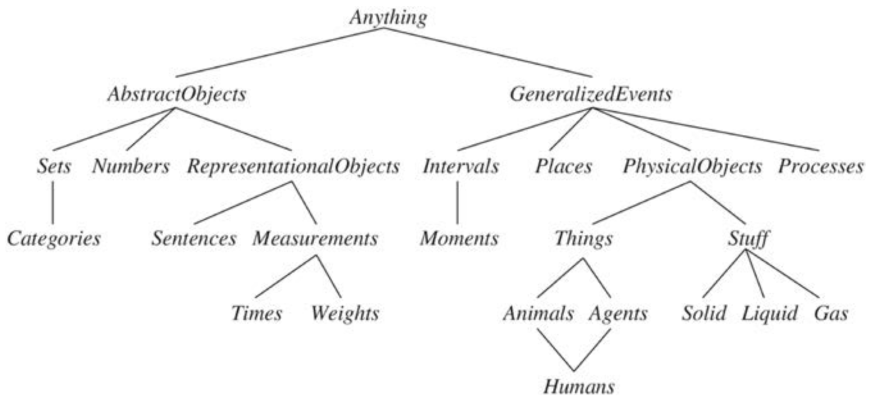
272SM: Introduction to Artificial Intelligence

Knowledge Representation

Ontological Engineering

- Representing abstract concepts, such as events, time, physical objects and beliefs
- Leave placeholders where new knowledge for any domain can fit in → define what it means to be a physical object, details of different types can be filled in later
- **Upper ontology** = general framework of concepts to make simplifying assumptions



Ontological Engineering

- **General-purpose** ontologies:
  - Applicable in (more or less) any special-purpose domain → no representational issue can be finessed
  - In any sufficiently demanding domain, different areas of knowledge must be unified
- None of the top AI applications make use of a general ontology (**special-purpose knowledge** and machine learning)
  - Google Knowledge Graph uses semistructured content from Wikipedia, combining it with other content gathered from across the web under human curation

Categories and Objects

- Organization of objects into **categories**
  - Much reasoning takes place at the level of categories
  - Serve to make predictions about objects once they are classified (using category information)
- Two choices for representing categories in first-order logic: **predicates** *Basketball(b)* and **objects** *Basketballs*
  - *Member(b, Basketballs)* or $b \in Basketballs$: *b* is member of **category** of *basketballs*
  - *Subset(Basketballs, Balls)* or $Basketballs \subset Balls$: *Basketballs* is **subcategory** of *Balls*
- Organize knowledge through **inheritance**
- Subclass relations organize categories into a **taxonomy**
  - Largest taxonomy organizes 10 million living and extinct species into a single hierarchy

Categories and Objects

- **First-order logic** to relate objects to categories or quantify over their members:
  - Object is **member** of category: $BB_9 \in Basketballs$
  - Category is **subclass** of another category: $Basketballs \subset Balls$
  - **All members** of category have some **properties**: $(x \in Basketballs) \Rightarrow Spherical(x)$
  - **Members** of category can be **recognized** by some **properties**: $Orange(x) \wedge Round(x) \wedge Diameter(x) = 9.5'' \wedge x \in Balls \Rightarrow x \in Basketballs$
  - **Category** as a whole has some **properties**: $Dogs \in DomesticatedSpecies$
  - Categories are **disjoint** if they have no members in common: *Disjoint({Animals,Vegetables})*
  - *ExhaustiveDecomposition({Americans,Canadians,Mexicans}, NorthAmericans)*
  - Exhaustive decomposition of disjoint sets is **partition**: *Partition({Animals,Plants,Fungi,Protista,Monera}, LivingThings)*

Physical Composition

- Objects can be grouped into **PartOf** hierarchies, reminiscent of Subset hierarchy: *PartOf(Bucharest,Romania); PartOf(EasternEurope,Europe)*
  - Transitive and reflexive
- Composite objects are often characterized by structural relations among parts: a biped is an object with exactly two legs attached to a body

$$\begin{aligned} Biped(a) \;\Rightarrow\; & \exists l_1, l_2, b \;\; Leg(l_1) \land Leg(l_2) \land Body(b) \land \\ & PartOf(l_1, a) \land PartOf(l_2, a) \land PartOf(b, a) \land \\ & Attached(l_1, b) \land Attached(l_2, b) \land \\ & l_1 \neq l_2 \land [\forall l_3 \;\; Leg(l_3) \land PartOf(l_3, a) \;\Rightarrow\; (l_3 = l_1 \lor l_3 = l_2)] \,. \end{aligned}$$

- Object is composed of parts in its *PartPartition* relation
- Define composite objects with definite parts but no particular structure; “the apples in this bag weigh two pounds” → need **bunch** as albeit object: *BunchOf({*$Apple_1$*,*$Apple_2$*,*$Apple_3$*)}*

Physical Composition

- *BunchOf(Apples)* is **composite object** consisting of all apples - not *Apples*, the **category** or set of all apples
- Define *BunchOf* in terms of *PartOf* relation:

  $\forall x: x \in s \Rightarrow PartOf(x, BunchOf(s))$
- *BunchOf* is the smallest object satisfying this condition, it must be part of any object that has all the elements of s as parts:

  $\forall y: [\forall x: x \in s \Rightarrow PartOf(x,y)] \Rightarrow PartOf(BunchOf(s),y)$
- **Logical minimization**

Measurements

- **Values** we assign for properties of objects: height, mass, cost, etc.
- Universe includes **abstract measure** objects, such as length that can have different names in language, f.ex. 1.5 inches or 3.81 centimeters
- **Units function** represent measures and take number as argument: $Length(L_1) = Inches(1.5) = Centimeters(3.81)$
  - Conversion is done by multiplication: $Centimeters(2.54 * d) = Inches(d)$
- Used to describe objects:
  - $Diameter(Basketball_{12}) = Inches(9.5)$
  - $Weight(BunchOf(\{Apple_1, Apple_2, Apple_3\})) = Pounds(2)$

Measurements

- Measures that cannot be quantified can be compared if they can be **ordered**
  - Norvig’s exercises are tougher than Russell’s:

$$e_1 \in Exercises \land e_2 \in Exercises \land Wrote(Norvig, e_1) \land Wrote(Russell, e_2) \Rightarrow$$
$$Difficulty(e_1) > Difficulty(e_2) .$$

- **Monotonic relationships** among measures form basis for field of qualitative physics
  - Subfield of AI that investigates how to reason about physical systems without detailed equations and numerical simulations

Natural Kinds

- Some categories have **strict definitions**, but natural kind categories don't
  - Tomatoes have **variations**: some are yellow or orange, unripe ones are green, some smaller or larger than average, etc.
  - Problem for a logical agent that cannot be sure that an object it has perceived is a tomato and which of the properties of typical tomatoes this one has → inevitable consequence of **partially observable environments**
  - Useful approach: separate what is true of all instances of a category from what is true only of **typical instances**
    - *Typical(Tomatoes)* maps category to subclass that contains only typical instances
    - Most knowledge about natural kinds will be about their typical instances

      $x \in Typical(Tomatoes) \Rightarrow Red(x) \wedge Round(x)$

Things and Stuff

- Real world consists of **primitive objects** and **composite objects** built from them
- Significant portion of reality that seems to defy any obvious **individuation** (division into distinct objects): **stuff**
- Distinction between **stuff** and **things** (count nouns and mass nouns)
- **Representation of stuff**
  - Recognize a lump of butter as the one left on the table and can pick it up, sell it, whatever → object $Butter_3$
  - Define category *Butter*: its elements will be all those things of which one might say it's butter, also $Butter_3$
  - Any part of a butter-object is also a butter-object: $b \in Butter \land PartOf(p,b) \Rightarrow p \in Butter$

Things and Stuff

- Can define properties, f.ex. Butter melts at 30 degrees centigrade:
  $b \in Butter \Rightarrow MeltingPoint(b, Centigrade(30))$
- **Intrinsic** properties: belong to very substance of object, rather than object as a whole (density, flavor, color, etc.)
- **Extrinsic** properties: not retained under subdivision (weight, length, shape, etc.)
- A category of objects that includes in its definition only intrinsic properties: substance, or **mass noun**
- A class that includes any extrinsic properties in its definition: **count noun**
- Stuff and thing are the most general substance and object categories, respectively

Events/Actions

- **Event calculus** to consider continuous actions
- Objects of event calculus are **events**, **fluents** and **time** points
- Reify events to add any amount of arbitrary information about them

$T(f, t_1, t_2)$	Fluent f is true for all times between t_1 and t_2
$Happens(e, t_1, t_2)$	Event e starts at time t_1 and ends at t_2
$Initiates(e, f, t)$	Event e causes fluent f to become true at time t
$Terminates(e, f, t)$	Event e causes fluent f to cease to be true at time t
$Initiated(f, t_1, t_2)$	Fluent f become true at some point between t_1 and t_2
$Terminated(f, t_1, t_2)$	Fluent f cease to be true at some point between t_1 and t_2
$t_1 < t_2$	Time point t_1 occurs before time t_2

- Extend to represent **simultaneous**, **exogengeous**, **continuous**, and **nondeterministic** events

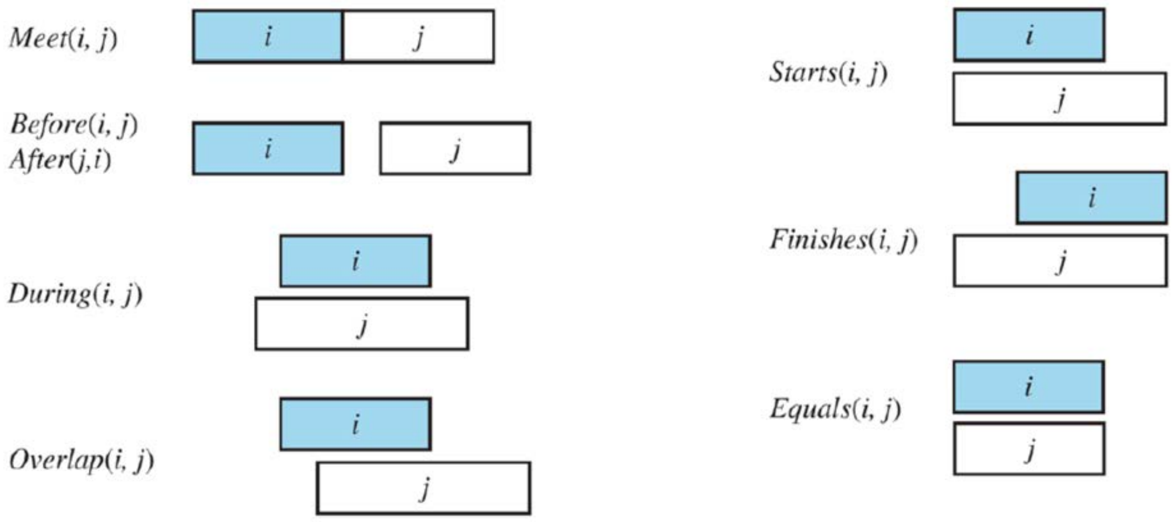
Time

- Time intervals: **moments** and **extended intervals**, only moments have 0 duration
- Invent arbitrary time scale and associate points on scale with moments to get absolute times: measure in seconds, moment at midnight on January 1, 1900 has time 0
  - *Begin* and *End*: pick out earliest and latest moments in an interval
  - *Time*: delivers point on time scale for a moment
  - *Duration*: gives difference between end and start time
  - *Date*: takes 6 arguments (hours, minutes, second, day, month, year) and returns time point

Time Interval Relations

$$
\begin{aligned}
&Meet\,(i,j) &&\Leftrightarrow && End\,(i) = Begin\,(j)\\
&Before\,(i,j) &&\Leftrightarrow && End\,(i) < Begin\,(j)\\
&After\,(j,i) &&\Leftrightarrow && Before\,(i,j)\\
&During\,(i,j) &&\Leftrightarrow && Begin\,(j) < Begin\,(i) < End\,(i) < End\,(j)\\
&Overlap\,(i,j) &&\Leftrightarrow && Begin\,(i) < Begin\,(j) < End\,(i) < End\,(j)\\
&Starts\,(i,j) &&\Leftrightarrow && Begin\,(i) = Begin\,(j)\\
&Finishes\,(i,j) &&\Leftrightarrow && End\,(i) = End\,(j)\\
&Equals\,(i,j) &&\Leftrightarrow && Begin\,(i) = Begin\,(j) \wedge End\,(i) = End\,(j)
\end{aligned}
$$

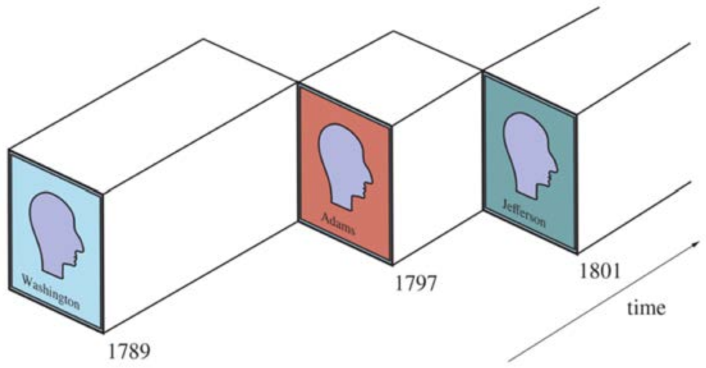
Figure 10.2



Predicates on time intervals.

Fluents and Objects

- **Physical objects** can be viewed as **generalized events**: chunk of space-time
  - F.ex.: USA as an event that began in 1776 as a union of 13 states and is still in progress today as a union of 50
    - Describe **changing properties** using **state fluents**, such as *Population(USA)*
    - *President(USA)* denotes single object that consists of different people at different times: *T(Equals(President(USA),GeorgeWashington),Begin(AD1790),End(AD1790))*: George Washington was president throughout 1790



Mental Objects and Modal Logic

- Agents **have beliefs** and can **deduce new beliefs**, but don't have any knowledge about beliefs or about deduction
- **Knowledge** about **reasoning process** is useful for controlling inference
- Model of **mental objects** that are in someone's head (or something's knowledge base) and of **mental processes** that manipulate those objects
- Agent can have **propositional attitudes** towards mental objects: *Believes, Knows, Wants*, and *Informs*
    - Behave differently from "normal" predicates

Mental Objects and Modal Logic

- Ex.: Lois knows that Superman can fly: *Knows(Lois, CanFly(Superman))*
- We normally think of *CanFly(Superman)* as a sentence, but here it appears as a term → reifying *CanFly(Superman);* making it a fluent
- **Problem**: If it is true that Superman is Clark, then we must conclude that Lois knows that Clark can fly, which is wrong because Lois does not know that Carl is Superman

  (Superman = Clark) ∧ Knows(Lois, CanFly(Superman))

  ⊨ Knows(Lois, CanFly(Clark))
- **Referential transparency**: it doesn't matter that term a logic uses to refer to an object, what matters is the object that the term names
- For propositional attitudes we would like to have **referential opacity**: terms used do matter, because not all agents know which terms are co-referential

Mental Objects and Modal Logic

- **Modal Logic** includes special **modal operators** that take sentences (rather than terms) as arguments
- „A knows P“ = $\mathbf{K}_A P$, **K** is modal operator for knowledge, A an agent, *P* a sentence
- More complicated model of semantics: consists of collection of **possible worlds** rather than just one true world
- Worlds are connected in a graph by **accessibility relations**, one relation for each modal operator
- World w_1 is accessible from world w_0 wrt. modal operator $\mathbf{K}_A$ if everything in w_1 is consistent with what A knows in w_0
- $\mathbf{K}_A P$ is true in world w if and only if *P* is true in every world accessible from w

Mental Objects and Modal Logic

- Truth of more complex sentences is derived by **recursive application** of this rule and the normal rules of first-order logic
- Modal logic can be used to reason about **nested knowledge sentences**: what one agent knows about another agent‘s knowledge
- **Axioms**:
  - Agents can draw **conclusions**: $(\mathbf{K}_a P \wedge \mathbf{K}_a (P \Rightarrow Q)) \Rightarrow \mathbf{K}_a Q$
    - $\mathbf{K}_A(P \vee \neg P)$ is a tautology
    - $(\mathbf{K}_A P) \vee (\mathbf{K}_A \neg P)$ is not a tautology
  - If you know something, it must be **true**: $\mathbf{K}_a P \Rightarrow P$
  - Agents can **introspect** on their own knowledge: $\mathbf{K}_a P \Rightarrow \mathbf{K}_a(\mathbf{K}_a P)$

Mental Objects and Modal Logic

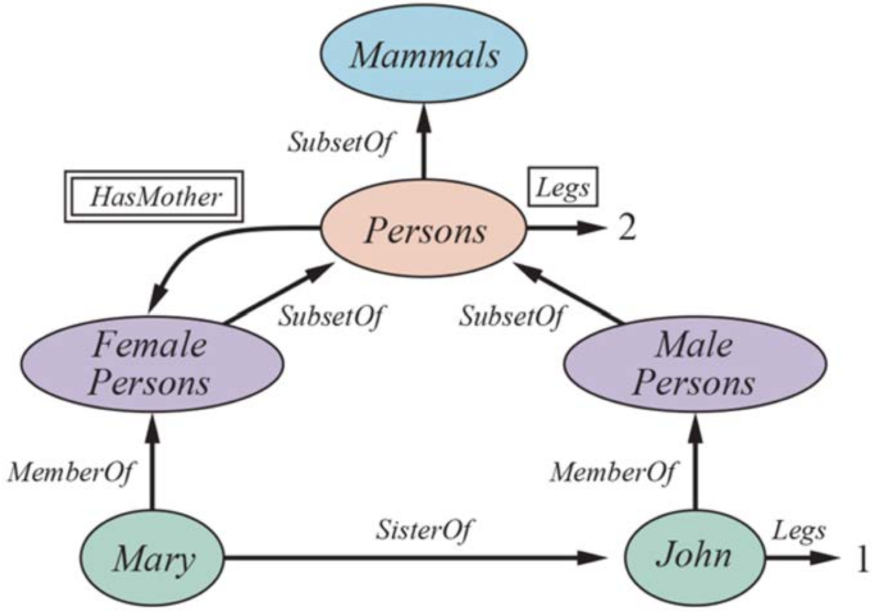
- Similar axioms for **belief** and other modalities
- **Problem**: assumes **logical omniscience** on the part of agents
  - If an agent knows a set of axioms, then it knows all consequences of those axioms
- Other modal logics
  - Add operators for *possibility* and *necessity*
  - Linear temporal logic: *next, finally, globally, until*
  - Deriving additional operators from these makes the logic more complex, but allows to state certain facts in more succinct form

Reasoning System for Categories

- **Semantic networks:**
  - Graphical aids for visualizing a knowledge base
  - Efficient algorithms for inferring properties of an object on the basis of its category membership
- **Description logics:**
  - Formal language for constructing and combining category definitions
  - Efficient algorithms for deciding subset and superset relationships between categories

Semantic Networks

- Represent individual objects, categories of objects, and relations among objects
- Network with 4 objects (John, Mary, 1, 2) and 4 categories:

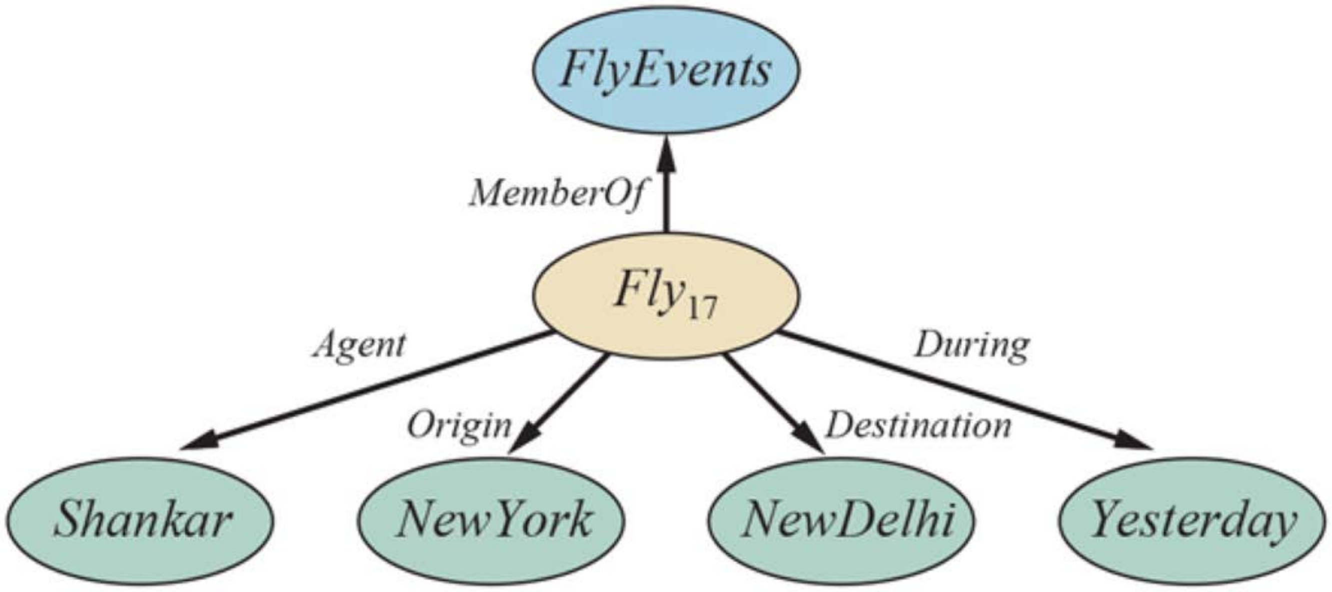


Semantic Networks

- Convenient to perform **inheritance** reasoning → simplicity and efficiency
- **Multiple inheritance** more complicated: object can belong to more than one category or a category can be a subset of more than one other category
  - Algorithm might find 2 or more conflicting values answering the query
  - Banned in some object-oriented programming languages

Semantic Networks

- Drawback: only **binary relations** between bubbles
  - Obtain effect on **n-ary assertions** by **reifying** proposition as an event belonging to an appropriate event category



Semantic Networks

- *Negation*, *disjunction*, *nested function* symbols, and *existential quantification* are still **missing**
- Possible to extend notion to make it equivalent to first-oder logic, but this negates one of main advantages of semantic networks – **simplicity** and **transparency** of inference
- When expressive power proves to be too limiting, many semantic network systems provide for **procedural attachment** to fill in the gaps
  - A query about a certain relation results in a call to a special procedure designed for that relation rather than a general inference algorithm

Semantic Networks

- Ability to represent **default values** for categories
  - F.ex.: John has 1 leg, despite the fact he is a person and all persons have 2 legs
  - Contradiction in a strictly logical KB
- Default semantics is enforced **naturally** by the inheritance algorithm, follows links upwards from the object itself and stops as soon as it finds a value
  - Default is **overridden** by the more specific value

Description Logics

- Notations to easily describe definitions and properties of categories
- Principial inference task:
  - **Subsumption**: checking if one category is a subset of another by comparing their definitions
  - **Classification**: checking whether an object belongs to a category
  - **Consistency**: checking whether the membership criteria are logically satisfiable

Description Logics

- CLASSIC Language
  - Syntax of descriptions in a subset:
    - Algebra of operations on predicates
    - Any description can be translated into an equivalent first-order sentence

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

Description Logics

- Emphasis on **tractability of inference**: problem instance is solved by describing it and then asking if it is subsumed by one of several possible solution categories
  - Ensure that subsumption-testing can be solved in time polynomial in the size of the descriptions
- Either hard problems cannot be stated at all, or they require exponentially large descriptions
  - Tractability results shed lights on **what sorts of constructs cause problems** and helps user to understand how different representations behave

Reasoning with Default Information

- Reasoning processes can violate the **monotonicity property** of logic
- Simple introspection suggests that these failures are widespread in commonsense reasoning
- **Nonmonotonicity**: if new evidence arrives, the default conclusion can be retracted
- **Circumscription**: more powerful and precise version of closed-world assumption
  - Specify particular predicates that are assumed to be "**as false as possible**" – false for every object except those for which they are known to be true

    $Bird(x) \land \neg Abnormal_1(x) \Rightarrow Flies(x)$
  - $Abnormal_1$ is to be **circumscribed** → circumscriptive reasoner assumes $\neg Abnormal_1(x)$ unless $Abnormal_1(x)$ is known to be true
  - Example of model preference logic: sentence is entailed if it is true in all preferred models of the KB
  - Model is preferred if it has fewer abnormal objects

Reasoning with Default Information

- **Default logic:** formalism in which default rules can be written to generate contingent, **nonmonotic conclusions**: *Bird(x):Flies(x)/Flies(x)*
  - If Bird(x) is true, and if Flies(x) is consistent with knowledge base, then Flies(x) may be concluded by default
  - **Default rule**: $P : J_1, \ldots, J_n / C$, where P is the prerequisite, C the conclusion and J_i the justifications (if any of them can be proven false, the conclusion cannot be drawn)
    - Any variable that appears in J_i or C must also appear in P
  - **Extension** of a default theory: maximal set of consequences of the theory
    - Extension S consists of the original known facts and a set of conclusions from the default rules, such that no additional conclusions can be drawn from S, and the justifications of every default conclusion in S are consistent with S

Reasoning with Default Information

- **Truth maintenance systems (TMS)**
  - **Belief revision**: inferred facts turn out to be wrong and will have to be retracted in the face of new information
  - Suppose KB contains a sentence *P*, perhaps a default conclusion recorded by forward-chaining algorithm, and we want to execute *TELL(KB, ¬P)*
    - To avoid creating a contradiction, first execute *RETRACT(KB, P)*
    - Problems arise if any **additional sentences were inferred** from *P* and asserted in the KB
      - $P \Rightarrow Q$ might have been used to add *Q*
      - Obvious solution: retract all sentences inferred from *P* → **fails** because such sentences may have other justifications besides *P* (if *R* and $R \Rightarrow Q$ are also in KB, then *Q* does not have to be removed)
  - TMS are designed to handle these kinds of complications

Reasoning with Default Information

- **Approach:** Keep track of the order in which sentences are told to KB by **numbering** them from P_1 to P_n
  - When call *RETRACT(KB, P_i)* is made, the system reverts to the state just before P_i was added → removing P_i and any inferences that were derived from P_i
  - Sentences P_{i+1} through P_n can then be added again
  - Simple, guarantees KB will be consistent, but requires retracting and reasserting *n-i* sentences & undoing and redoing all inferences from these sentences → **impractical**
- More efficient: **justification-based truth maintenance system (JTMS)**
  - Each sentence in KB is annotated with **justification** consisting of set of sentences from which it was inferred
  - If KB already contains $P \Rightarrow Q$, then *TELL(P)* will cause *Q* to be added with the justification $\{P, P \Rightarrow Q\}$
  - Justification makes retraction **efficient**
  - *Retract(P):* JTMS will delete exactly those sentences for which *P* is a member of every justification
  - When sentence loses all justifications, it is marked as being *out* of KB
    - If subsequent assertion restores one of the justifications, it is marked as being back *in*
    - Retains all inference chains

Reasoning with Default Information

- **Assumption-based truth maintenance system (ATMS)**
  - Efficient context-switching between hypothetical worlds
  - Represents all states that have ever been considered at the same time
  - Keeps track, for each sentence, which assumptions would cause the sentence to be true → label that consists of a set of **assumption sets**, sentence is true only when all the assumptions in one of the assumption sets are true
- TMS provide mechanism for generating **explanations**: explanation of sentence *P* is a set of sentences *E* such that *E* entails *P*
  - If sentences in *E* are already known to be true, then *E* simply provides a sufficient basis for proving that *P* must be the case
  - Can also include **assumptions**: sentences that are not known to be true, but would suffice to prove *P* if they were true