MODULE 3

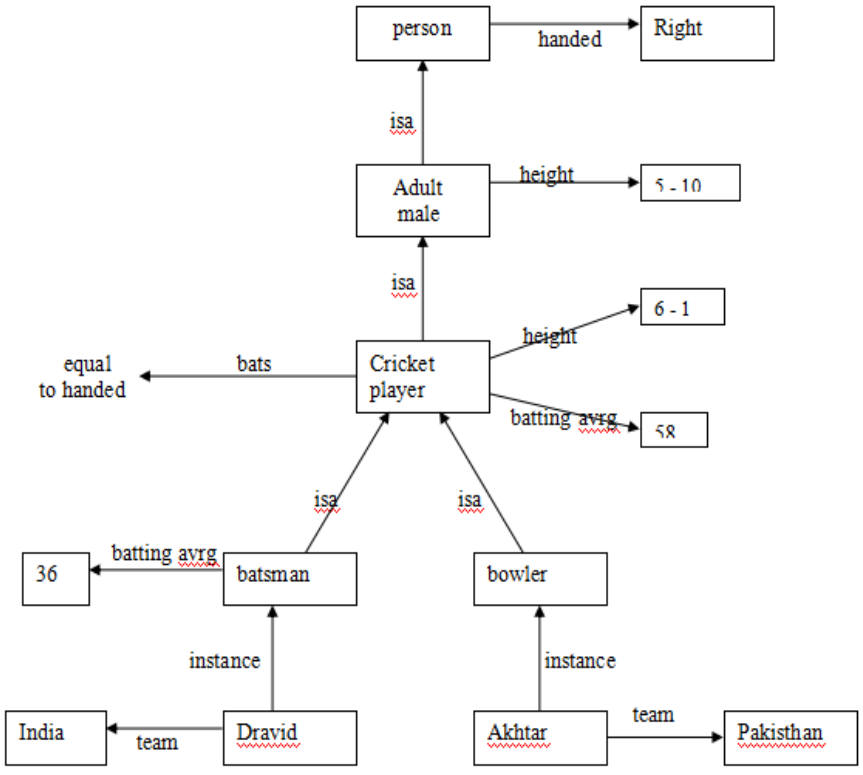
SYLLABUS

- AI representational schemes- Semantic nets, conceptual dependency, scripts, frames, introduction to agent based problem solving
- Machine learning-symbol based-a frame work for symbol based learning.

AI Representation Schemes

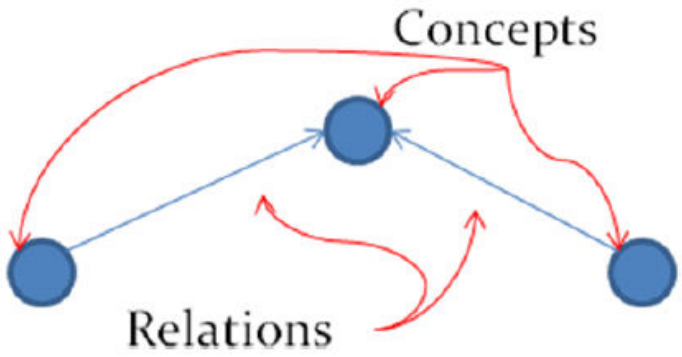
- Knowledge is very important in artificial intelligence systems.
- Knowledge is to be represented properly.
- Two of the knowledge representation schemes are
  1. semantic nets and
  2. frames

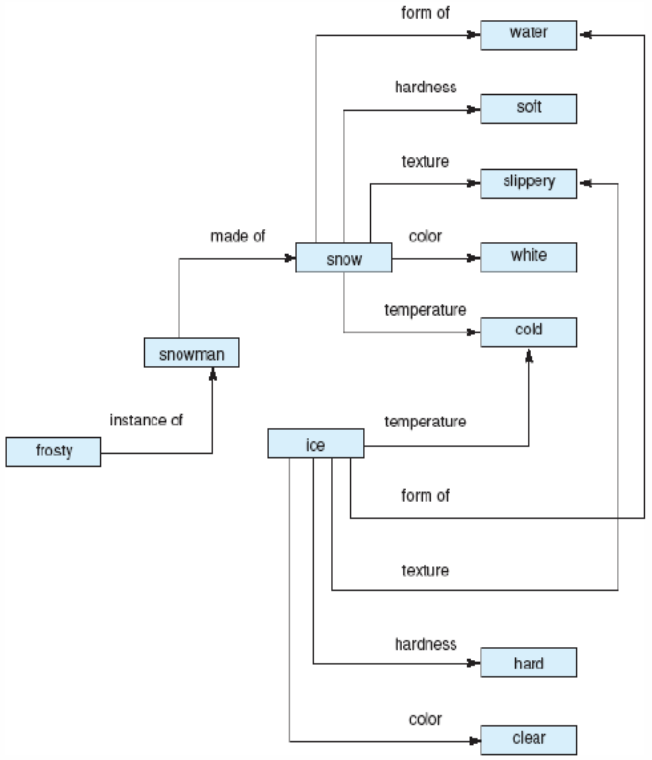
Semantic nets



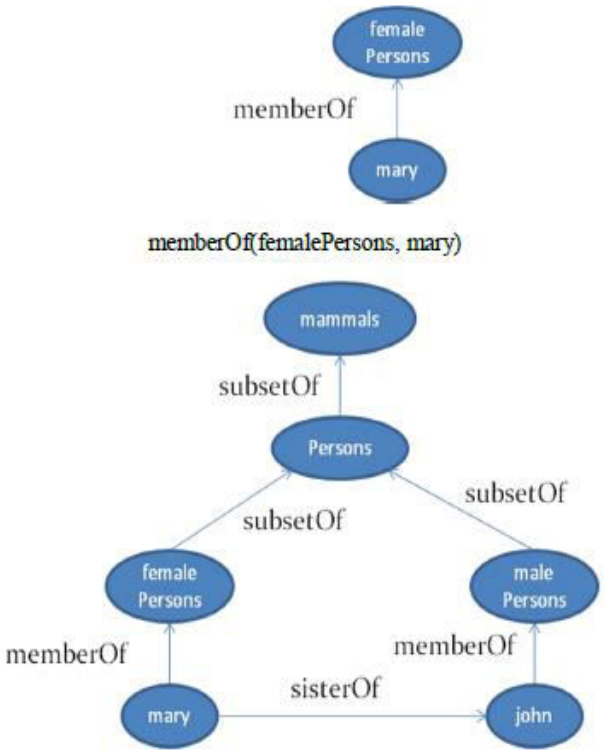
Semantic Networks

- Define objects in terms of their association with other objects
  - e.g. snow, white, snowman, ice, slippery.
- The nodes correspond to facts or concepts, and the arcs to relations or associations between concepts.
- Both nodes and links are generally labeled.
- Represent knowledge as a graph:
- Concepts at lower levels inherit characteristics from their parent concepts.



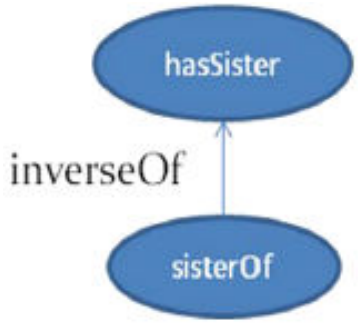


- This network can be used to answer a range of questions about snow, ice, and snowman.
- These inferences are made by following the links to related concepts.
- Semantic networks also implement inheritance;
- for example, frosty inherits all the properties of snowman.
- Well designed semantic networks are a form of logic.



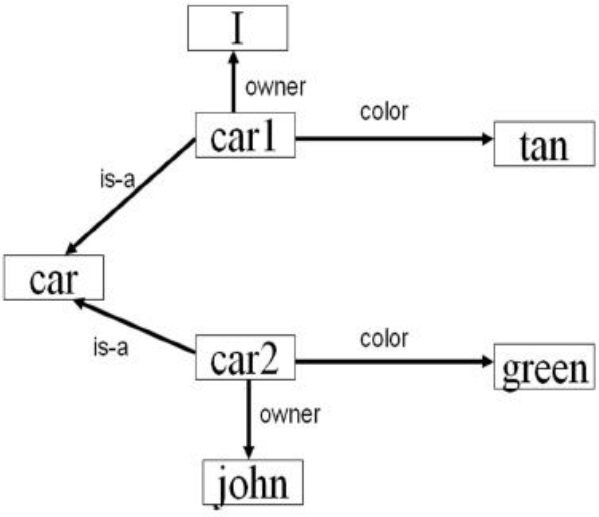
Inference Mechanism

- Inheritance
  - e.g. Persons by default have 2 legs.
  - How many legs does Mary have? John?
- Use of Inverse Links
  - e.g. hasSister(p, s) and
  - sisterOf(s, p)



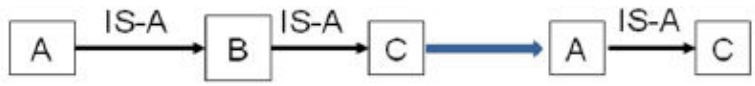
Examples of Semantic Net

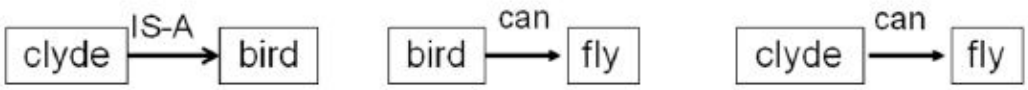
- My car is tan and John's car is green



Inference in a Semantic Net

- Inheritance
  - the is-a and instance-of representation provide a mechanism to implement this
  - Inheritance also provides a means of dealing with default reasoning





Semantic Networks Advantages

- Simple and transparent inference processes.
- Ability to assign default values for categories.
- Ability to include procedural attachment.

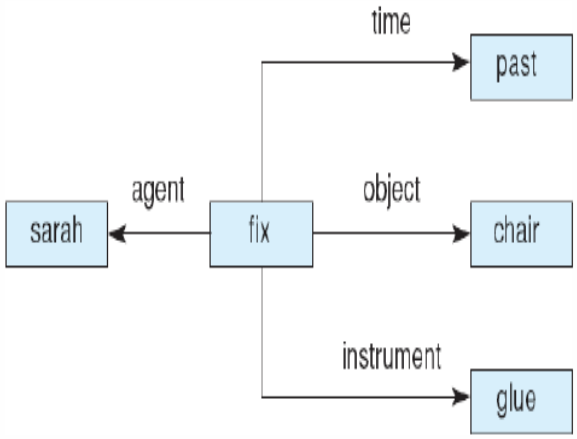
Semantic Networks - Disadvantages

- Simple query language may be too limiting to express complex queries.
- Does not represent full FirstOrderLogic(FOL) since it does not provide means to use negation, disjunction, and existential quantification.
- n-ary functions must be mapped onto binary functions

Standardization of Network Relationships

- Simmons addressed the need for standard relationships by focusing on the case structure of English verbs.
- In this verb-oriented approach, links define the roles played by nouns and noun phrases in the action of the sentence.
- Case relationships include agent, object, instrument, location, and time.
- A sentence is represented as a verb node, with various case links to nodes representing other participants in the action.
- This structure is called a case frame.

- In parsing a sentence, the program finds the verb and retrieves the case frame for that verb from its knowledge base.
- It then binds the values of the agent, object, etc., to the appropriate nodes in the case frame.
- Example: Sarah fixed the chair with glue



Conceptual Dependency (CD) theory

- CD theory was developed to represent the meaning of NL sentences.
  - It helps in drawing inferences
  - It is independent of the language
  - CD representation of a sentence is not built using words in the sentence rather built using conceptual primitives which give the intended meanings of words.
- CD provides structures and specific set of primitives from which representation can be built.

Conceptual category

- There are four primitive conceptual categories
  - ACT Actions {one of the CD primitives}
  - PP Objects {Picture Producers}
  - AA Modifiers of actions {Action Aiders}
  - PA Modifiers of PP's {Picture Aiders}

Primitive ACTs of CD theory

- ATRANS Transfer of an abstract relationship (i.e. give)
- PTRANS Transfer of the physical location of an object (e.g., go)
- PROPEL Application of physical force to an object (e.g. push)
- MOVE Movement of a body part by its owner (e.g. kick)
- GRASP Grasping of an object by an action (e.g. throw)
- INGEST Ingesting of an object by an animal (e.g. eat)

- EXPEL Expulsion of something from the body of an animal (e.g. cry)
- MTRANS Transfer of mental information (e.g. tell)
- MBUILD Building new information out of old (e.g decide)
- SPEAK Producing of sounds (e.g. say)
- ATTEND Focusing of a sense organ toward a stimulus (e.g. listen)

- Primitives of meaning
  1. Actions
  2. Objects
  3. Modifiers of actions
  4. Modifiers of objects
- Conceptual syntax rules
  - Built using these primitives
  - Constitute a grammar of meaningful semantic relationships.
- Conceptual dependency relationships
  - Are defined using the conceptual syntax rules
  - Can be used to construct an internal representation of an english sentence.

PP ⇔ ACT	indicates that an actor acts.
PP ⇔ PA	indicates that an object has a certain attribute.
ACT ←(O) PP	indicates the object of an action.
ACT ←(R) PP, PP	indicates the recipient and the donor of an object within an action.
ACT ←(D) PP, PP	indicates the direction of an object within an action.
ACT ←(1) ⇕	indicates the instrumental conceptualization for an action.
X ⇑ Y	indicates that conceptualization X caused conceptualization Y. When written with a C this form denotes that X COULD cause Y.
PP ⇐ PA2, PA1	indicates a state change of an object.
PP1 ← PP2	indicates that PP2 is either PART OF or the POSSESSOR OF PP1.

Fig. Basic Conceptual Dependency

Few conventions

- Arrows indicate directions of dependency
- Double arrow indicates two way link between actor and action.
  - O – for the object case relation
  - R – for the recipient case relation
  - P – for past tense
  - D – destination
  - Tense and mode are added.

p	past
f	future
t	transition
k	continuing
t_s	start transition
?	interrogative
t_f	finish transition
c	conditional
/	negative
nil	present
delta?	timeless

Example:

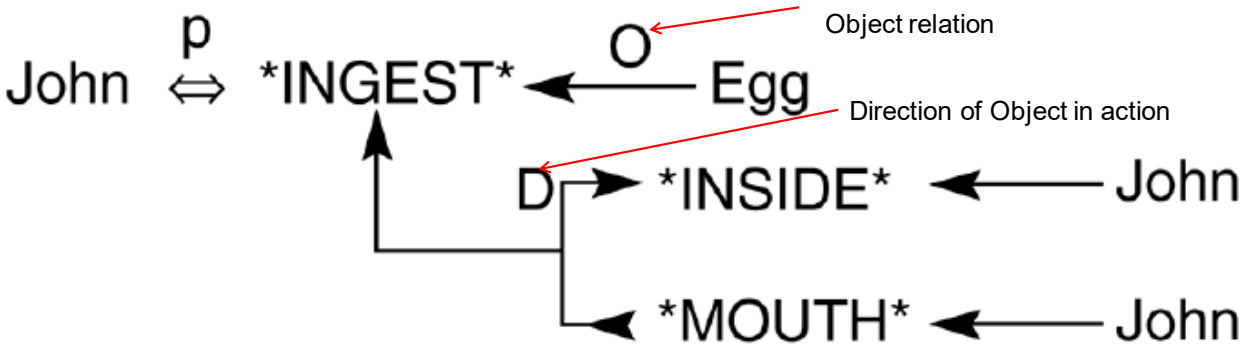
- "John throws the ball"

- "John threw the ball"

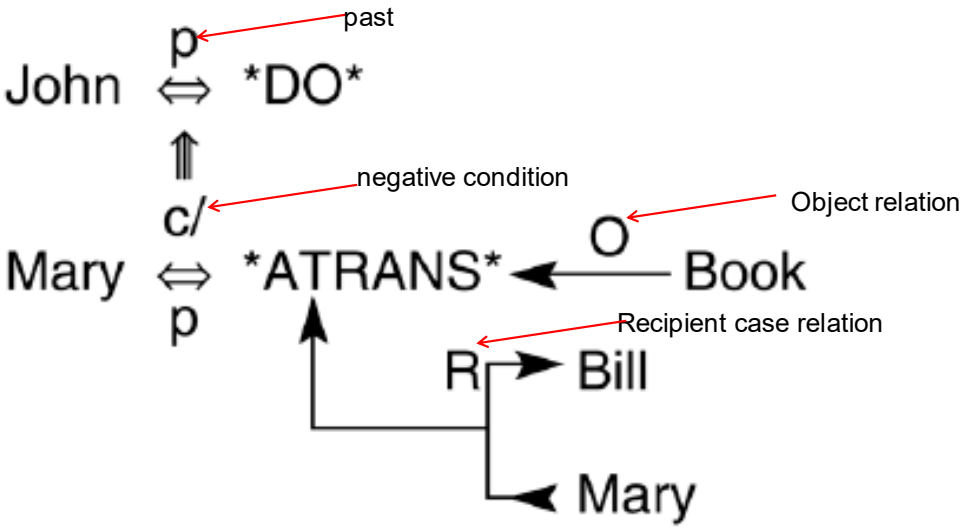
- The symbols have the following meanings:

←	indicates the direction of dependency
⇔	indicates the agent—verb relationship
p	indicates past tense
INGEST	is a primitive act of the theory
O	object relation
D	indicates the direction of the object in the action

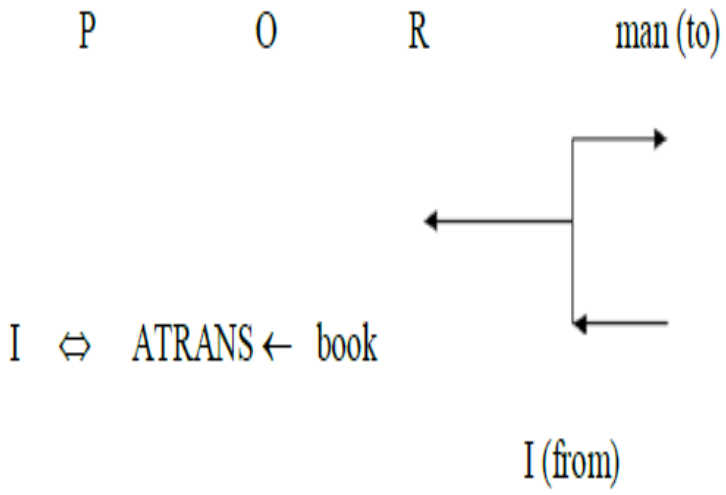
"John ate the egg"



"John prevented Mary from giving a book to Bill"



"I gave a book to the man" CD representation

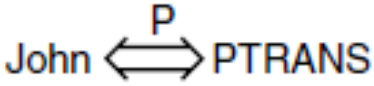


- It should be noted that this representation is same for different saying with same meaning. For example
  - I gave the man a book,
  - The man got book from me,
  - The book was given to man by me etc.

Rule 1

$$PP \Leftrightarrow ACT$$

- It describes the relationship between an actor and the event he or she causes.
  - This is a two-way dependency, since neither actor nor event can be considered primary.
  - The letter P in the dependency link indicates past tense.
- Example: John ran



Rule 2:

$$ACT \leftarrow PP$$

- It describes the relationship between a ACT and a PP (object) of ACT.
  - The direction of the arrow is toward the ACT since the context of the specific ACT determines the meaning of the object relation.
- Example: John pushed the bike

$$\text{John} \Leftrightarrow \text{PROPEL} \overset{O}{\leftarrow} \text{bike}$$

Rule 3

$$PP \leftrightarrow PP$$

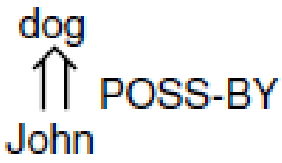
- It describes the relationship between two PP's, one of which belongs to the set defined by the other.
- Example: John is doctor

$$John \leftrightarrow Doctor$$

Rule 4:

$$PP \leftarrow PP$$

- It describes the relationship between two PP's, one of which provides a particular kind of information about the other.
  - The three most common types of information to be provided in this way are possession (shown as POSS-BY), location (shown as LOC), and physical containment (shown as CONT).
  - The direction of the arrow is again toward the concept being described.
- Example: John's dog



Rule 5

$$PP \Leftrightarrow PA$$

- It describes the relationship between a PP and a PA that is asserted to describe it.
  - PA represents states of PP such as height, health etc.
- Example: John is fat

$$\text{John} \Leftrightarrow \text{weight}\ (> 80)$$

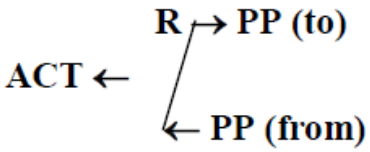
Rule 6:

$PP \leftarrow PA$

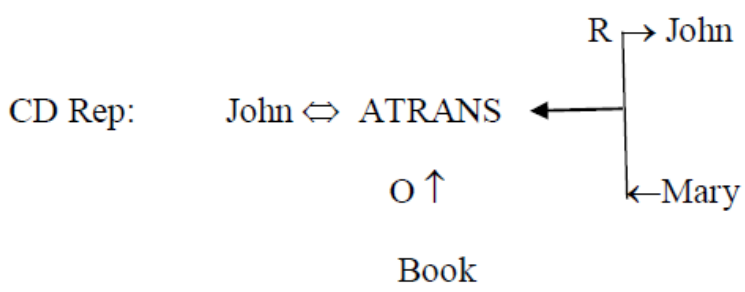
- It describes the relationship between a PP and an attribute that already has been predicated of it.
  - Direction is towards PP being described.
- Example: Smart John

$$John \leftarrow Smart$$

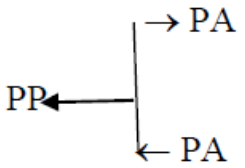
Rule 7



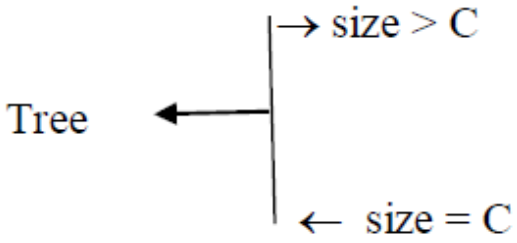
- It describes the relationship between an ACT and the source and the recipient of the ACT
- Example: John took the book from Mary



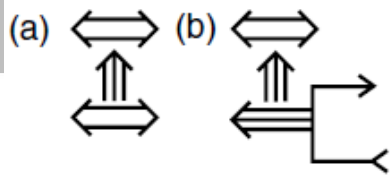
Rule 8:



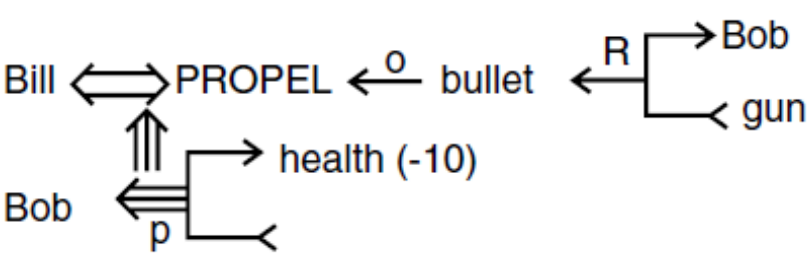
- It describes the relationship that describes the change in state.
- Example: Tree grows



Rule 9:



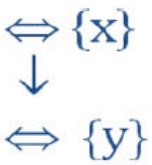
- It describes the relationship between one conceptualization and another that causes it.
- Here {x} causes {y} i.e., if x then y
- − Example: Bill shot Bob



{x} : Bill shot Bob

{y} : Bob's health is poor

Rule 10



- It describes relationship between one conceptulization with another that is happening at the time of first
- here {y} is happening while {x} is in progress
- Example: While going home i saw a snake

I am going home

↓

I saw a snake

Conceptual Dependency theory: Advantages

- Provides a formal theory of natural language semantics
- Reduces problems of ambiguity.
- Representation directly captures much of the natural language semantics
- Sentences with similar meaning will have similar representations (canonical form).

Disadvantages:

- No program exists that can reliably reduce sentences to canonical form.
- Primitives not sufficient to represent more subtle concepts.

Scripts

- A script is a structured representation describing a stereotyped sequence of events in a particular context.
  - i.e. if the system isn't told some detail of what's going on, it assumes the "default" information is true
- Scripts are used in natural language understanding systems to organize a knowledge base in terms of the situations that the system is to understand.

Why scripts?

- Because real-world events do follow stereotyped patterns. Human beings use previous experiences to understand verbal accounts; computers can use scripts instead.
- Because people, when relating events, do leave large amounts of assumed detail out of their accounts. People don't find it easy to converse with a system that can't fill in missing conversational detail.
- Scripts predict unobserved events.
- Scripts can build a coherent account from disjointed observations.

Applications

- This sort of knowledge representation has been used in intelligent front-ends, for systems whose users are not computer specialists.
- It has been employed in story-understanding and news-report-understanding systems.

Components of Scripts

- Script name (eg: restaurant)
- Entry conditions:
  - descriptors of the world that must be true for the script to be called.
  - Eg:- open restaurant, hungry customer
- Roles
  - actions that the individual participants perform.
  - Eg:- waiter takes orders, customer orders, eats

- Props
  - the "things" that support the content of the script.
  - Eg:- tables, waiters
- Scenes:
  - the script is broken into a sequence of scenes each of which presents a temporal aspect of the script.
  - Eg:- entering, ordering, eating, etc
- Results:
  - facts that are true once the script has terminated.
  - Eg:- the customer is full and poorer

- The elements of the script are represented using conceptual dependency relationships.
- Placed together in a frame-like structure, they represent a sequence of meanings, or an event sequence.
- The program reads a small story about restaurants and parses it into an internal conceptual dependency representation.

- The program binds the people and things mentioned in the story to the roles and props mentioned in the script.
- The result is an expanded representation of the story contents, using the script to fill in any missing information and default assumptions.
- The program then answers questions about the story by referring to the script.

Restaurant script

Script: Restaurant

Track: Coffee Shop

Roles: S=Customer

W=Waiter

C=Cook

M=Cashier

O=Owner

Props: Tables

F=Food

Check

Money & MENU

Entry conditions: S is hungry.
S has money.

Results: S has less money
O has more money
S is not hungry
S is pleased (optional)

Scene 1: Entering

S PTRANS S into restaurant
S ATTEND eyes to tables
S MBUILD where to sit
S PTRANS S to table
S MOVE S to sitting position

Scene 2: Ordering

(Menu on table) (W brings menu)
S PTRANS menu to S

(S asks for menu)
S MTRANS signal to W
W PTRANS W to table
S MTRANS 'need menu' to W
W PTRANS W to menu

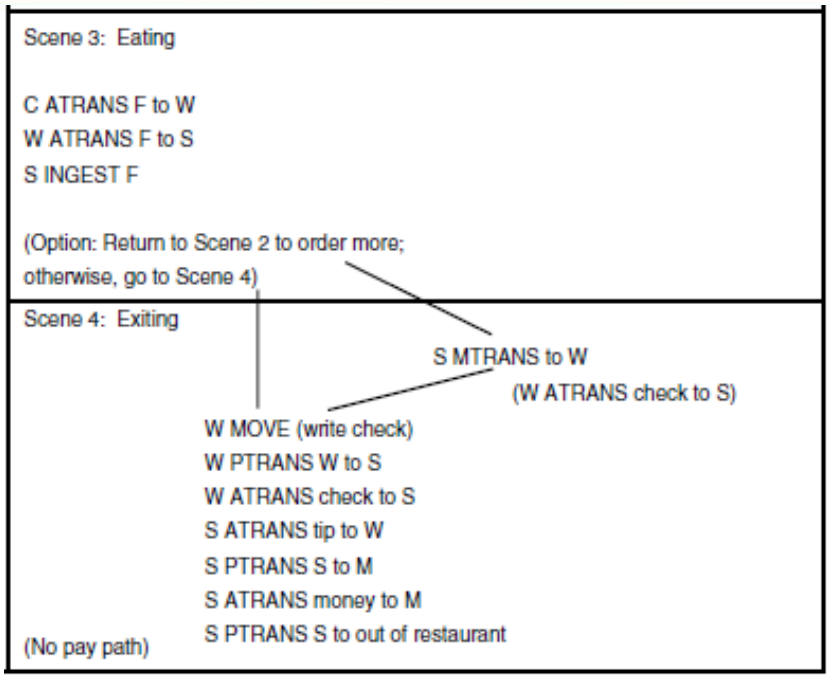
W PTRANS W to table
W ATRANS menu to S

S MTRANS food list to CP (S)
*S MBUILD choice of F
S MTRANS signal to W
W PTRANS W to table
S MTRANS 'I want F' to W

W PTRANS W to C
W MTRANS (ATRANS F) to C

C MTRANS 'no F' to W
W PTRANS W to S
W MTRANS 'no F' to S
(go back to *) or
(go to Scene 4 at no pay path)

C DO (prepare F script)
to Scene 3



Advantages

- Capable of predicting implicit events
- Single coherent interpretation may be build up from a collection of observations.

Disadvantage

- More specific (inflexible) and less general than frames.
- Not suitable to represent all kinds of knowledge.
- To deal with inflexibility, smaller modules called memory organization packets (MOP) can be used.
- MOPs represent knowledge as smaller components along with rules for dynamically combining them to form a schema that is appropriate to the current situation.

Frames

- Frames support the organization of knowledge into more complex units reflecting the organization of objects in the domain.
- Can be viewed as a static data structure with values attached.
- Each individual frame may be seen as a data structure, similar to the traditional "record", that contains information relevant to stereotyped entities.

- The slots in the frame contain information such as:
  - Frame identification information.
  - Relationship of this frame to other frames.
  - Descriptors of requirements for a frame.
  - Procedural information on use of the structure described – attaches procedural code to a slot
  - Frame default information
    - slot values taken as true when no evidence is found.
    - Eg:- chair has 4 legs
  - New instance information
    - slots may be left unspecified until needed.
    - Eg:- color of bedspread

Advantages

- Frames add power and clarity to semantic nets by allowing complex objects to be represented as a single frame.
- Frames provide an easier framework to organize information hierarchically than semantic nets.
- Frames allow for procedural attachment which runs a demon (piece of code) as a result of another action in the KB (this has also been done to some semantic nets).
- Both frames and semantic nets support class inheritance.

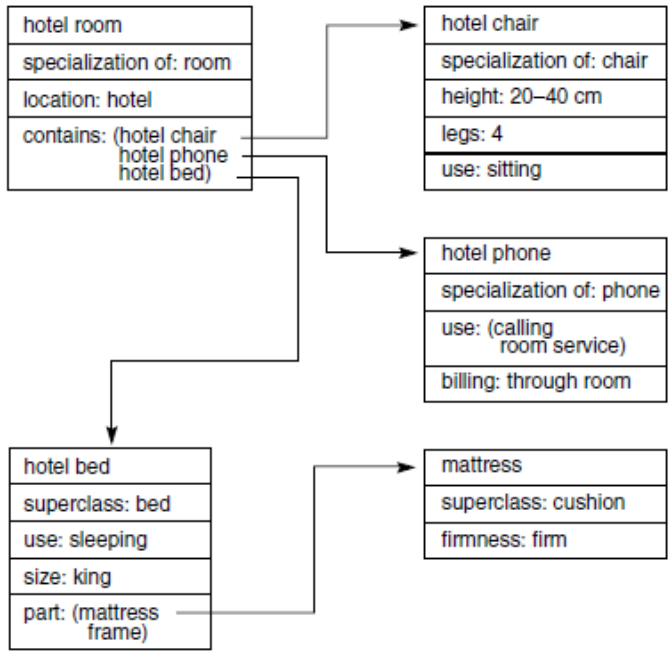


Figure 7.12 Part of a frame description of a hotel room. Specialization indicates a pointer to a superclass.

Agent-Oriented Problem Solving

- An agent is a problem solver that is:
- Situated (interacts with its environment)
- Autonomous (makes its own decisions without external intervention)
- Flexible (responds to stimuli from the environment, and initiates actions based on situation).
- Social (can interact appropriately with other agents or with humans).

Multi-Agent Problem Solvers

- The term multi-agent system refers to all types of software systems composed of multiple semi-autonomous components.
- A particular problem can be solved by a number of modules (agents) which cooperate by dividing and sharing the knowledge about the problem and its evolving solution.
- Multi-agent systems are ideal for representing problems that include many problem-solving methods, multiple viewpoints, and multiple entities.

- Agents interact to
  - cooperate towards achieving a common goal.
  - coordinate in organizing the problem-solving activity.
  - negotiate sub-problem constraints to improve performance
- Multi-agent systems form a “loosely coupled network of agents that work together” to achieve solutions to problems beyond the capabilities of any individual agent.

- Application domains where agent-based problem solving is appropriate include:
  1. Manufacturing.
  2. Automated Control.
  3. Telecommunications.
  4. Transportation Systems,
  5. Information Management.
  6. E-Commerce.
  7. Interactive Games and Theater.

Agent-Oriented Problem Solving Example: ROBOCUP

- RoboCup is an annual international robotics competition founded in 1997.
- “An international research and education initiative. “
- Provides “a standard problem where wide range of technologies can be integrated and examined.”
- Main domain: Soccer.
- Format: Two teams of robots.
- Robots compete in a soccer match on a standard platform.

- Agents and objects (an instantiation of a class in OOP) share some similarities but are quite different.
- However, we can use objects to create agents.

Objects in OOP vs. Agents

- Similarities: Objects (like agents) have
  - Systems with encapsulated states.
  - Certain methods are associated with the object's state.
  - Methods support interaction with the environment.
  - Different objects communicate by message passing.

- Differences
  - Objects do not usually control their own behaviour.
  - Agents can initiate their own actions. Object generally do not.
  - Objects do not have a social behaviour.
  - Agents do not invoke methods in one another.
  - Interacting agents usually have their own individual thread of control.
  - Agents can use more than just simple messages to communicate.
  - Objects are associated with their class. Agents can have multiple associations which may also change at any time.
  - Emergence can occur from groups of agents but not from objects.

Machine Learning

- AI systems grow from a minimal amount of knowledge by learning
- Learning definition by Herbert Simon (1983):

  Any change in a system that allows it to perform better the second time on repetition of the same task or on another task drawn from the same population

Machine learning issues

- Generalization from experience
- Induction
  - For large domains, a learner usually examines only a fraction; from this limited experience, the learner generalizes to the unseen instances of the domain.
- Inductive biases
  - Learners generalize heuristically, that is, they select those aspects of their experience that are most likely to prove effective in the future.
- Performance change:
  - improve or degrade

Machine Learning Categories

- 1. Symbol-based learning
  - Supervised learning
    - Inductive learning -- learning by examples
    - Inductive Bias
    - Explanation-based learning
  - Unsupervised learning
    - Clustering
    - Reinforcement learning: an agent is situated in an environment and receives feedback from that context
- 2. Neural/connectionist networks
- 3. Genetic/evolutionary learning

Factors for characterizing learning algorithms:

1. The data and goals of the learning task.
   - Describes the properties and quality of the training data.
   - The data may come from a teacher from the outside environment, or it may be generated by the program itself.
   - Data may be reliable or may contain noise.
   - It can be presented in a well-structured fashion or consist of unorganized data.
   - It may include both positive and negative examples or only positive examples.
   - Data may be readily available, the program may have to construct experiments, or perform some other form of data acquisition.

2. The representation of learned knowledge.
   - Concepts can be represented as
     - Logic expressions in predicate calculus
     - Structured representation such as frames or objects.
     - Decision trees
     - Rules
3. A set of operations.
   - Given a set of training instances, the learner must construct a generalization, heuristic rule, or plan that satisfies its goals.
   - ie manipulate representations
   - Generalizing or specializing symbolic expressions
   - Adjusting the weights in a neural network
   - Modifying the program's representations.

4. The concept space.
   - Search space: its representation, format
   - The learner searches this space to find the desired concept.
   - The complexity of this concept space is a primary measure of the difficulty of a learning problem.
5. Heuristic search
   - Learning programs must select a direction and order of search as well as the use of available training data and heuristics to search efficiently

A general model of the learning process