Artificial Intelligence

Beyond First-Order Logic

OUTLINE

- Non-Standard Quantifiers
- Multi-valued Logics
- Non-monotonic Inference in Logic
- Semantic Networks and Inheritance
- Frames, facets and Procedural Attachment
- Beyond Deductive Inference

Non-Standard Quantifiers

- Quantifier Extensions
  - Uniqueness: U(x)P(x)
    If x exists there is only one.
    E.g., Agnostic monotheism: U(x)God(x)
  - Unique Existence: $\exists$U(x)P(x)
    $\forall$(x)$\exists$U(p) [Country(x)→ President(p,x)]
  - Bounded Existence: At-most(N,x) P(x)
    $\forall$(x) At-most(4,y) [Living-grandparents(x,y)]
  - Fuzzy Universals: Most(x)P(x)
    Most(x)[Bird(x)→ Fly(x)]

Multi-Valued Logics

- **The third value: Unknown**
  - Truth values: T, F.?
  - ? = value of variable is unknown
  - Enable inference nonetheless

x	y	(x&y)	(x→y)	(x&y)→y
T	T	T	T	T
T	F	F	F	T
T	?			
F	T			
F	F			
F	?			
?	T			
?	F			
?	?			

Multi-Valued Logics

- **Observations**
  - Conservation of ?'s →

    Indeterminacy does not increase
  - Useful for reasoning under uncertainty
  - Useful for pinpointing needed info
  - Not useful for probabilistic inference

Multi-Valued Logics II

- **The Fourth value: Contradiction**
  - Truth values: T,F,?,#
  - ? Means ***either*** T ***or*** F (unknown)
  - # means ***neither*** *T* ***nor*** F
- **Why is 4-valued logic useful?**
  - Know when you ***don't know*** (inconclusive)→ *seek more knowledge*
  - Know when you ***can't know*** (inconsistency)→ *check accuracy of "known" facts*
  - *"He knoweth not, and worse, he knoweth not that he knoweth not"*—(Some famous dead writer)
  - → Metareasoning: reasoning about what you know (part of Epistemology)

Non-Monotonic Reasoning

- **Inferential Monotonicity Property**

  $\forall(P, A_i), \{A_i, \ldots, A_n\} \mid\text{--}\ P \rightarrow \{A_i, \ldots, A_n, A_{n+1}\} \mid\text{--}\ P$
- **Monotonic Systems**
  - Respect inferential monotonicity
  - FOL with *modus ponens* or resolution is *monotonic*
  - Strict semantic net inheritance is *monotonic*
- **Non-Monotonic Systems**
  - Do not respect inferential monotonicity
  - "Most(x)" quantifier leads to *nonmonotonicity*
  - Default reasoning is *nonmonotonic*
  - Semantic net inheritance with exceptions is *nonmonotonic* (typical kind of inheritance)
  - Metaknowledge inference is *nonmonotonic*

Non-Monotonic Reasoning

- **Example: LOK Inference**
  1. Items of type T_a if true are normally in KB.
  2. Proposition $t_{i,a}$ of type T_a is queried, but not resolvable with KB.
  3. Therefore t_a is presumed false, unless/until later proven true.

Semantic Networks

- **Definition**

<$N_{instances}$, $N_{classes}$, L, I,C(I)>, where:

$N_{instances}$: A set of nodes with individual denotations, such as "Clyde", "box-1", "AI core"

$N_{classes}$: A set of nodes with set or class denotations, such as "CMU students", "Punic War Vets", "Elephants"

L: A set of links (binary relations) over N, such as "brother-of", "eats", "likes"

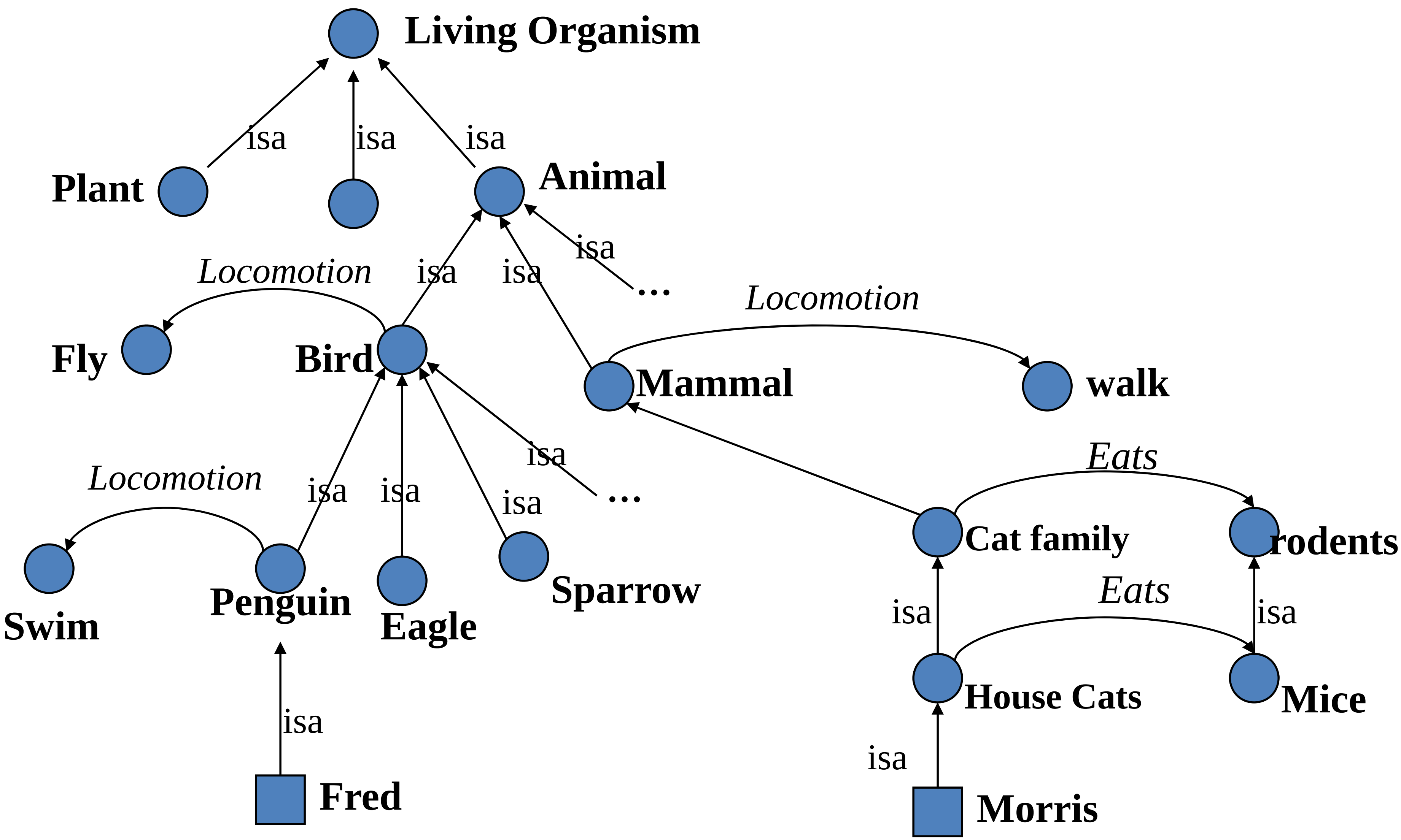
I: Inheritance relations over N and L, such as "isa", "part-of"

C(I): Properties of each inheritance relation, e.g. "isa" is transitive, asymmetric (general-to-specific), non-reflexive, and potentially agglomerative

Semantic Networks

- **Uses of Semantic Nets**
  - Coding static world knowledge
  - Built-in fast inference method (inheritance)
  - Localization of information

Fragment of Semantic Network



Inheritance in Semantic Networks

- **Search Method**
  - Depth-first (good if unique answer)
  - Breadth-first (maximally-local answer)
  - Preferred-parent-first search (greedy HC)
  - All-branch search (agglomerate all closest answers)
  - Complete-search (know when answer is exception to the norm)

Inheritance in Semantic Networks

- **Coping with "Contradiction"**
  - Ignore it: stop at first answer
  - Default overrides on individual paths
  - Report all answers & paths (buck doesn't stop here)
  - Explicit cancel-link semantics (e.g. NETL)
  - Resource-bounded search

    whatever you can find in < N steps)

Frames in Knowledge Representation

PHILOSOPY

Frame system = Semantic Net +
structured nodes +
procedural attachment

INFERENCE PROCESSES:

- Inheritance
- Procedural attachment (demons)
- Frame Matching (a type of unification)

HISTORY

- Minsky, 1975 (first ideas)
- Bobrow & Winograd, 1977 (KRL)
- By 1980 in wide-spread use (FRL, SRL, Units)
- By 1985 in robust packaged form (CRL, KEE, FrameKit,…)
- By 1990 in general use for knowledge bases, and evolved into object-oriented data bases (OODBs)

Frames = Semantic Networks + Meta-structured nodes + Procedural Attachment

FRAME	SLOT	FACET	FILLER
[PC	[isa	[value	COMPUTER]]
	[manufacturer	[type-r	COMPANY]]
	[retail-price	[puller	(* &markup &wholesale)]
		[range-min	500]
		[range-max	10000]
		[unit	USD]]
	[markup	[value	1.5]]
	[owner	[type-r	LEGAL-ANIMATE]]]
[DELL-150/L			
	[isa	[value	PC]]
	[manufacturer	[value	DELL]]
	[processor	[value	pentium-4L]]
	[wholesale	[value	1400]]]

Procedural Attachment

- **Types of Attachments**
  - ***Pullers***(aka “if needed” demons) calculate values on demand (optionally cache)

    Caveat: *control forward cascade*
  - ***Pushers*** (aka “if added” demons) propagate values through network

    Caveat: *check for loops*
  - ***If Referenced***- Tally, check cache validity,…
  - ***If Deleted***- Part of truth-maintenance system Check for cache-validation
  - ***If Changed*** = if deleted + if added

Episodic Knowledge and Scripts

- **Semantic vs. Episodic**
  - Events vs. Facts
  - Temporal and Causal sequences
  - Use Semantic memory as component
- **Scripts**
  - Causally-connected event sequence
  - Generalized by alternate paths:
    - Tree or DAG structure
    - Conditionals on branches
  - Script-role generalization
    - Constants→ Typed variables with restrictions
    - Climb a frame hierarchy

Episodic Knowledge and Scripts

- **Script Application Process**
  - Match Trigger events, including roles
  - Instantiate forwards and backwards ruling out alternate branches
  - Interpolation inference (abduction)
  - Extrapolation inference (prediction)

Types of Inference

- **Deduction**
  - If A, A→B, then B (modus ponens)
  - Truth-preserving, formal reasoning
  - Examples: inheritance, modus ponens, resolution
  - Used in: Proofs, logic, logical argument
  - *Deductive closure:* everything deducible from premises
  - Requires retraction upon contradiction IF non-monotonic
  - Resolution is truth-conditional equivalent to deduction with transitive closure

Types of Inference

- **Abduction**
  - If B, A → B, then perhaps A
  - Seeks "explanation" for B being true
  - Given *closed-world hypothesis* Abduction = Deduction (in reverse).

    E.g. B, A → B, C→ B, then A v C
  - *Abductive closure of A_i:* all legal explanations for A_i.
  - Requires retraction upon contradiction.

Types of Inference (continued)

- **Induction**
  - If P(A), P(B) P(C)..., then $\forall$(x) P(x)
  - If P(A,A), P(B,B), ~P(A,B)... then $\forall$(x) P(x,x)
  - Falsity-preserving inference
  - *Generalization from instances*
  - Used in Machine Learning
  - Requires retraction upon contradiction (e.g. if ~P(Boo,Boo) retract $\forall$(x) P(x,x))

Types of Inference

- **Analogy**
  - If [P(A) → P(B)] & [R(A)→ R(B)] Then perhaps Q(A) → Q(B)
  - Analogy = Induction + Deduction
  - Preserves neither truth nor falsity
  - Yet, very useful:
    - Argumentation and rhetoric
    - Education and explanation
    - Insight for scientific discovery
    - Case-based reasoning & planning