Categories and Objects

- Organizing objects into Categories is fundamental to Knowledge Representation.

- Interactions take place at object level but the reasoning takes place at the category level.
    - A shopper goes to a shop to buy a basketball
      (as supposed to buying the 3rd basketball on the 4th shelf)

- Agent infers the object using percepts then uses the category information to make prediction about the object
    - If Agent detects yellow peel, yellow soft pulp and a hard shell inside then Agent infers that the object is Mango
    - Using its category information, it infers that it can used to make milkshake

Categories and Objects

- There are two ways to represent categories and objects in First Order Logic:
  - Make Categories as unary predicates
    - BasketBall(B1)
  - Make the category also as an object (Reification)
    - Member(B1, BasketBalls) also written as B1 ∈ BasketBalls
    - SubClass(BasketBalls, Balls) also written as BasketBalls ⊆ Balls
- Categories are organized through inheritances
  - Subclasses organize categories into taxonomic hierarchy
    - Examples:
      - Library Dewey Decimal system
      - Biological Taxonomy : Kingdom, Phylum, Class, Order, Family, Genus, Species

Categories in First order logic

- Many natural properties of Categories can be stated in First Order Logic

    - An object is a member of a category
        - $B1 \in BasketBalls$

    - A category is a subclass of another category
        - $BasketBalls \subseteq Balls$

    - Members of a class can be recognized by some properties
        - $Orange(x) \land Spherical(x) \land Diameter = 9.5" \land x \in Balls \Rightarrow x \in BasketBalls$

    - All members of a category have some common property
        - $Dogs \in DomesticatedSpecies$
        - Here DomesticatedSpecies is a category of categories

Categories in First order logic

- Sometimes we also want to say two categories are disjoint / exhaustive / partition:
    - Disjoint({Animals, Vegatables, Basketballs})
        - Disjoint(s) : ∀ c1, c2 c1 ∈ s ∧ c2 ∈ s ∧ c1 ≠ c2 ⇒ Intersection(c1, c2) = { }
    - ExhaustiveDecomposition({CSE, AE, EE,...} , BTechStudents)
        - ExhaustiveDecomposition (s,c) : ∀ i i ∈ c ⇔ ∃ c2 c2 ∈ s ∧ i ∈ c2
    - Partition ({BTech, DualDegree, MTech, MS, PhD} , Students)
        - Partition (s,c) : Disjoint(s) ∧ ExhaustiveDecomposition(s,c)
- Some categories can also have necessary and sufficient conditions for membership
    - x ∈ Bachelors ⇔ Unmarried(x) ∧ x ∈ Adults ∧ x ∈ Males

Representing Physical Composition

- Nose is part of Face
- Chennai is part of Tamil Nadu
- Tamil Nadu is part of India
- India is part of Asia

- Objects can be grouped into Partof hierarchies
  - Partof(x,y) is reflexive and transitive

- Category of composite objects can be characterized using PartOf
  - Archipelago is a collection of at least 2 islands
  - $x \in \text{Archipelago} \Leftrightarrow \exists\ l1\ l2\ \ l1 \neq l2 \wedge \text{PartOf}(l1, x) \wedge \text{PartOf}(l2, x) \wedge l1 \in \text{island} \wedge l2 \in \text{island}$

    $\wedge\ \forall\ l\ \ \text{PartOf}(l, x) \Rightarrow l \in \text{island}$

Representing Measurements

- Objects have height / mass / cost ...

- Same length has different measurements (inches / cm / ..)
  - Each unit can be a function
  - length(l1) = Inches(1.5) length(l1) = centimeters(3.8)

- Examples of simple measurements:
  - Diameter(BasketBall1) = Inches(9.5)
  - MRP(BasketBall1) = INR(1200)
  - $d \in days \Rightarrow$ (duration(d) = hours(24))

- Inches(0) and Centimeters(0) refer to the same object, but different from seconds(0)

Representing Measurements

- Not clear how to represent measurement that do not have agreed scale of values
  - Tasty / difficult / scared

- Does not make sense to impose numerical scale
  - Does it mean we cannot do any inference about it?
  - We are generally interested in the relative ordering, not the absolute value
  - As long as we can define $>$ we can make inferences

  - Fried food is tastier than salad
    - $\forall$ f1,f2 f1 $\in$ FriedFood $\wedge$ f2 $\in$ salad $\Rightarrow$ Tastyness(f1) > Tastyness(f2)
  - Tasty things are unhealthy
    - $\forall$ f1,f2 Tastyness(f1) > Tastyness(f2) $\Rightarrow$ HealthScore(f1) < healthscore(f2)

- Such inferences in AI is called qualitative physics
  - Reasoning about physical systems without looking at detailed equations and numerical simulations

Representing Objects

- Real world consists of primitive objects
    - Composite objects are built from primitive objects
    - Part of is useful in many ontologies

- There are objects that cannot be broken into primitive objects
        - Such objects are called Stuff

- If I have a banana and butter in front of me:
    - I can say I see 1 banana (Banana is a thing)
    - Can I say I see 1 butter? (Butter is stuff)

- There are ontologies to deal with this distinction

Representing Objects

- For stuff, we can have a category
  - $\forall$ p, b   b $\in$ Butter $\wedge$ PartOf(p, b) $\Rightarrow$ p $\in$ Butter
  - $\forall$ b   b $\in$ Butter $\Rightarrow$ MeltingPoint(b) = centigrade(30)

- PoundofButter is not stuff, it is a thing
- SalterButter is stuff and is a subcategory of Butter

- Intrinsic properties like density, melting point etc can be attributed to stuff

- Extrinsic properties like weight, length etc can be attributed to things

Events

- We had fluents in the Wumpus world like $HaveArrow^t$
  - Here time is discrete

- How to model continuous time?
  - Like Water filling a bucket:
    - Initial state : Empty bucket   Final state : Full bucket
    - Transitions?

- How to model simultaneous events?
  - The boy was brushing his teeth while waiting for the bucket to fill

- To handle this, we have Event calculus

Event Calculus

- Objects in event calculus are Events, Fluents and Time points
  - At(Alice, Chennai) is a fluent that states Alice is in Chennai

- Alice flew from Chennai to Mumbai
  - Option 1: Make this a predicate (of appropriate arity)

  - Option 2:
    - $E1 \in Flyings \land Flyer(E1, Alice) \land Origin(E1, Chennai) \land Destination(E1, Mumbai)$
    - Since events are also objects, we can add more properties about events:
      - The flight was delayed by 1 hour: delayed(E1) = hour(1)

- Continuous time:
  - T(At(Alice, Chennai), t1, t2) T is special : It takes a predicate as input
  - Happens(E1, t1, t2)

- What all such predicates do we need to model time/events?
  - Starts / terminates / ...

Event Calculus

- Predicates in Event Calculus (1999, Shanahan)
  - T(f, t1, t2) — Fluent f is true for all times between t1 and t2
  - Happens(e, t1, t2) — Event e starts at time t1 and ends at time t2
  - Initiates(e, f, t) — Event e causes Fluent f to become true at time t
  - Terminates(e, f, t) — Event e causes Fluent f to become false at time t
  - Initiated(f, t1, t2) — Fluent f become(s) true at some point between t1 and t2
  - Terminated(f, t1, t2) — Fluent f become(s) false at some point between t1 and t2
  - t1 < t2 — Time point t1 occurs before t2

- Example:
  - E1 ∈ Flyings ∧ Flyer(E1, Alice) ∧ Origin(E1, Chennai) ∧ Destination(E1, Mumbai) ∧ Happens(E, t1, t2) ⇒ Terminates(E, At(Alice, Chennai), t1) ∧ Initiates(E, At(Alice, Mumbai), t2)

Event Calculus

- There is a distinguished start event
  - Describes initial state and fluents that are true/false at the beginning
  - We can then describe what fluent are true/false at what points of time

- Example:
  - Suppose an event e happens between t1 and t3. At time t2 somewhere between t1 and t3, the event e changes the value of the fluent f by initiating it. Then at time t4 in the future (after t3), if no other event has changed the value of f then the fluent f remains True between t2 and t4.
    - $Happens(e, t1, t3) \land Initiates(e, f, t2) \land \neg Terminated(f, t2, t4) \land t1 \leq t2 \leq t3 \leq t4 \Rightarrow T(f, t2, t4)$

- Event calculus can be extended to include:
  - Simultaneous events (two children playing see-saw)
  - Exogenous event (wind moving an object)
  - Continuous event (rise of tides)
  - Non-deterministic events (pressing a button on coffee vending machine)

Time

- Event calculus allows us to talk about time points and time intervals
  - Time points are special time intervals with 0 duration
  - Partition({Moments, ExtendedIntervals}, Intervals)
    i ∈ Moments ⇔ Duration(i)=Seconds(0)

- How to indicate absolute time?
  - Start from an arbitrary absolute time point as 0
    - Jaunary 1, 1900 (GMT)   has time 0
  - Begin and End functions pick the earliest and latest moments in an interval
  - Time function gives a point on the time scale

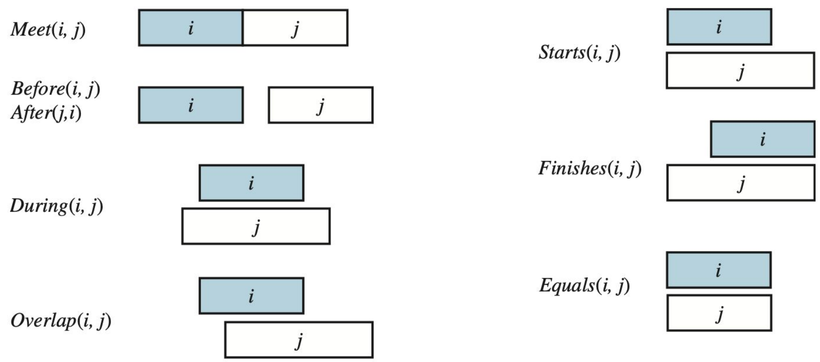
- Examples:
  - Time(Begin(AD1900)) = Seconds(0)
  - Time(Begin(AD2001)) = Seconds(3187324800)
  - Time(End(AD2001)) = Seconds(3218860800)
  - Interval(i) ⇒ Duration(i) = (Time(End(i)) − Time(Begin(i)))
    - Duration(AD2001) = Seconds(31536000)

  - Time(Begin(AD2001)) = Date(0, 0, 0, 1, Jan, 2001)
  - Date(0, 20, 21, 24, 1, 1995) = Seconds(3000000000)

Time Intervals

- Two events can interact in the following 7 ways (Allen, 1983)

- Meet(i, j) ⇔ End(i)=Begin(j)
- Before(i, j) ⇔ End(i) < Begin(j)
- After(j, i) ⇔ Before(i, j)
- During(i, j) ⇔ Begin(j) < Begin(i) < End(i) < End(j)
- Overlap(i, j) ⇔ Begin(i) < Begin(j) < End(i) < End(j)
  - Not symmetric : i should begin before j
- Starts(i, j) ⇔ Begin(i) = Begin(j)
- Finishes(i, j) ⇔ End(i) = End(j)
- Equals(i, j) ⇔ Begin(i) = Begin(j) ∧ End(i)=End(j)

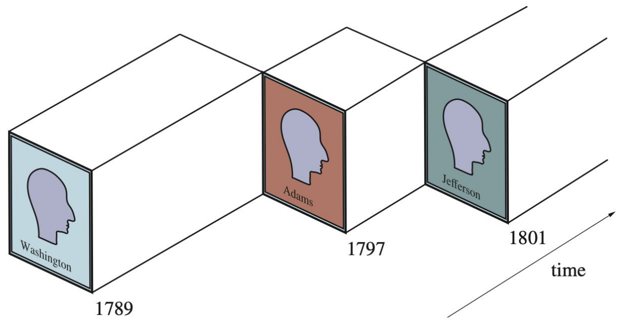


Reign of Elizabeth II immediately followed that of George VI, and the Reign of Elvis overlapped with the 1950s

Meets(ReignOf (GeorgeVI), ReignOf (ElizabethII))
Overlap(Fifties, ReignOf (Elvis))
Begin(Fifties) = Begin(AD1950)
End(Fifties) = End(AD1959)

Fluents and Objects

- Objects can be viewed as generalized events
  - A chunk of space-time
  - USA is an event that started in 1776 with 13 states
  - Population(USA) will be



- Can President(USA) be a term if time is involved?
  - President(USA) can denote a single object that consists of different people along with time intervals.
  - T (Equals(President(USA), GeorgeWashington), Begin(AD1790), End(AD1790))
    - Here we use Equals instead of = because = cannot have predicate as an argument
    - Also, GeorgeWashington and President(USA) are not interpreted as same objects

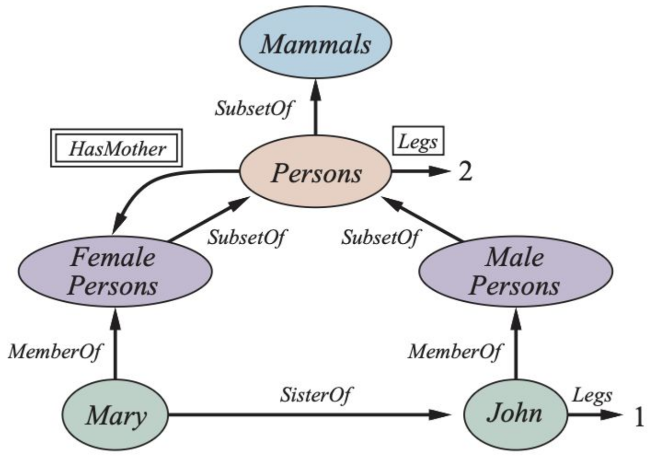
Reasoning Systems with Category

- Two commonly used systems of organizing and reasoning about categories:
  - Semantic Networks
  - Description Logics

- Semantic Networks: Graph representation of categories, objects and relationships among them

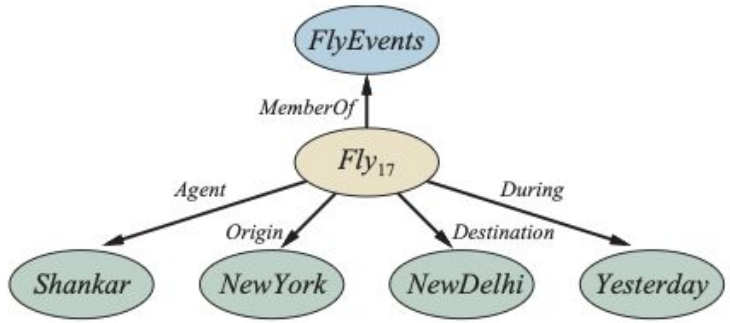
- HasMother is a property
- How many legs does Mary have?
- How many legs does John have?

- Reasoning is through graph traversal
  - Simple and efficient



Semantic Networks

- Multiple inheritance can lead to contradicting inferences
  - We will discuss this later
- How to assert Shankar Flew from NewYork to New Delhi Yesterday ?



- Strictly less expressive that first order logic
  - In particular no existential quantifiers
- There are extensions
  - At the cost of simplicity

Description Logics

- Evolved from semantic networks

- Mainly used to do the following inferences
    - Subumption : A category is a subclass of another category
    - Classification : An object belongs to a category
    - Consistency : An object satisfies the property of the category to which it belongs

$$
\begin{array}{rcl}
\textit{Concept} & \rightarrow & \textbf{Thing} \mid \textit{ConceptName} \\
 & \mid & \textbf{And}(\textit{Concept},\ldots) \\
 & \mid & \textbf{All}(\textit{RoleName},\textit{Concept}) \\
 & \mid & \textbf{AtLeast}(\textit{Integer},\textit{RoleName}) \\
 & \mid & \textbf{AtMost}(\textit{Integer},\textit{RoleName}) \\
 & \mid & \textbf{Fills}(\textit{RoleName},\textit{IndividualName},\ldots) \\
 & \mid & \textbf{SameAs}(\textit{Path},\textit{Path}) \\
 & \mid & \textbf{OneOf}(\textit{IndividualName},\ldots) \\
\textit{Path} & \rightarrow & [\textit{RoleName},\ldots] \\
\textit{ConceptName} & \rightarrow & \textit{Adult} \mid \textit{Female} \mid \textit{Male} \mid \ldots \\
\textit{RoleName} & \rightarrow & \textit{Spouse} \mid \textit{Daughter} \mid \textit{Son} \mid \ldots
\end{array}
$$

- Bachelor = And(Unmarried, Adult, Male)
- FemaleCSEProfs = AND(Faculty, CSEdept, PhD)

- All Men with :
    - at least 3 sons, who are all unemployed, all of whom are married to doctors
    - At most 2 daughters all of whom are professors in Math of Physics
    - AND(Male, Atleast(3,Son), Atmost(2, Daughter), All(Son, AND(Unemployed, Married, All(Spouse,Doctor)), All(Daughter, AND(Professor, Fills(Department, Physics, Math))))

Description Logics

- Inference can be done in Polynomial time
- Description Logic does not have negation and only has limited disjunctions (in Fills and OneOf)

$$
\begin{array}{rcl}
\mathit{Concept} & \rightarrow & \textbf{Thing} \mid \mathit{ConceptName} \\
 & \mid & \textbf{And}(\mathit{Concept}, \ldots) \\
 & \mid & \textbf{All}(\mathit{RoleName}, \mathit{Concept}) \\
 & \mid & \textbf{AtLeast}(\mathit{Integer}, \mathit{RoleName}) \\
 & \mid & \textbf{AtMost}(\mathit{Integer}, \mathit{RoleName}) \\
 & \mid & \textbf{Fills}(\mathit{RoleName}, \mathit{IndividualName}, \ldots) \\
 & \mid & \textbf{SameAs}(\mathit{Path}, \mathit{Path}) \\
 & \mid & \textbf{OneOf}(\mathit{IndividualName}, \ldots) \\
\mathit{Path} & \rightarrow & [\mathit{RoleName}, \ldots] \\
\mathit{ConceptName} & \rightarrow & \mathit{Adult} \mid \mathit{Female} \mid \mathit{Male} \mid \ldots \\
\mathit{RoleName} & \rightarrow & \mathit{Spouse} \mid \mathit{Daughter} \mid \mathit{Son} \mid \ldots
\end{array}
$$

Mental models

- Till now agents perceive things from which they form their belief and derive new beliefs
  - They do not have beliefs about their own beliefs (Introspection)
- Scenario 1:
  - Alice : What is the square root of 400?
  - Bob : I do not know
  - Alice : Think harder.
    Bob realizes that if he puts more more thought, he should be able to figure out the answer
- Scenario 2:
  - Alice : Is Anantha's desktop in his office switched on of switched off ?
  - Bob : I do not know
  - Alice : Think harder.
    Bob should be able to say, there is no way he can get the answer by thinking harder
- Scenario 2:
  - Alice : Is Anantha's desktop in his office switched on of switched off ?
  - Bob : I do not know
  - Alice : Does Anantha know?
    Bob : Yes, obviously!
    Bob should be able to reason about Anantha's knowledge.
- For such reasoning, the agents should be able to model the knowledge of other agents

Knowledge

- Knows : can be a predicate
  - Knows(Alice, Fly(Superman)) here Fly(Superman) is a predicate and not a term
  - Since Clark = Superman we have : { Clark = Superman, Knows(Alice, Fly(Superman)) } ⊨ Knows(Alice, Fly(Clark))
- Referential Transparency : Property depends only on the object that the term is referring to, not the syntax of the term itself
  - If 2+2 = 4 and 4 < 5 then 2+2 < 5
- When modelling Knowledge of agents, we want Referential Opacity
  - Modal Logic can do this

Modal Logic

- Modal operators take sentences as inputs
  - A knows P is represented as $K_A P$
- Syntax is same as First Order Logic with additional operator $K_A P$
- Example (with propositions) : Alice is in Chennai and Bob is in Bangalore.
  - Alice knows that it is raining in Chennai
    - K_{Alice} (RainingInChennai)
  - Bob does not know whether it is raining in Chennai
    - $\neg K_{Bob}$ (RainingInChennai) $\wedge$ $\neg K_{Bob}$ ($\neg$ RainingInChennai)
  - Bob knows that it is either raining in Chennai or not
    - K_{Bob} (RainingInChennai $\vee$ $\neg$ RainingInChennai)
  - Bob knows that Alice knows whether it is raining in Chennai
    - K_{Bob} (K_{Alice} (RainingInChennai) $\vee$ K_{Alice} ($\neg$ RainingInChennai))

Modal Logic

- Models in Modal Logic consists of Possible worlds
  - There is an indistinguishability relationship between the worlds for a given agent based on whether the agent can distinguish between them (also called accessibility relation).
  - $K_A P$ is true at a world w if P is true in all worlds that are accessible by A from w.

- Properties of Knowledge:
  - $K_A(\alpha \Rightarrow \beta) \Rightarrow (K_A \alpha \Rightarrow K_A \beta)$ (Agent knows Modus Ponens)
  - $K_A \alpha \Rightarrow \alpha$ (Agent's knows cannot be false)
  - $K_A \alpha \Rightarrow K_A K_A \alpha$ (Agent can positively introspect)
  - $\neg K_A \alpha \Rightarrow K_A \neg K_A \alpha$ (Agent can negatively introspect)
- Above properties + Axioms of Propositional Logic is a sound and complete axiom system for Propositional Epistemic Logic

- Logical Omniscience : Agent knows all consequences of the axioms $\alpha \Rightarrow K_A \alpha$
  - True for ideal agents
  - Real agents are assumed to have resource bounded inference power

Modal Logic

- Knowledge over predicates.
  - Can describe subtle properties
- Example:
  - Alice knows that someone killed Mary
    - $K_{Alice}(\exists x \, Killed(x, Mary))$
  - Alice knows who killed Mary
    - $\exists x \, K_{Alice}(Killed(x, Mary))$
- There are other Modal Logics variants:
  - Temporal Logic : To reason about Time Always / Until / Sometime in Future / ...
  - Doxastic Logic : Belief (need not be true)
  - Deontic Logic : It is obligatory for adults to play taxes / Request / Permission
  - Term Modal Logic : There exists an Agent who knows alpha

Reasoning about exceptions

- In a semantic network, a subcategory can override a property
- Humans believe in default reasoning unless we are presented with an opposing evidence.
    - If my friend says I have parked my car outside, I will always assume that it has 4 wheels.
    - If I see someone limping, I will always assume that his leg is hurt.
- These are examples of nonmonotonic situations
    - Set of inferences does not monotonically grow as new evidence is presented
- There are two logics to deal with such scenarios:
    - Circumscription
    - Default logic

Circumscription

- Circumscribed Predicates are false for every object except for those specified otherwise
  - Bird(x) ∧ ¬$Abnormal_1$(x) ⇒ Flies(x)

- Here $Abnormal_1$ is a circumscribed predicate
  - It is assumed to be false for every object unless specified otherwise
  - We can add Penguin(x) ⇒ $Abnormal_1$(x)

- Handling multiple inheritance:
  - Let Alice is a scientist (and hence believes Bigbang created the universe) She is also a believer of god (and hence believes God created the universe and not bigbang).
  - Scientist(Alice) ∧ Theist(Alice)
  - Scientist(x) ∧ ¬$Abnormal_2$(x) ⇒ BelievesBigBang(x)
  - Theist(x) ∧ ¬$Abnormal_3$(x) ⇒ ¬ BelievesBigBang(x)
  -
  - If $Abnormal_3$(Alice) is in the knowledgebase, then we can conclude that she believes in god but also in Bigbang
  - But if we have both $Abnormal_2$(Alice) and $Abnormal_3$(Alice) ?
    - Inferencing algorithm (rightly) concludes that it cannot say either.

Default Logic

- Example 1:
    - Bird(x) : Flies(x) / Flies(x)
    - If Bird(x) is true and if Flies(x) is consistent with the KB then Flies(x) is concluded by default

- General Rule looks like : $P : J_1 \dots J_n$ / C
    - If P holds and J_1 .... J_n is consistent with the KB then conclude C

- Example 2:
    - Scientist(x) : BelievesBigBang(x) / BelievesBigBang(x)
    - Theist(x) : ¬ BelievesBigBang(x) / ¬ BelievesBigBang(x)
    - Scientist(Alice) Λ Theist(Alice)

- In default logic, we conclude a property only if it holds in every maximal consistent extension of the initial set S

Nonmonotonic Logic

- Nonmonotonic logics are non-modular
  - Every exception needs to be handled in its own way
  - There is no common way to handle all rules at a time
- Should we have default for every rule?
  - Which rules should have default and which should not?
- My car brakes are always OK
  - Given no other information, the probability that my car's brakes are OK is high enough for me to drive without checking them
  - We can use probability / threshold also to deal with defaults

Belief Revision

- Default is the default status.
  - But when new evidence is presented, some birds do not fly anymore
  - So we need to remove agent's beliefs and all its consequences
- This process is called Belief Revision
  - RETRACT(KB, P)
  - Should remove P and also all inferences that were added to KB that were inferred using P
- Example :
  - KB = { P ⇒ Q, Q, P} (where Q was an inference) then RECTACT(KB, P) = { P ⇒ Q }
  - KB = { P ⇒ Q, Q, P, R ⇒ Q, R} (where Q was an inference) then RECTACT(KB, P) = { P ⇒ Q, R, Q }
- Truth Maintenance Systems are designed to do such revisions efficiently.

Truth Maintenance Systems

- Keep track of the order in which the sentences are added to KB : $P_1 \ldots P_n$
  - To remove P_i first revert to the state just before P_i was added. Then add $P_{i+1} \ldots P_n$
  - Costly if revision happens frequently

- Justification based Truth Maintenance : Keep track of justification for the inferences.
  - KB = { P ⇒ Q, P } then Justification(Q) = { {P ⇒ Q, P} }
  - KB = { P ⇒ Q, P, R ⇒ Q, R} then ustification(Q) = { {P ⇒ Q, P}, {R ⇒ Q, R} }
    - RETRACT(P) will remove every Justification set that contains P.
  - Time for RETRACT(P) is proportional to the number of sentences that were inferred using P.
  - When an inferred sentence is removed from the KB it is marked as out (but its justification is remembered)
    - If later one of the justification set is added back then we do not have to recompute the entire proof

- Assumption based Truth Maintenance : For each sentence keeps track of which assumptions would make the sentence true.

Inference in the age of LLMs

- LLMs are great but
  - They are wrong in many cases
  - We do not know when it is right and when it is wrong
- Can we make LLMs use some Knowledge base?
  - Inferencing takes a lot of time
  - What should be stored in the knowledge base?
- Current research
  - Inference guided LLMs : Get an output from LLM, ask how it arrived at the conclusion. Then use a logical system to verify that the out is correct
    - It is easy to check a proof than to come up with a proof.
    - Guide it by saying at this step the reasoning is incorrect, if there is some flaw.
  - When an LLM says “This is how I arrived at a conclusion” did it really do it or just saying the most likely thing that you will accept?