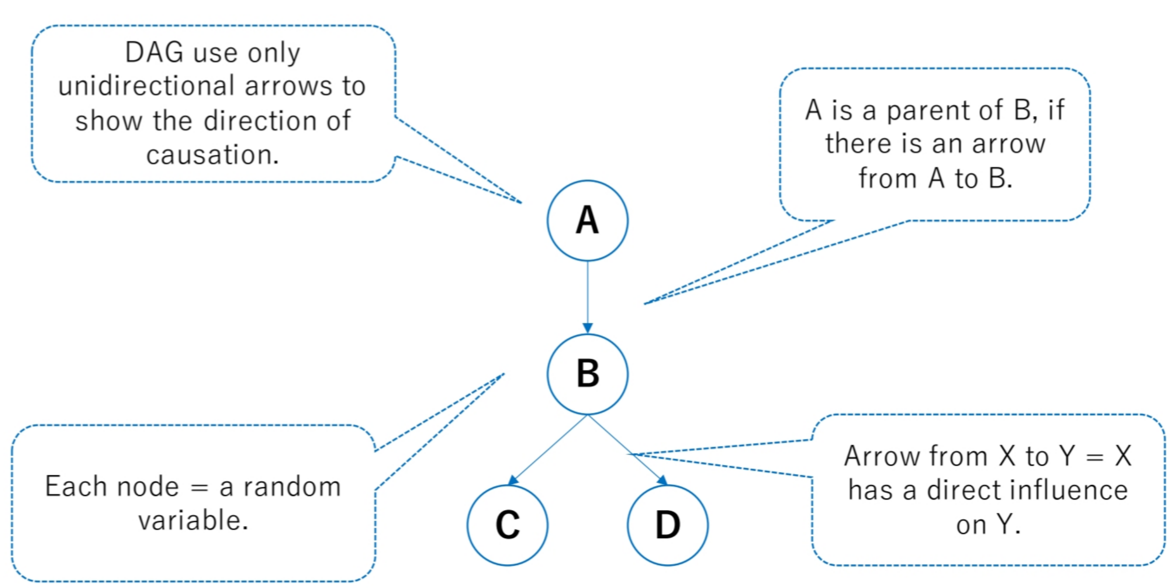
- $\underbrace{P(A}_{\text{Terjadi}}|\underbrace{B)}_{\text{Penyebab}} = \dfrac{P(A \cap B)}{P(B)} \equiv P(A \cap B) = P(A|B) \cdot P(B)$
  $= P(B|A) \cdot P(A)$

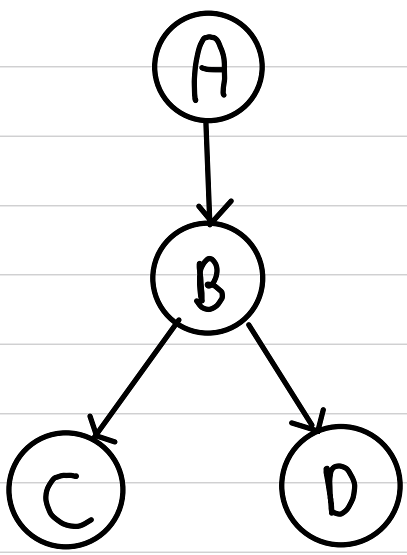
- $P(A|B) = \dfrac{P(B|A) \cdot P(A)}{P(B)}$

Week 6 – Bayesian Network

- Probabilistic reasoning is a method of representation of knowledge where the concept of probability is applied to indicate the uncertainty in knowledge.

- Bayesian Network :
  - Useful for representing and using probabilistic function
  - Harus tahu joint probability function
  - Model :
    - Directed graph over the variables
    - Associated probability distribution
  - Terdiri atas :
    - Directed Acyclic Graph (DAG)
    - Set of Conditional Probability Tables (CPT) for each node
  - Syntax :
    - Node = RV (Discrete / Continuous)
    - Arrows = Probabilistic dependencies antar node
    - Each node X_i has a conditional probability distribution $P(X_i \mid Parents(X_i))$ that quantifies effect of parent on the node

- 



- $P(A=true, B=true, C=true, D=true)$
  $= P(A=true) * P(B=true \mid A=true) * P(C=true \mid B=true) * P(D=true \mid B=true)$
- $P(B=true) = P(A=true) * P(B=true \mid A=true) + P(A=false) * P(B=true \mid A=false)$

Inference by Enumeration

$$P(X \mid e) = \alpha P(X, e) = \alpha \sum_y P(X, e, y)$$

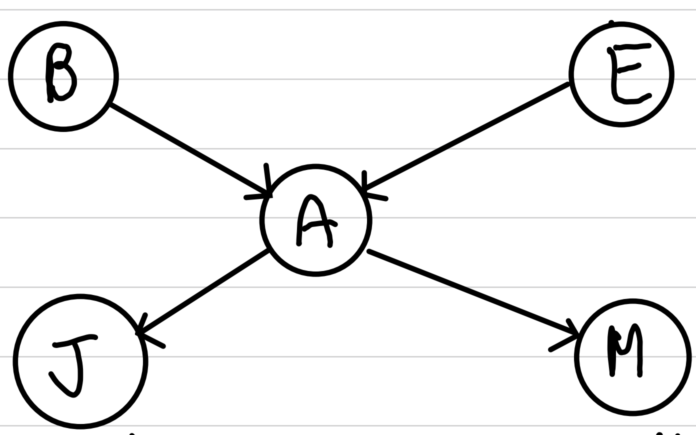
X = The query variables

e = A particular observed event

e = The evidence variables

y = Non-evidence, non-query variables (Hidden variables)

α = The normalization constant need to make the entries in $P(X \mid y)$ sum to 1



- $P(B \mid j, m) = \alpha P(B, j, m)$
  $= \alpha \sum_e \sum_a P(B, e, a, j, m)$
  $= \alpha \sum_e \sum_a P(b)\, P(e)\, P(a \mid b, e)\, P(j \mid a)\, P(m \mid a)$
  $= \alpha P(b) \sum_e P(e) \sum_a P(a \mid b, e) P(j \mid a) P(m \mid a)$

- If $b=t, a=t, e=t \rightarrow P(b)\, P(e)\, P(a \mid b, e)\, P(j \mid a)\, P(m \mid a)$
- If $b=t, a=t, e=f \rightarrow P(b)\, P(\neg e)\, P(a \mid b, \neg e)\, P(j \mid a)\, P(m \mid a)$
- If $b=t, a=f, e=t \rightarrow P(b)\, P(e)\, P(\neg a \mid b, e)\, P(j \mid \neg a)\, P(m \mid \neg a)$
- If $b=t, a=f, e=f \rightarrow P(b)\, P(\neg e)\, P(\neg a \mid b, \neg e)\, P(j \mid \neg a)\, P(m \mid \neg a)$

- $P(b \mid j, m) = \alpha \cdot 0.00059224$
- $\alpha = 1/(P(X \mid e) + P(\neg X \mid e))$
- $P(\neg b \mid j, m) = \alpha \cdot 0.0014919$

Inference by Variable Elimination

- Enumeration is inefficient : Repeated computation
- Carry out summations R to L, storing intermediate results to avoid re-computation
- $P(B|j,m) = \alpha P(B) \sum_e P(e) \sum_a P(a|B,e)\ P(j|a)\ P(m|a)$
  $= \alpha f_1(B) \sum_e f_2(E) \sum_a f_3(A,B,E)\ f_4(A)\ f_5(A)$

  ↳ $f_4(A)$ dan $f_5(A)$ corresponding to $P(j|a)$ dan $P(m|a)$ depends on A bcs J & M are fixed by query

- $f_4(A) = \begin{pmatrix} P(j|a) \\ P(j|\neg a) \end{pmatrix} = \begin{pmatrix} 0.9 \\ 0.05 \end{pmatrix}$
- $f_5(A) = \begin{pmatrix} P(m|a) \\ P(m|\neg a) \end{pmatrix} = \begin{pmatrix} 0.7 \\ 0.01 \end{pmatrix}$
- $f_6(B,E) = \sum_a f_3(A,B,E) \cdot f_4(A) \cdot f_5(A)$
  $= f_3(a,B,E) \cdot f_4(a) \cdot f_5(a) + f_3(\neg a,B,E) \cdot f_4(\neg a) \cdot f_5(\neg a)$

  ↳ $P(B|j,m) = \alpha f_1(B) \sum_e f_2(E) \cdot f_6(B,E)$

- $f_7(B) = \sum_e f_2(E) \cdot f_6(B,E) = f_2(e) \cdot f_6(B,e) + f_2(\neg e) \cdot f_6(B,\neg e)$

  ↳ $P(B|j,m) = \alpha f_1(B) \cdot f_7(B)$

- $P(A=a, B=\neg b) \rightarrow$ Hitung $P(A)$ saja
- Pakai Bayes biasa kalau dlm kondisi sgt spesifik, pakai bayes network kalau dlm kondisi in general (Semua kemungkinan bisa masuk brti ada T & F)

Week 7 - Markov Chain

- Markov chain is a mathematical system that experience transition from one state to another according to certain probabilistic rules.

- Used to compute probabilities of events occuring by viewing them as states transitioning into other states, or transitioning into same state kek sblmny.

- One special type of discrete time = Markov chain

- $P(X_{t+1} = i_{t+1} \mid X_t = i_t, \dots, X_1 = i_1, X_0 = i_0) = P(X_{t+1} = i_{t+1} \mid X_t = i_t)$

- $P(X_{t+1} = j \mid X_t = i) = P_{ij}$

- P_{ij} = Transition probabilities matrix

- $P(X_0 = i) = q_i \rightarrow$ When $i = 0$

- $$P = \begin{bmatrix} P_{11} & \dots & P_{1S} \\ P_{21} & \dots & P_{2S} \\ P_{S1} & \dots & P_{SS} \end{bmatrix} \quad \text{Tiap } i \rightarrow \sum_{j=1}^{S} P_{ij} = 1$$

  ↳ i&j objectnya
  ↳ i = Saat ini
  ↳ j = Kedepannya

- $P(X_{t+1} = j \mid X_t = i) = P_{ij}$: $n = 1$
  ↳ 1 langkah saja

- $P(X_{t+n} = j \mid X_m = i) = P(X_n = j \mid X_0 = i) = P_{ij}(n)$: $n > 1$
  ↳ Beberapa langkah

  n-step transition probabilities

- Always t, t+1; t-1 gbs diliat (Always next, can't see previous)