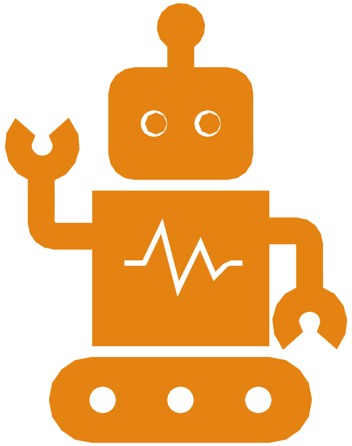


CS341 Artificial Intelligence

Lecture 8

DR. HEBA MOHSEN

Logical Agents

- **Logical agents** are designed that can form representations of a complex world, use a process of inference to derive new representations about the world, and use these new representations to deduce what to do.

Logical Agents

- Humans, it seems, know things; and what they know helps them do things.
- AI concerns about how the intelligence of humans is achieved not by purely reflex mechanisms but by processes of **reasoning** that operate on internal **representations** of knowledge. And this approach is represented in the **knowledge-based agents (Logical agents)**.

Knowledge-based Agents

- Knowledge-based agents can accept new tasks in the form of explicitly described goals
- They can achieve competence quickly by being told or learning new knowledge about the environment
- They can adapt to changes in the environment by updating the relevant knowledge.

Knowledge-based Agents

- The central component of a knowledge-based agent is its **knowledge base**, or KB.
- A knowledge base is a set of sentences. Each sentence is expressed in a language called a knowledge representation language and represents some assertion about the world.
- There must be a way to add new sentences to the knowledge base and a way to query what is known and Both operations may involve inference which is deriving new sentences from old.

Knowledge-based Agents

- Each time the agent program is called, it does:

1. it TELLs the knowledge base what it perceives.
2. It ASKs the knowledge base what action it should perform.
   - In the process of answering this query, extensive reasoning may be done about the current state of the world, about the outcomes of possible action sequences, and so on.
3. The agent program TELLs the knowledge base which action was chosen, and the agent executes the action.

Knowledge Representations

For certain problem solving techniques

- The “best” representation has already been worked out
- Often it is an obvious requirement of the technique
- Or a requirement of the programming language (e.g., Prolog)

Examples:

- First order theorem proving (first order logic)
- Inductive logic programming (logic programs)
- Neural networks learning (neural networks)

Some general representation schemes

- Suitable for many different (and new) AI applications

Four General Representation Types

1. Logical Representations
2. Production Rules
3. Semantic Networks
4. Frames

What is a Logic?

A language with concrete rules
- No ambiguity in representation (may be other errors!)
- Allows unambiguous communication and processing
- Very unlike natural languages e.g. English

Many ways to translate between languages
- A statement can be represented in different logics
- And perhaps differently in same logic

Expressiveness of a logic
- How much can we say in this language?

Syntax and Semantics of Logics

Syntax

- How we can construct legal sentences in the logic
- Which symbols we can use (English: letters, punctuation)
- How we are allowed to write down those symbols

Semantics

- How we interpret (read) sentences in the logic
- i.e., what the meaning of a sentence is

Example: “All lecturers are six foot tall”

- Perfectly valid sentence (syntax)
- And we can understand the meaning (semantics)
- This sentence happens to be false (there is a counterexample)

Propositional Logic

Syntax

- Propositions, e.g. “it is wet”
- Connectives: and, or, not, implies, iff (equivalent)

$$\wedge \quad \vee \quad \neg \quad \rightarrow \quad \leftrightarrow$$

- Brackets, T (true) and F (false)

Semantics (Classical AKA Boolean)

- Define how connectives affect truth
  - “P and Q” is true if and only if P is true and Q is true
- Use **truth tables** to work out the truth of statements

Example

Let A denote "The earth is round"
Let B denote "The sun is cold"
Let C denote "It rains in Spain"

Then

(1) $A \wedge B$: "The earth is round and the sun is cold"

(2) $A \vee (\neg B)$: "Either the earth is round or the sun is not cold"

(3) $A \wedge (B \vee C)$: "The earth is round and either the sun is cold or it rains in Spain"

Predicate Logic

Propositional logic combines atoms
- An atom contains no propositional connectives
- Have no structure (today_is_wet, john_likes_apples)

Predicates allow us to talk about objects
- Properties: is_wet(today)
- Relations: likes(john, apples)
- True or false

In predicate logic each atom is a predicate

There are two types of predicate logic: First-order logic and Higher order logic

First and Higher Order Logic

First Order Logic:

- More expressive logic than propositional
- **Constants** are objects: john, apples
- **Predicates** are properties and relations:
  - likes(john, apples)
- **Functions** transform objects:
  - likes(john, fruit_of(apple_tree))
- **Variables** represent any object: likes(X, apples)
- **Quantifiers** qualify values of variables
  - True for all objects (Universal): $\forall$X. likes(X, apples)
  - Exists at least one object (Existential): $\exists$X. likes(X, apples)

A **Higher Order Logic** allows predicates to accept arguments which are themselves predicates.

Example: FOL Sentence

"Every rose has a thorn"

$$\forall X.(rose(X) \rightarrow \exists Y.(has(X, Y) \wedge thorn(Y)))$$

For all X

- if (X is a rose)
- then there exists Y
  - (X has Y) and (Y is a thorn)

Logic is a Good Representation

Fairly easy to do the translation when possible

Branches of mathematics devoted to it

It enables us to do logical reasoning
- Tools and techniques come for free

Basis for programming languages
- Prolog uses logic programs (a subset of FOL)

Why Non-Logical Representations?

Production rules

Semantic networks

- Conceptual graphs
- Frames

Logic representations have restrictions and can be hard to work with that's why many AI researchers searched for better representations

Production Rule Representations

Consists of <condition,action> pairs

Agent checks if a condition holds

- If so, the production rule "fires" and the action is carried out
- This is a recognize-act cycle

Given a new situation (state)

- Multiple production rules will fire at once
- Call this the **conflict set**
- Agent must choose from this set
  - Call this **conflict resolution**

Production system is any agent

- Which performs using recognize-act cycles

Production Rules Example

IF (at bus stop AND bus arrives) THEN action(get on the bus)

IF (on bus AND not paid AND have credit card) THEN action(pay with credit card) AND add(paid)

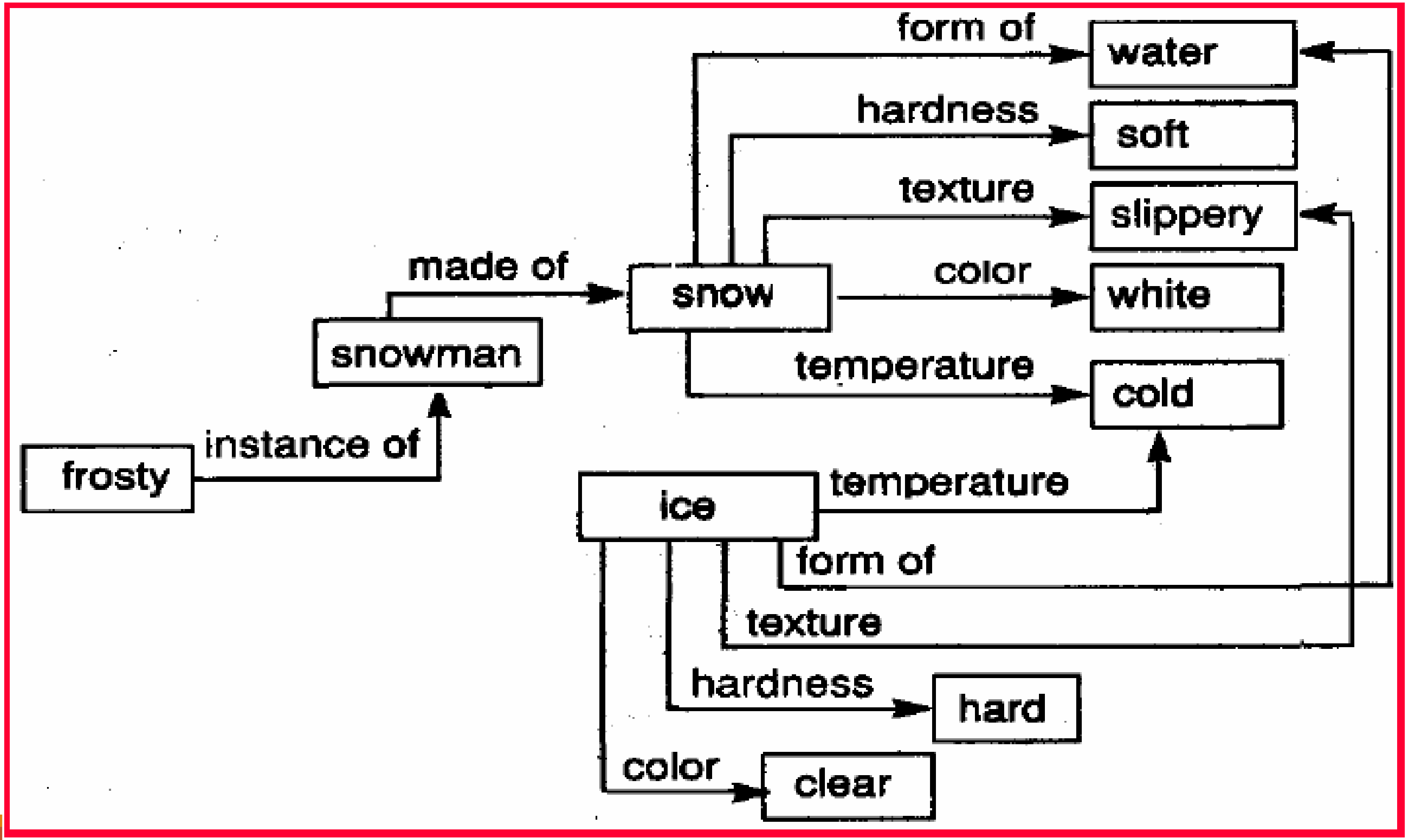
IF (on bus AND paid AND empty seat) THEN sit down

conditions and actions must be clearly defined

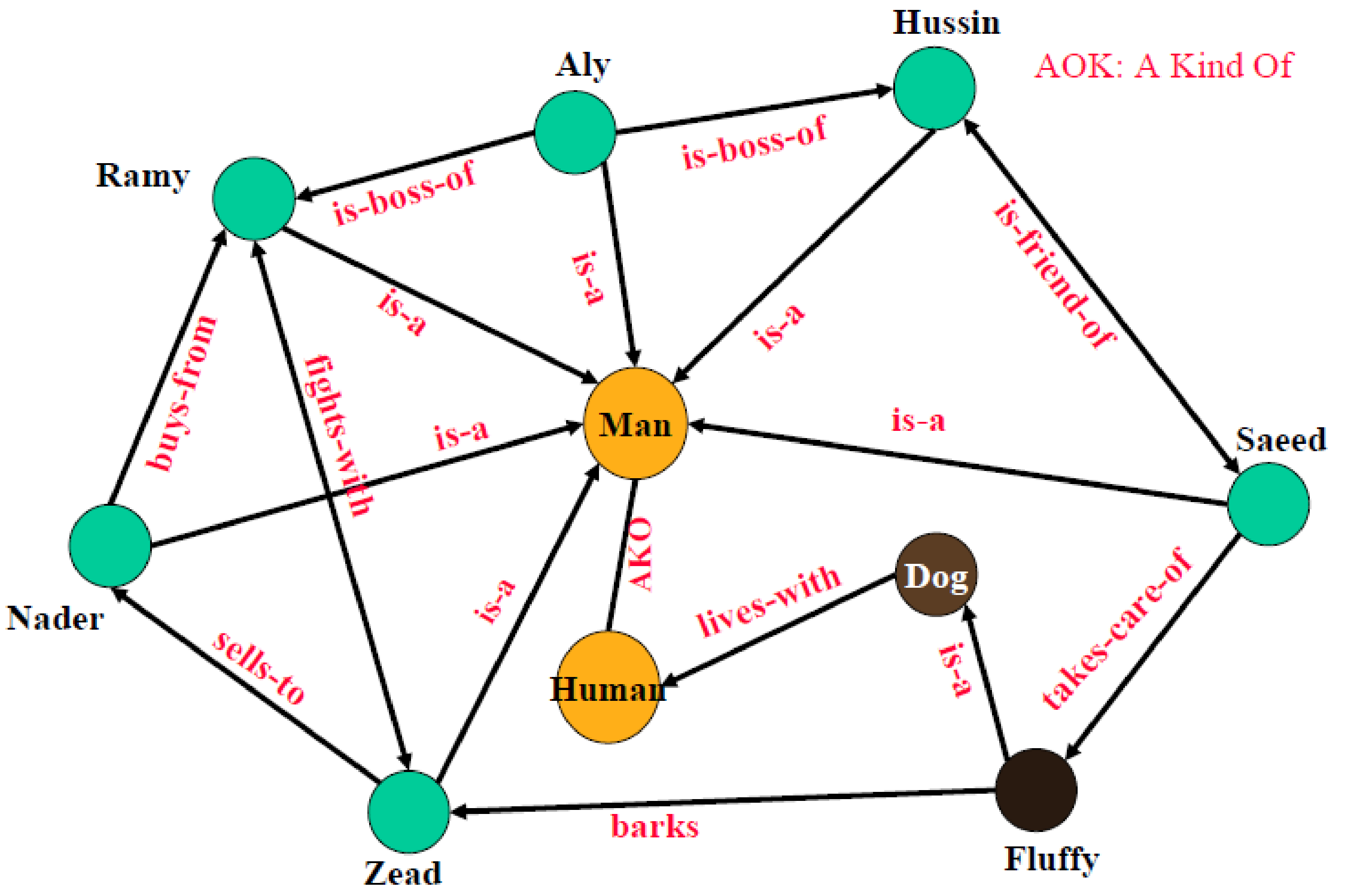
Semantic Nets

- Graphical representation for propositional information originally developed as a model for human memory
- It is labeled, directed graph where
  - nodes represent objects, concepts, or situations
    - labels indicate the name
    - nodes can be instances (individual objects) or classes (generic nodes)
  - links represent relationships
    - the relationships contain the structural information of the knowledge to be represented
    - the label indicates the type of the relationship

Semantic Net Examples

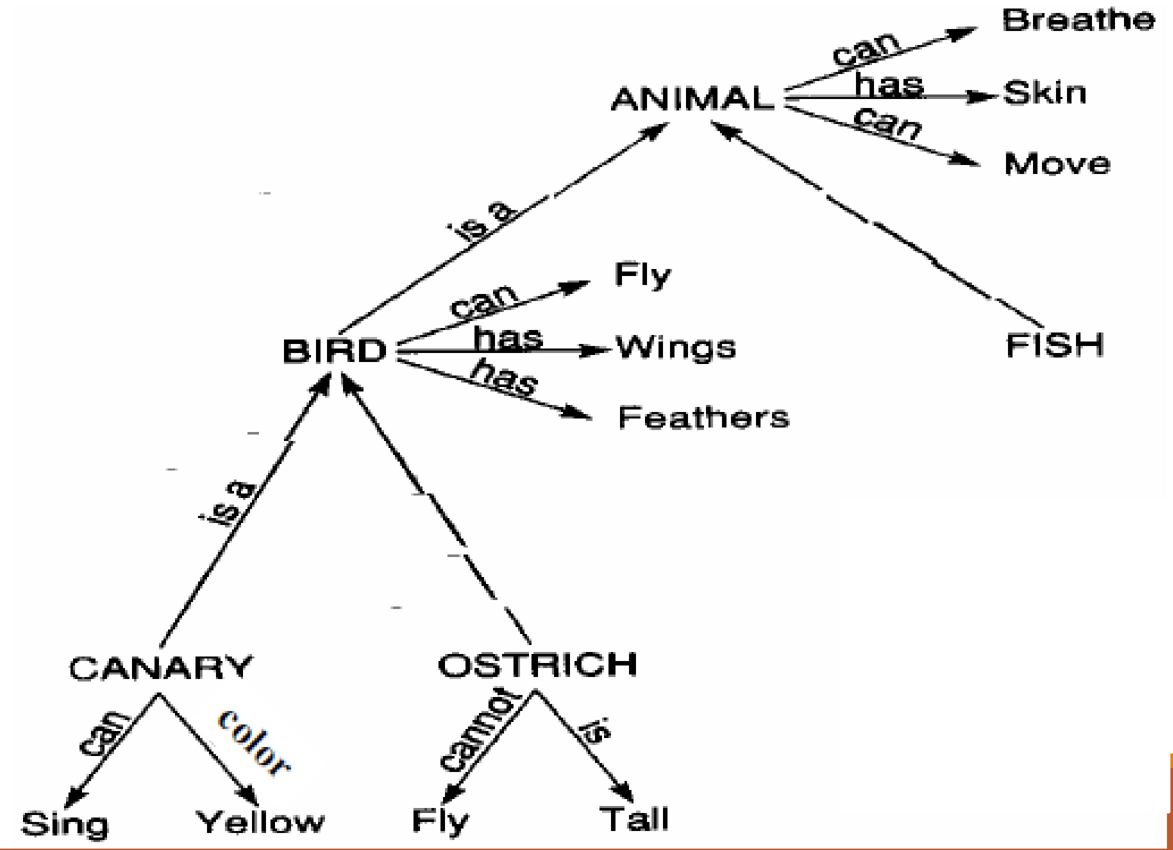


Semantic Net Examples



Semantic Net Examples

Semantic Net can be used for data retrieval, also it can provide for *deduction* using inheritance



Since a canary is a bird, it *inherits* the properties of birds (likewise, animals)
e.g., canary can fly, has skin, …

Frame Representations

Information retrieval when facing a new situation
- The information is stored in frames with slots
- Some of the slots trigger actions, causing new situations

Frames are templates
- Which are to be filled-in in a situation
- Filling them in causes an agent to
  - Undertake actions and retrieve other frames

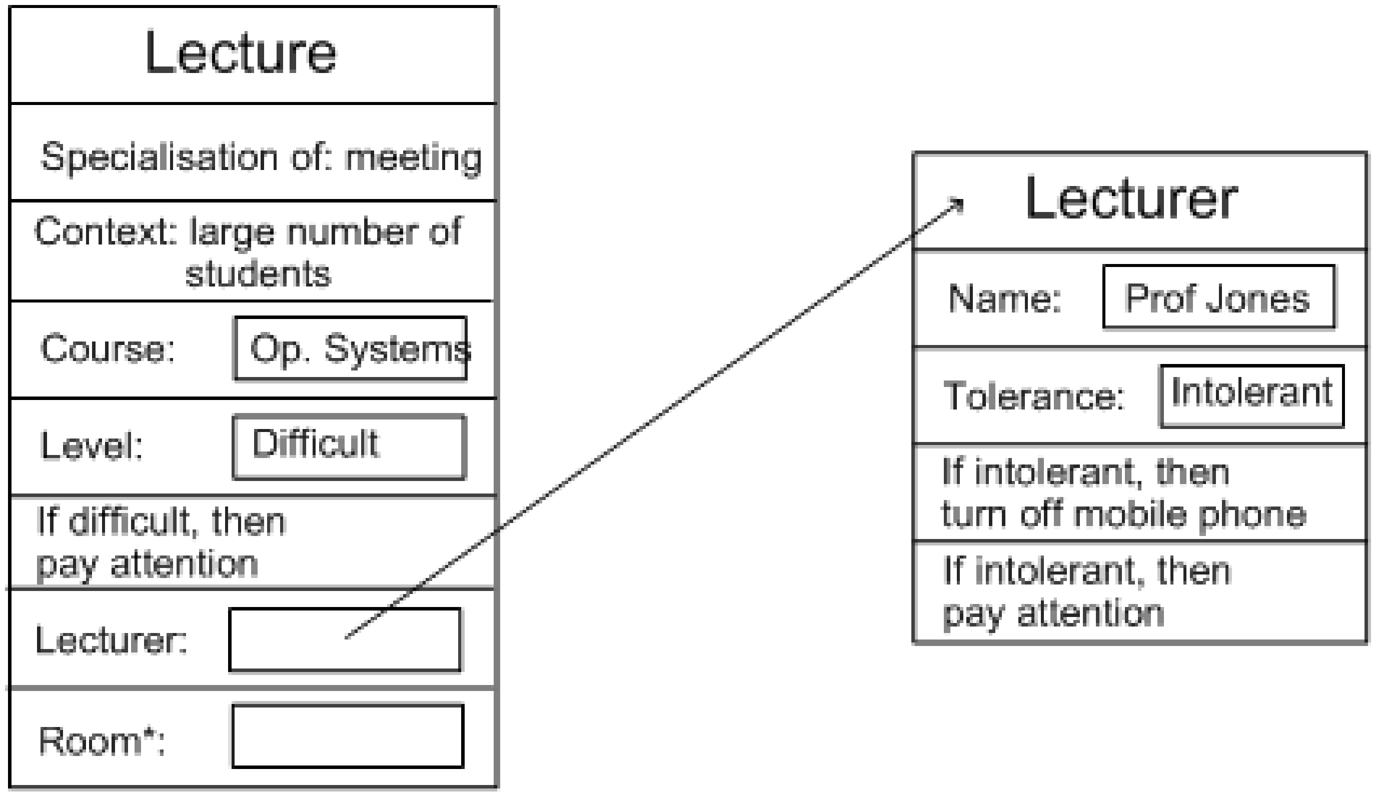
Frames are extensions of record datatype in databases
- Also very similar to objects in OOP

Flexibility in Frames

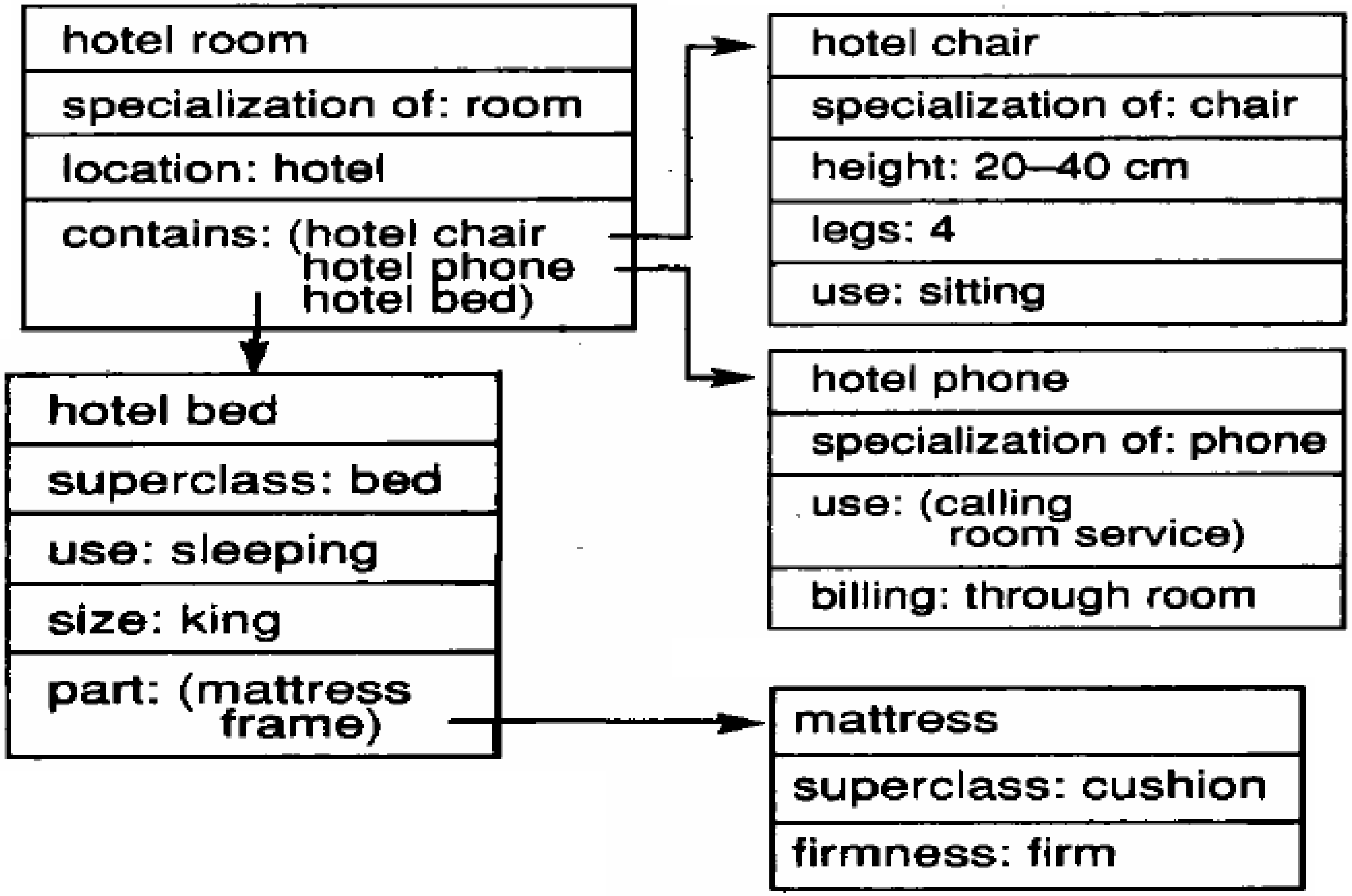
Slots in a frame can contain

- 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

Example Frame



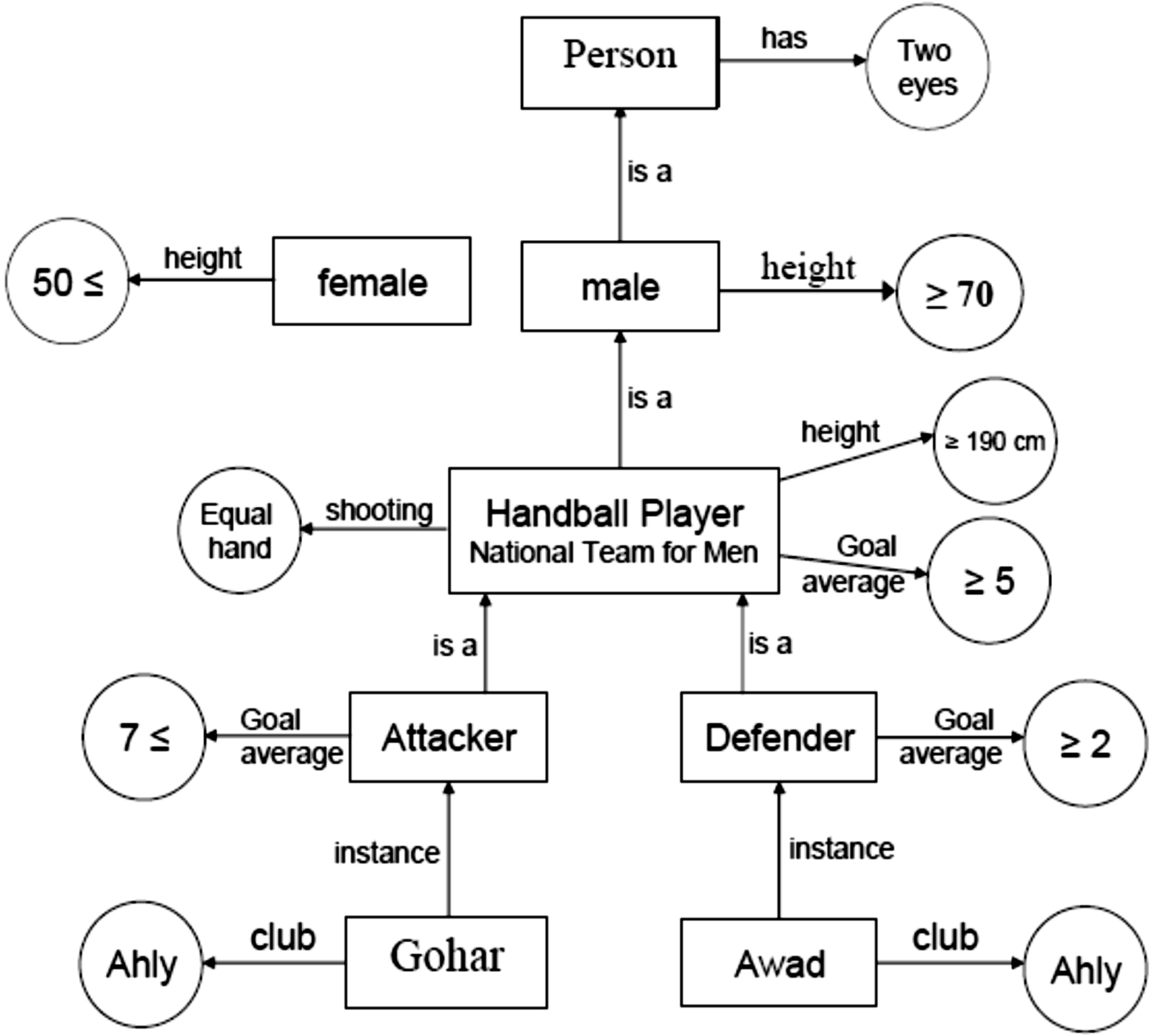
Example (2)



hotel room
specialization of: room
location: hotel
contains: (hotel chair hotel phone hotel bed)
hotel chair
specialization of: chair
height: 20–40 cm
legs: 4
use: sitting
hotel phone
specialization of: phone
use: (calling room service)
billing: through room
hotel bed
superclass: bed
use: sleeping
size: king
part: (mattress frame)
mattress
superclass: cushion
firmness: firm

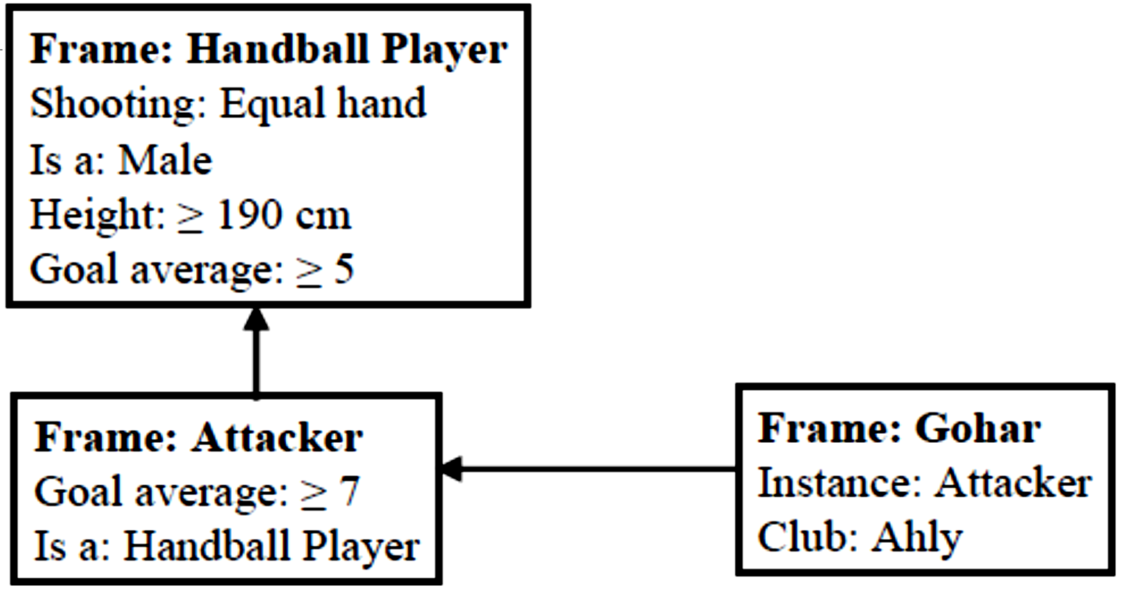
Example: National Handball Team for Men

1- Semantic Net



Example: National Handball Team for Men

2- Frames



3- Rules

Rule-1: IF a Handball player THEN his height is ≥ 190 cm AND he is a male AND his shooting is …………………….