

Princess Nora University
Faculty of Computer & Information Systems

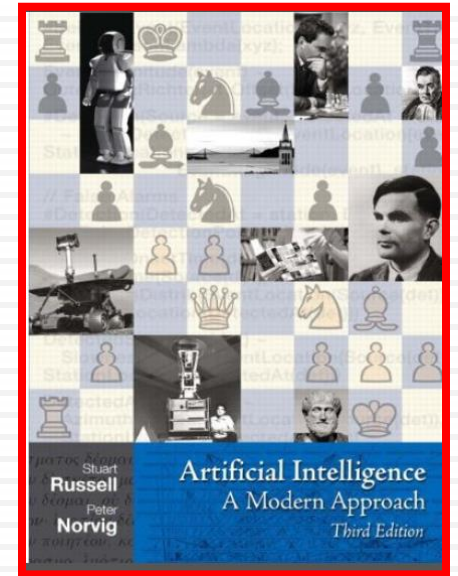


ARTIFICIAL INTELLIGENCE

(CS 370D)



جامعة الأميرة نورة بنت عبد الرحمن
Princess Nora Bint Abdul Rahman University



(CHAPTER-7) LOGICAL AGENTS



Some General Representations

1. Logical Representations
2. Production Rules
3. Semantic Networks
 - Conceptual graphs, frames, scripts
4. *Description Logics* (not covered in this course)





Non-Logical Representations?





Non-Logical Representations?

1. Production rules
2. Semantic networks
 - Conceptual graphs
 - Frames
 - Scripts





Production Rules





Production Rules

- Rule set of $\langle \text{condition}, \text{action} \rangle$ pairs
 - “if condition then action”
- Match-resolve-act cycle
 - **Match:** Agent checks if each rule’s condition holds
 - **Resolve:**
 - Multiple production rules may fire at once (**conflict set**)
 - Agent must choose rule from set (**conflict resolution**)
 - **Act:** If so, rule “fires” and the action is carried out





IF the interest-rate out look is down,
THEN do not buy money-market funds..

Rules-of-Thumb

- An apple a day keeps the doctor away .
- A stitch in time saves nine .



Fuzzy Rules

IF you're **old**,
THEN you have owned **several** homes .

IF you have owned **several** homes THEN
you have had **numerous** headaches .

IF the interest-rate out look is **up** and the risk
you can accept is **low**,
THEN buy a conservative money-market
fund .



IF the interest-rate out look is **up** and the risk
you can accept is **high**,
THEN buy **aggressive** money-market fund .

Rules with certainty factors

IF the patient is sneezing,
AND has a runny nose,
AND has watery eyes,
THEN the patient has cold, **CF=0.5** .





Production Rules Example

- **IF** (at bus stop **AND** bus arrives)
THEN action(get on the bus)
- **IF** (on bus **AND** not paid **AND** have oyster card)
THEN action(pay with oyster) **AND** add(paid)
- **IF** (on bus **AND** paid **AND** empty seat)
THEN sit down





Inference Engine

- **The inference engine** is a generic control mechanism for navigating through and manipulating knowledge and **deduce results** in an organized manner
- It applies a specific task take data and drive conclusions
- The inference engine is the part of the system that chooses which facts and rules to apply when trying to solve the user's query





Inference Engine

- The **forward chaining** , **backward chaining** and **tree search** are some of the techniques used for drawing inferences from the knowledge base

Inferences from rules

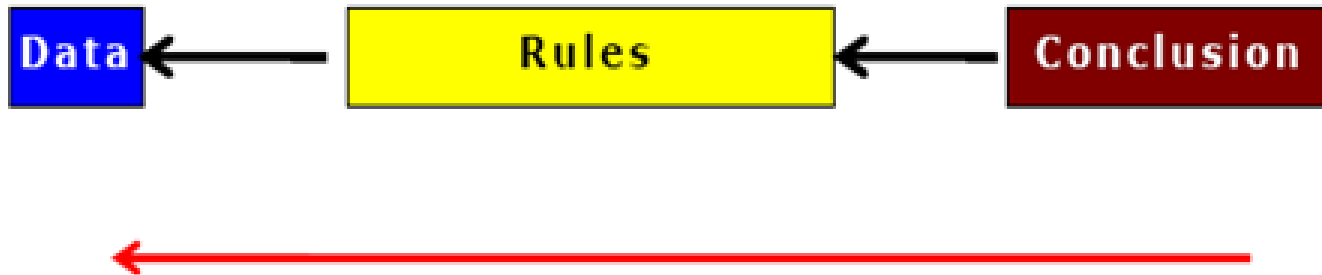
1. Goal driven = **backward chaining**
2. Data driven = **forward chaining**





➤ Goal driven or backward chaining

An inference technique which uses IF-THEN rules to repetitively break a goal into smaller sub-goals which are easier to prove





Example : KB contains Rule set :

Rule 1: if A and C then F

Rule 2: if A and E then G

Rule 3: if B then E

Rule 4: if G then D



■ Solution :

- (i) † Start with goal ie **D** is true
- † go backward/up till a rule "fires" is found.

First iteration :

- (ii) † **Rule 4** fires :
- † new sub goal to prove **G** is true
- † go backward
- (iii) † **Rule 2** "fires"; conclusion: **A** is true
- † new sub goal to prove **E** is true
- † go backward;
- (iv) † no other rule fires; end of first iteration.
- † new sub goal found at (iii);
- † go for second iteration

Second iteration :

- (v) † **Rule 3** fires :
- † conclusion **B** is true (2nd input found)
- † both inputs **A** and **B** ascertained
- † **Proved**





Data driven or Forward chaining

An inference technique which uses IF-THEN rules to deduce a problem solution from initial data





Example :

KB contains Rule set :

Rule 1:	If A and C	Then	F
Rule 2:	If A and E	Then	G
Rule 3:	If B	Then	E
Rule 4:	If G	Then	D

Problem : prove

If A and B true Then D is true



■ Solution :

- (i) † Start with input given **A, B** is true and then
- † start at **Rule 1** and go forward/down till a rule "fires" is found.

First iteration :

- (ii) † **Rule 3** fires : conclusion **E** is true
- † new knowledge found
- (iii) † No other rule fires;
- † end of first iteration.
- (iv) † Goal not found;
- † new knowledge found at (ii);
- † go for second iteration

Second iteration :

- (v) † **Rule 2** fires : conclusion **G** is true
- † new knowledge found
- (vi) † **Rule 4** fires : conclusion **D** is true
- † Goal found;
- † Proved





Advantages of Rules

- Rules are easy to understand
- Inference and explanation are easy to derive
- Modifications and maintenance are relatively easy
- Uncertainty is easily combined with rules
- Each rule is usually independent of all others





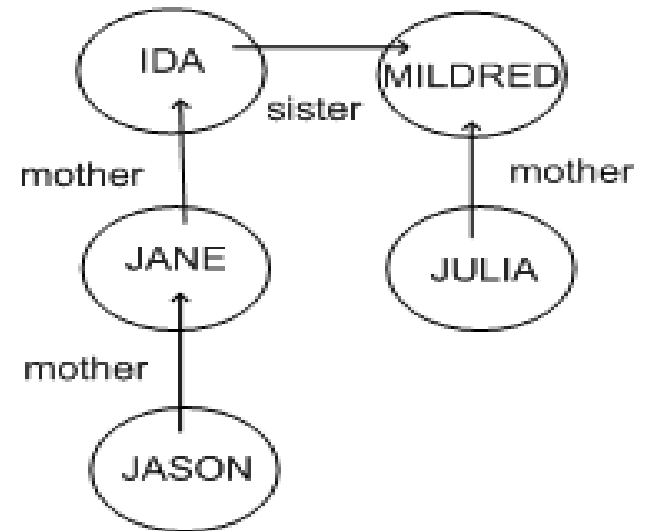
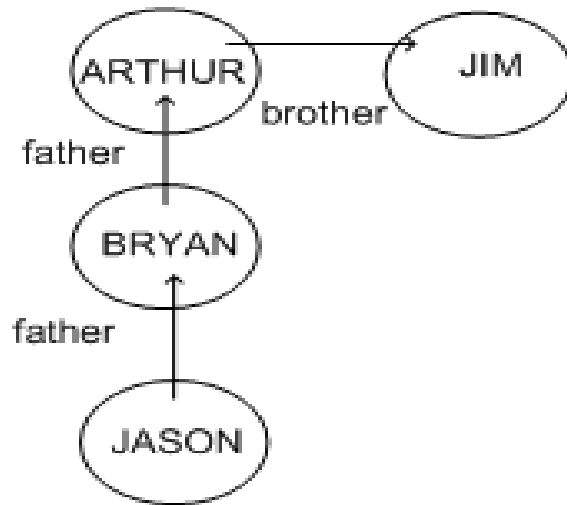
Graphical Representation





Graphical Representation

- Graphs easy to store in a computer
- To be of any use must impose a formalism

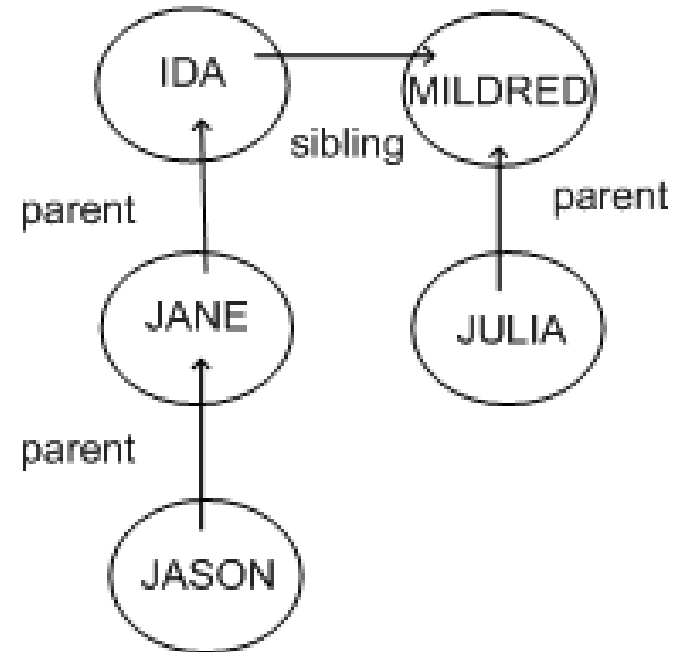
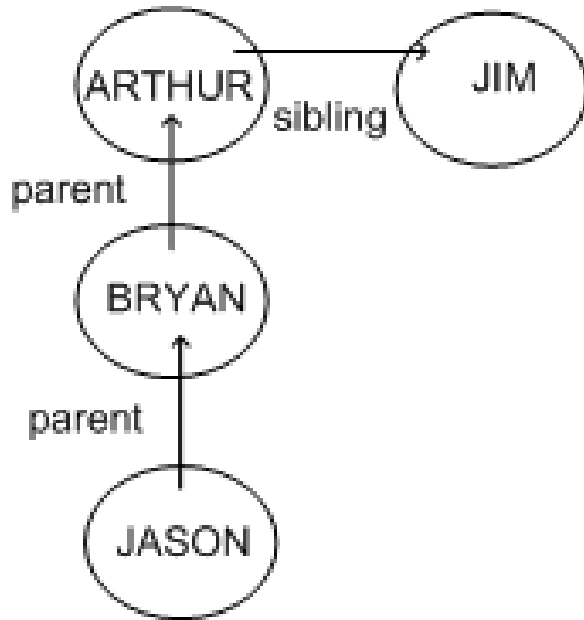


- ▣ Jason is 15, Bryan is 40, Arthur is 70, Jim is 74
- ▣ How old is Julia?





Semantic Networks



- Because the syntax is the same
 - ▣ We can guess that Julia's age is similar to Bryan's
- Formalism imposes restricted syntax





Semantic Networks

- Graphical representation (a graph)
 - ▣ Links indicate subset, member, relation, ...
- Equivalent to logical statements (usually FOL)
 - ▣ Easier to understand than FOL?
- Example: natural language understanding
 - ▣ Sentences with same meaning have same graphs
 - ▣ e.g. Conceptual Dependency Theory (Schank)





Semantic Networks

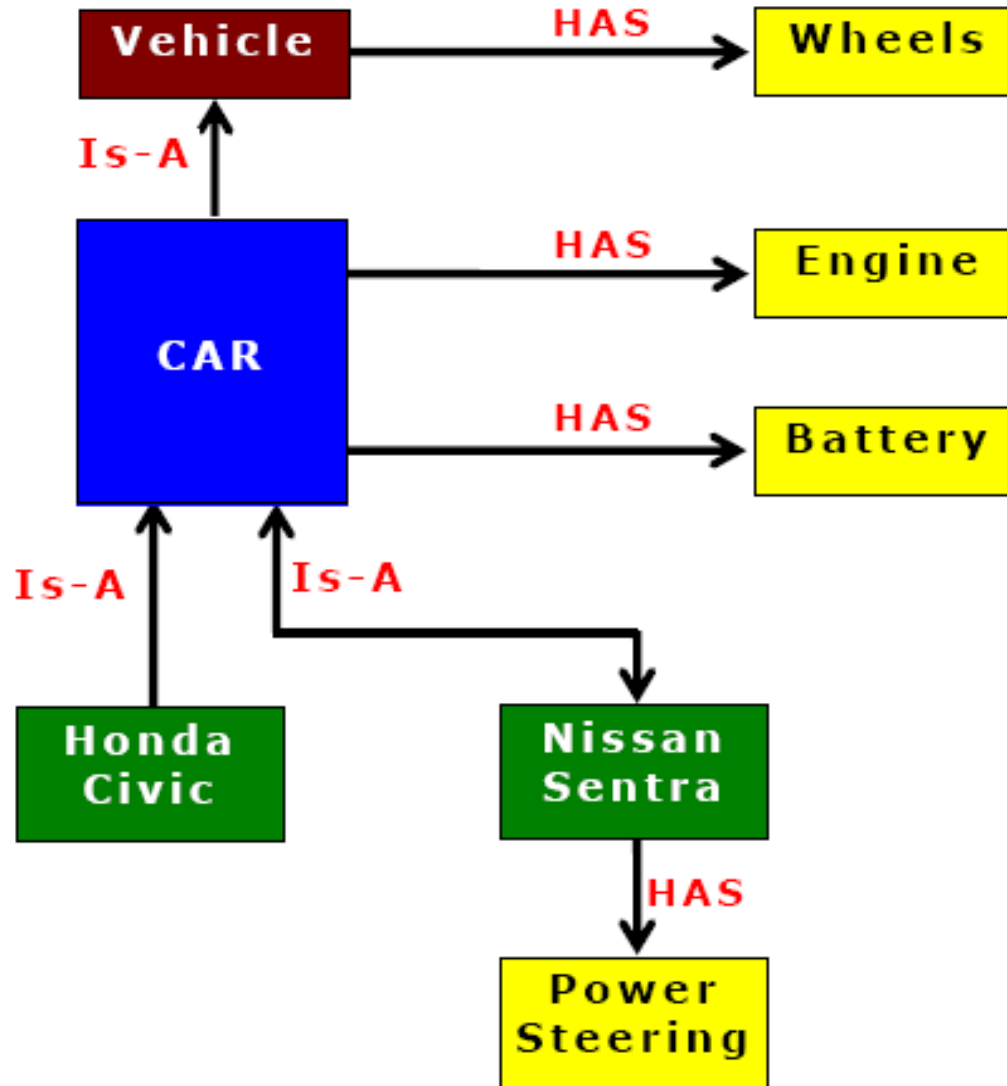
In this scheme , knowledge is represented in terms of objects and relationships between objects

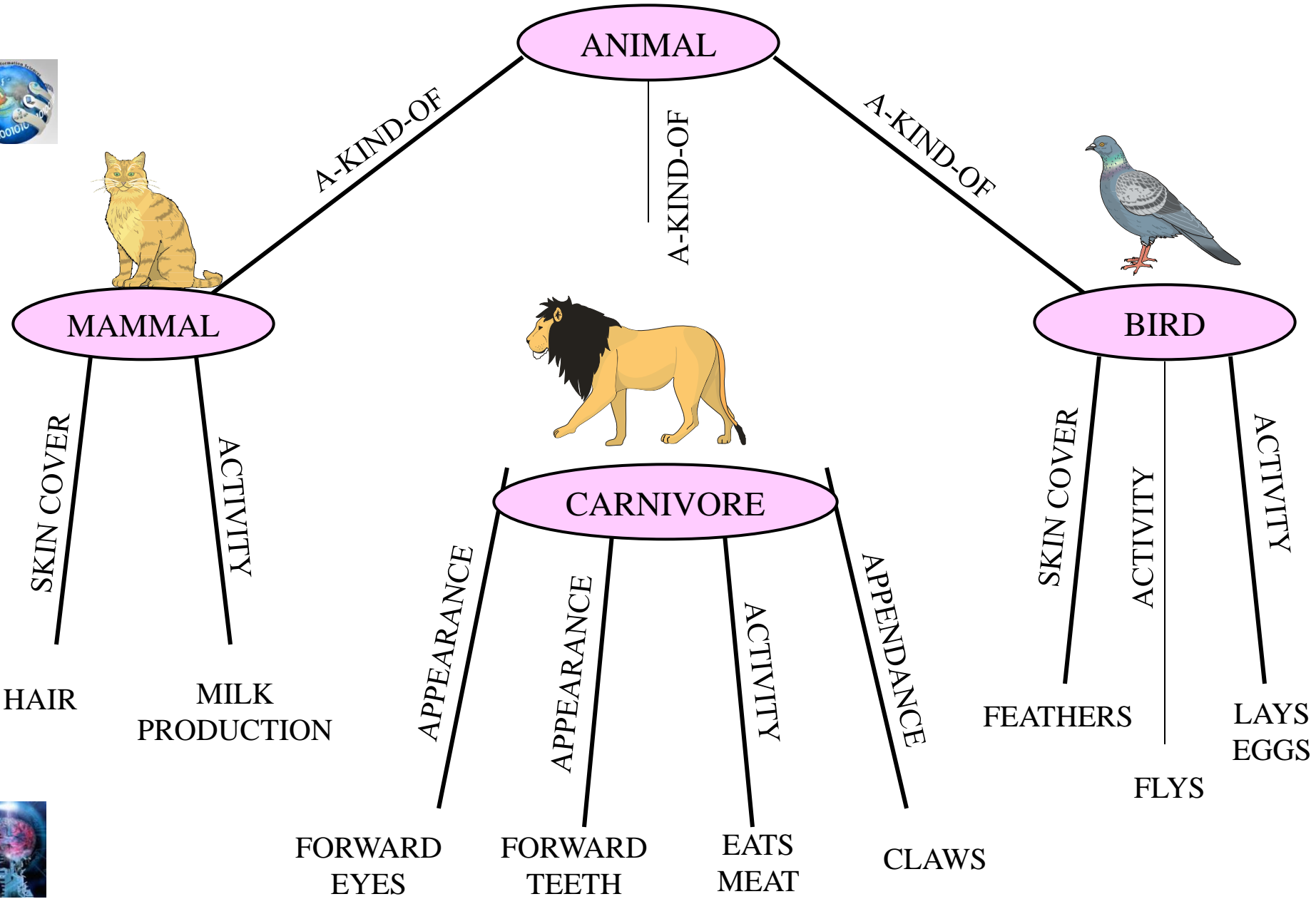
The objects are denoted as nodes of a graph. The relationship between two objects are denoted as a link between the corresponding two nodes

The most common form of semantic network uses the link between nodes to represent **IS-A** and **HAS** relationships between objects



Example of semantic network







Frames

- In this technique, knowledge is decomposed into highly modular pieces called frames, which are generalized record structures
- Knowledge consist of **concepts**, **situation**, **attributes** of concepts , **relationships** between concepts , and **procedure** to handle relationships
 - Each **concept** may be represented as a separate frame
 - The **attributes**, the **relationships** between concepts and the **procedures** are allotted to slots in a frame
 - The contents of a slot may be of any data type –**numbers**, **strings**, **functions** or **procedures** and so on
 - The frames may be linked to other frames, providing the same kind of inheritance as that provided by a semantic network





Frame Representations

- Semantic networks where nodes have structure
 - ▣ Frame with a number of slots (age, height, ...)
 - ▣ Each slot stores specific item of information
- When **agent** faces a new situation
 - ▣ Slots can be filled in (value may be another frame)
 - ▣ **Filling in** may trigger **actions**
 - ▣ May trigger **retrieval** of other frames
- Inheritance of properties between frames
 - ▣ Very similar to objects in OOP





Frames

- Basic frame design

Frame Name:

Class:

Properties:

Property1	Value1
Property2	Value2
***	***
***	***



Frame Representation of the “cow”

Frame Name:

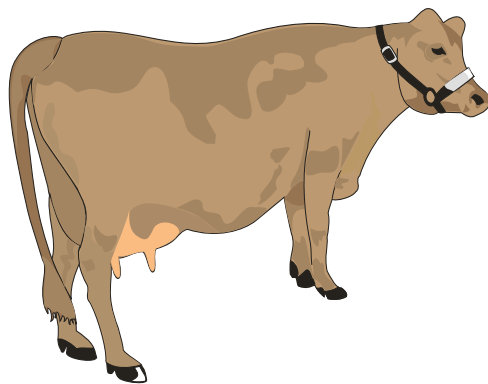
Cow

Class:

Mammal

Properties:

SKIN Cover	Hair
Activity	Produce Milk
tail	Long tail
***	***





Chair frame





Flexibility in Frames

- **Slots in a frame can contain**
 - Information for choosing a frame in a situation
 - Relationships between this and other frames
 - Procedures to carry out after various slots filled
 - Default information to use where input is missing
 - Blank slots: left blank unless required for a task
 - Other frames, which gives a hierarchy





Thank you



End of

Chapter 7- part2

Dr. Abeer Mahmoud
(course coordinator)

