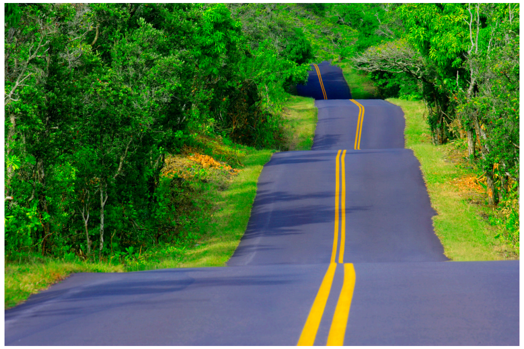
Knowledge-Based Systems

Foundation

Jeff Z. Pan

http://homepages.abdn.ac.uk/jeff.z.pan/pages/

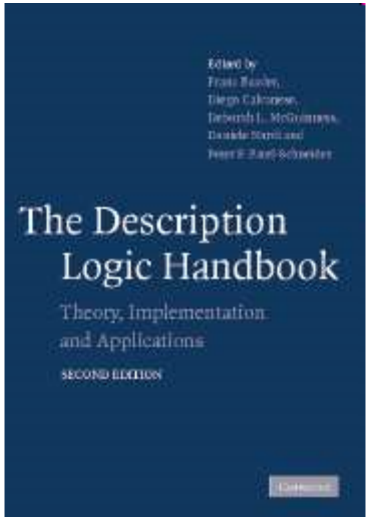
Roadmap



- Foundation
  - KR, ontology and rule; set theory
- Knowledge capture
- Knowledge representation
  - Ontology: Semantic Web standards RDF and OWL, Description Logics
  - Rule: Jess
- Knowledge reasoning
  - Ontology: formal semantics, tableaux algorithm
  - Rule: forward chaining, backward chaining
- Knowledge reuse and evaluation
- Meeting the real world
  - Jess and Java, Uncertainty, Invited talk

Lecture Outline

- Overview of KR (OR Why Ontology/ Description Logics and Rules)
  - Semantic Nets
  - Frames
  - FOL
  - KL-ONE
- Set theory
  - Brief introduction

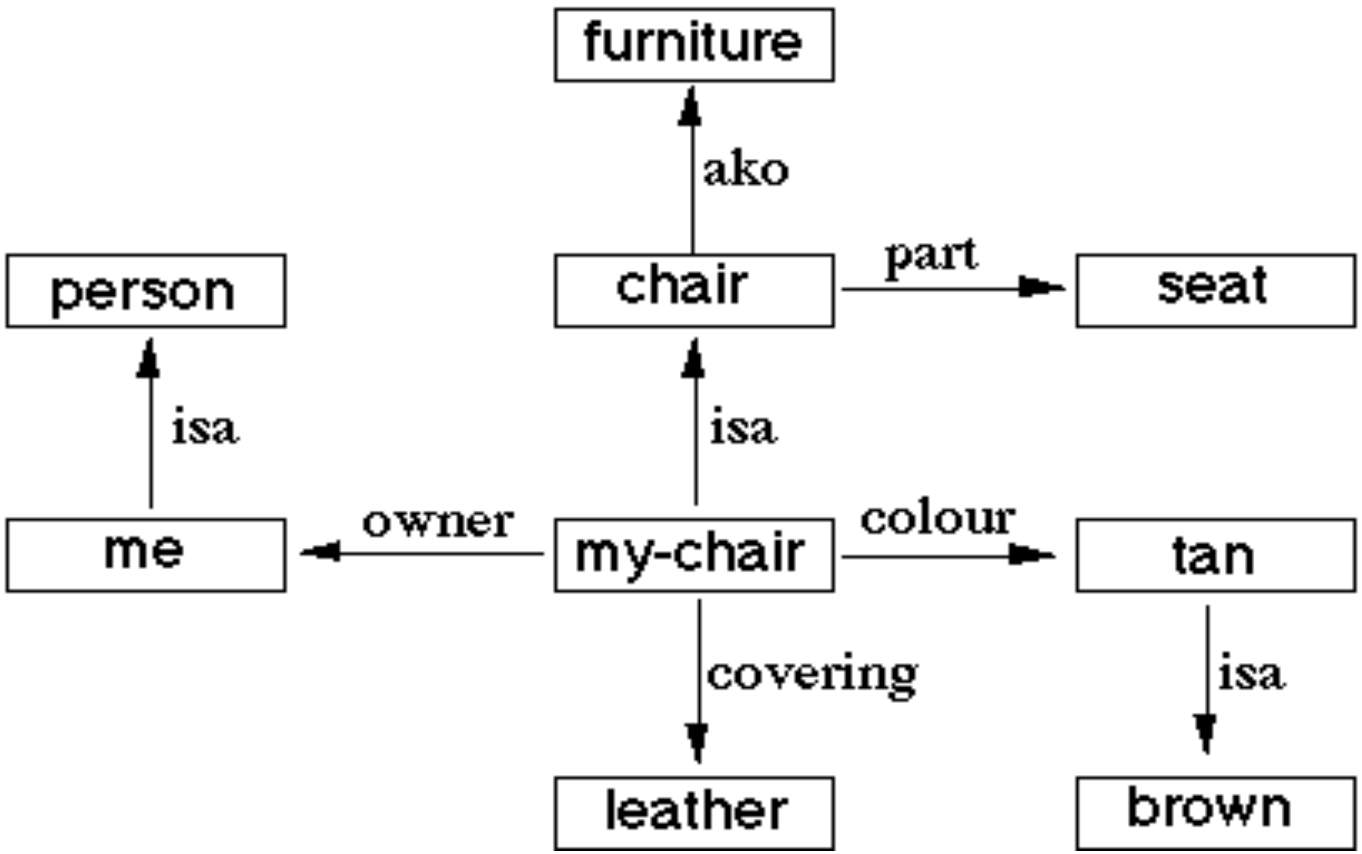


[Section 4.1]

Semantic Nets

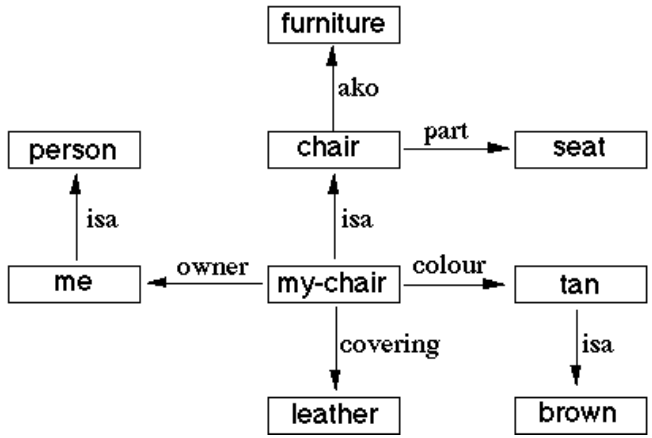
- Proposed by Quillian (1968) to analysis the meaning of words in sentences
  - Later applied to KR
- Basic notations:
  - Nodes: to represent objects, concepts, or situations
  - Arcs: to represent relationships

An Example Semantic Net



Pros/Cons of Semantic Nets

- Easy to follow hierarchy
- Easy to trace association
- Flexible



- No well defined syntax
- No formal semantics
- Not expressive enough to define meaning of labels
- Inefficient

Frames

- A frame [Minsky 1981] represents knowledge
  – About a narrow subject
  – That has default (*) knowledge
- Example:

```
Mammal
   subclass: Animal
   warm_blooded: yes
Elephant
   subclass: Mammal
   * colour: grey
   * size:  large
Cari:
  instance: Elephant
  size: small
```

Procedures in Frames

- Frames allow procedures called demons to be attached to their slots (properties)
  - **if_added:** triggered when a new value is put into a slot
  - **if_removed**: triggered when a value is removed from a slot
  - **if_replaced:** triggered when a slot value is replaced
  - **if_needed:** triggered when there is no value present in an instance frame and a value must be computed from a generic frame

Example: Procedures in Frames

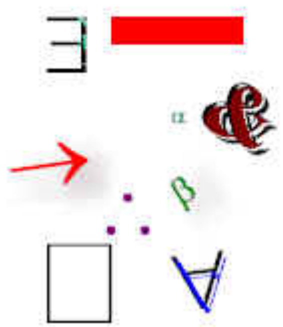
```
age:
    range       1..1500
    if_needed  ask("What  is the age of ", name of this person)
    if_added   if new value > 1500 then
                        print("Your ", this slot, " is too high!");
```

Pros/Cons of Frames

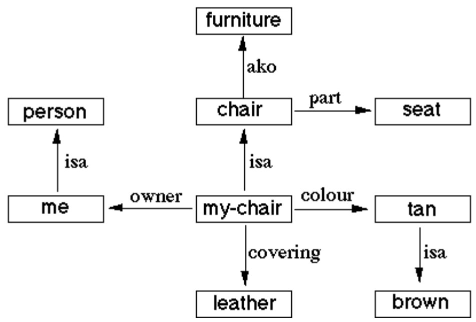
- Easy to include default information and detect missing values
- Easy to create specialised procedures

- Difficult to program
- Difficult for inference

First Order Logic



- Allow to use quantifiers in sentences
  - "for all" ($\forall$), "exists" ($\exists$)
  - Makes sentences more precise



- Example:
  - Chair is a sub-class of furniture
    - $\forall x\ (chair(x) \rightarrow furniture(x))$
  - A chair has a part seat
    - $\forall x\ (chair(x) \rightarrow \exists y\ (part(x,y) \wedge seat(y)))$

UNIVERSITY OF ABERDEEN

Pros/Cons of FOL

- Expressive
- Clear formal semantics

- Syntax too complex
- No support for uncertainty
- Not deci

Notes:

- Being **decidable** means there exists an algorithm returning all and only the correct answers in finite time.

KL-ONE

- Use FOL-based formal semantics, but limit the expressive power
- Formalising Semantic Nets
  - Three kinds of vocabulary
    - Concepts, properties and objects
  - Non-graphic Syntax
  - Formal semantics for built-in relationships
    - Sub-Class-Of
    - Instance-of
  - Provide constructors to define concepts

Example: KL-ONE

- Example:
  - Chair is a sub-class of furniture
    - Chair $\sqsubseteq$ Furniture
  - A chair has a part seat
    - Chair $\sqsubseteq \exists$ *part*.Seat

Pros/Cons of KL-ONE like languages

- Well defined and simplified syntax
- Clear formal semantics
- Good balance between expressive power and decidability

[Later known as **Description Logics**, and used as the Semantic Web standard **ontology language**]

- Not as expressive as FOL
- No support for procedures

Rules

- Another way to limit the expressive power of FOL
- General forms:
  - IF condition-list THEN conclusion-list / action-list
- Example

IF (has-P ?x ?p) and (has-B ?p ?u)
THEN (has-U ?x ?u)

Pros/Cons of Rules

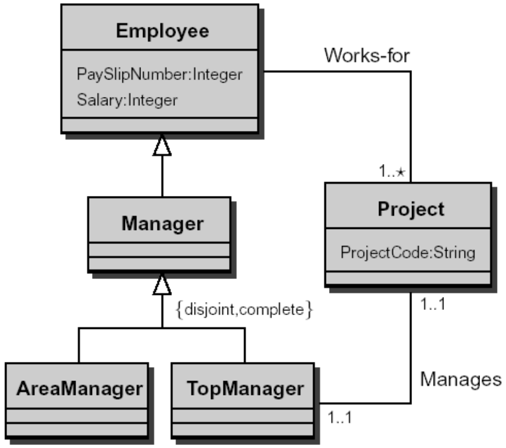
- Simple syntax
- Easy to understand
- High modular
- Production rules support the notion of action

- Hard to follow hierarchies
- No constructors to define concepts

Lecture Outline

- Overview of KR (OR Why Ontology/ Description Logics and Rules)
  - Semantic Nets
  - Frames
  - FOL
  - KL-ONE
- Set theory
  - Brief introduction

Meaning of Basic Constructs



- A concept is a set of instances
  - Employee: {E1, E2, E3, E4}
  - Project: {P1, P2}
- A relation is a set to pairs (tuples) of instances
  - Works-for: {<E1,P1>, <E2,P1>, <E2,P2>, <E3,P1>, <E3,P2>, <E4,P2>}
- Relational representation

Introduction to Set Theory

- A *set* is
  - a structure, representing an *unordered* collection (group, plurality) of
  - zero or more *distinct* (different) objects.
- Set theory deals with
  - operations between,
  - relations among, and
  - statements about sets.

Basic Notations for Sets

- We can denote a set *S* in writing by listing all of its elements in curly braces:
  - {a, b, c} is the set of whatever 3 objects are denoted by a, b, c.
- *Set builder notation*: For any proposition $P(x)$ over any universe of discourse, $\{x|P(x)\}$ is *the set of all x such that P(x).*

  e.g., $\{x \mid x$ is an integer where $x>0$ and $x<5\}$

Definition of Set Equality

- Two sets are declared to be equal *if and only if* they contain exactly the same elements.
- In particular, it does not matter *how the set is defined or denoted.*
- For example: The set {1, 2, 3, 4} =
  $\{x \mid x$ is an integer where $x>0$ and $x<5\}$ =
  $\{x \mid x$ is a positive integer whose square is >0 and $<25\}$

Infinite Sets

- Conceptually, sets may be *infinite* (*i.e.,* not *finite*, without end, unending).

- Symbols for some special infinite sets:
  N = {0, 1, 2, …} The natural numbers.
  Z = {…, -2, -1, 0, 1, 2, …} The integers.
  R = The "**r**eal" numbers, such as 374.1828471929498181917281943125…

Basic Set Relations: Member of

- $x \in S$ ("x is in S") is the proposition that object x is an ∈*lement* or *member* of set S.
  - *e.g.* $3 \in \mathbf{N}$, "a"$\in \{x \mid x$ is a letter of the alphabet$\}$
- Can define set equality in terms of $\in$ relation:
  $\forall S,T$: $S=T \leftrightarrow (\forall x$: $x \in S \leftrightarrow x \in T)$
  "Two sets are equal **iff** they have all the same members."
- $x \notin S :\equiv \neg(x \in S)$ "x is not in S"

Subset and Superset Relations

- $S \subseteq T$ ("S is a subset of T") means that every element of S is also an element of T.
- $S \subseteq T \Leftrightarrow \forall x\ (x \in S \rightarrow x \in T)$
- $\emptyset \subseteq S$, (the empty set is a subset of every set) $S \subseteq S$.
- S is a superset of T means $T \subseteq S$.
- Note $S = T \Leftrightarrow S \subseteq T \wedge T \subseteq S$.
- $S \nsubseteq T$ means $\neg(S \subseteq T)$, *i.e.* $\exists x (x \in S \wedge x \notin T)$

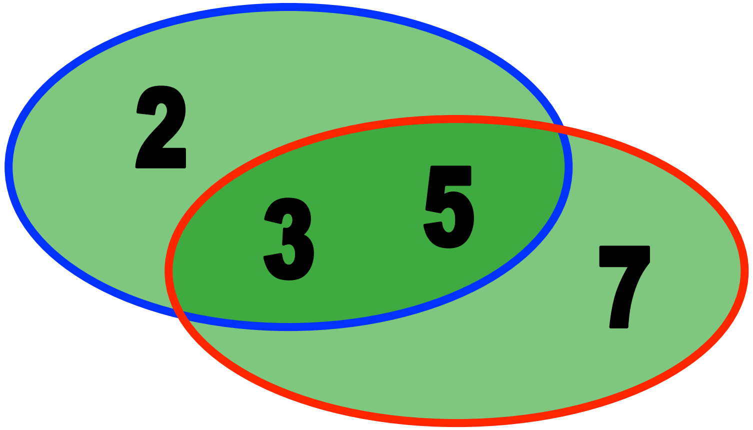
The Union Operator

- For sets A, B, their *union* $A \cup B$ is the set containing all elements that are either in A, **or** (“∨”) in B (or, of course, in both).
- Formally, $\forall A,B$: $A \cup B = \{x \mid x \in A \lor x \in B\}$.
- Note that $A \cup B$ contains all the elements of A **and** it contains all the elements of B:
  $\forall A, B$: $(A \cup B \subseteq A) \land (A \cup B \subseteq B)$

Union Examples

Required Form

- {a,b,c}∪{2,3} = {a,b,c,2,3}
- {2,3,5}∪{3,5,7} = {2,3,5,3,5,7} = {2,3,5,7}



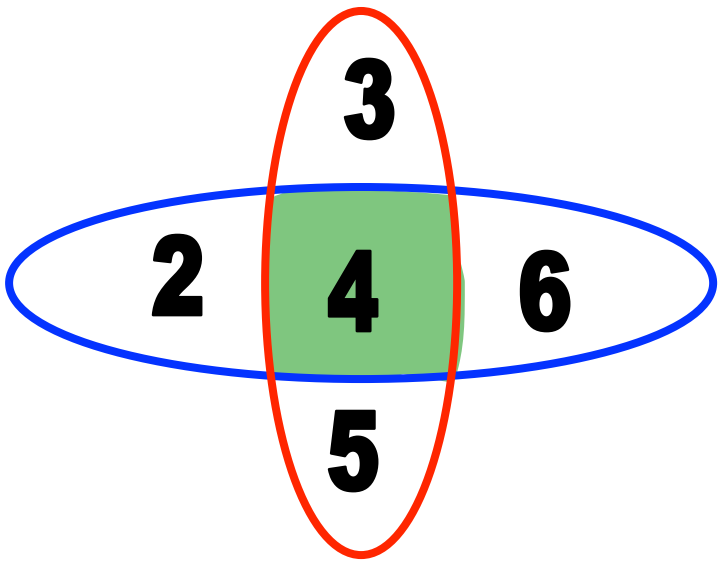
UNIVERSITY OF ABERDEEN

The Intersection Operator

- For sets A, B, their *intersection* $A \cap B$ is the set containing all elements that are simultaneously in A **and** ("$\wedge$") in B.
- Formally, $\forall A,B$: $A \cap B \equiv \{x \mid x \in A \wedge x \in B\}$.
- Note that $A \cap B$ is a subset of A **and** it is a subset of B: $\forall A, B$: $(A \cap B \subseteq A) \wedge (A \cap B \subseteq B)$

Intersection Examples

- $\{a,b,c\} \cap \{2,3\} = \underline{\varnothing}$
- $\{2,4,6\} \cap \{3,4,5\} = \underline{\{4\}}$



Set Difference

- For sets A, B, the *difference of A and B*, written A-B, is the set of all elements that are in A but not B.
- $A - B :\equiv \{x \mid x \in A \wedge x \notin B\}$
  $= \{x \mid \neg(x \in A \rightarrow x \in B) \}$
- Also called:
  The *complement of B with respect to A*.

Set Difference Examples

- $\{1,\cancel{2},\cancel{3},4,\cancel{5},6\} - \{2,3,5,7,9,11\} =$
- {1,4,6}
- **Z** - **N** = {… , -1, 0, 1, 2, … } - {0, 1, … }
  = {*x* | *x* is an integer but not a nat. #}
  = {*x* | *x* is a negative integer}
  = {… , -3, -2, -1}

Set Complements

- The *universe of discourse* can itself be considered a set, call it *U*.
- The *complement* of *A*, written $\overline{A}$, is the complement of *A* w.r.t. *U*, *i.e.*, it is *U*-*A.*
- *E.g.,* If *U*=**N**,

$$\overline{\{3,5\}} = \{0,1,2,4,6,7,\ldots\}$$

Summary

- Knowledge representation formalisms
  - Semantic Nets
  - Frames
  - FOL
  - KL-ONE (DL/Ontology)
  - Rules
- Set theory is a useful tool to understand the semantics of
  - Ontology
  - Rule

Acknowledgements

- Prof. Enrico Franconi (University of Bolzano)

- Prof. George Bebis (University of Nevada)