CS-208:Artificial Intelligence
Topic-18: Declarative Representation

Declarative Knowledge Representation

The methods of representing knowledge using predicate and propositional logics are often useful for representing simple facts. The major advantage of these methods is that they can be combined with powerful inference mechanism called resolution. But it is very difficult to represent complex knowledge using these representations.

Properties of Good Knowledge Representation System

Representational Adequacy	**The ability to represent all kinds of knowledge that are needed in a problem domain.**
Inferential Adequacy	**The ability to manipulate the representational structure in such a way to drive new structure corresponds to new knowledge.**
Inferential Efficiency	**The ability to incorporate in to knowledge structure the additional information that can be used to focus attention of the inference mechanism in a most promising direction.**
Acquisitional Efficiency	**The ability to acquire new information easily.**

Several techniques have been developed to meet these properties and they are classified in to two types

Declarative Methods	Procedural Methods
1. Each fact need only be stored once regardless of number of different ways it can be used. 2. It is easy to add new facts to the system without changing other facts or small procedure.	1. It is easy to represent knowledge of how to do things. 2. It is easy to represent the knowledge that does not fit in to many declarative scheme. 3. It is easy to represent heuristic knowledge of how to do things efficiently.

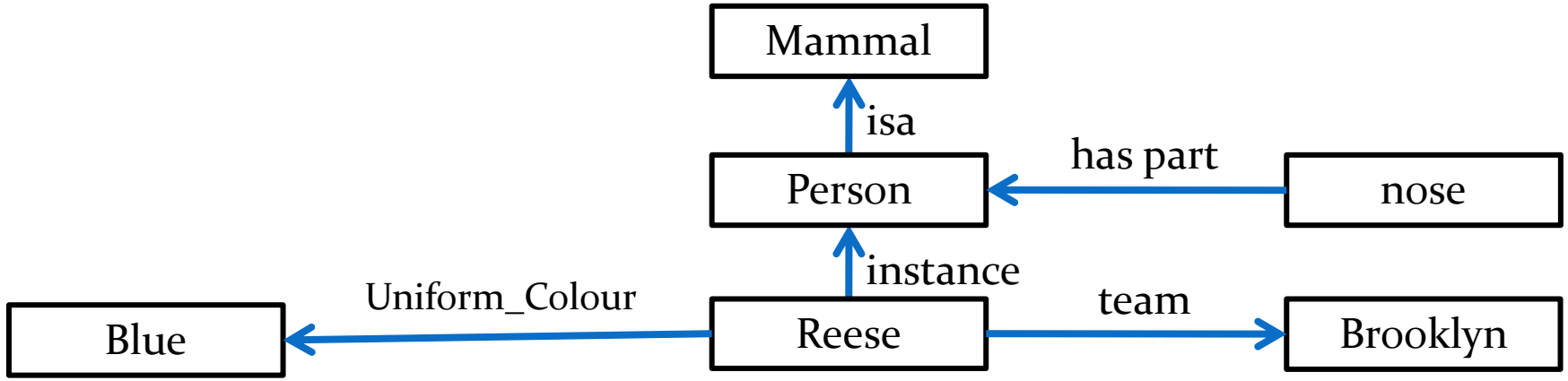
There are many declarative mechanism for representing knowledge and describe the following in detail:

1. Semantic Net
2. Frames
3. Conceptual Dependency
4. Scripts

Semantic Net

It was originally designed to represent the meanings of the English words. The idea behind the semantic net is that the meaning of a concept comes from the ways in which it is connected to other concepts.

A semantic network consist of nodes and labeled arcs, where each node represents a physical or mental concept that is the information is represented as a set of nodes and each arc represents the relationship between the nodes it connects. An example of semantic net representation is given below:



Intersection Search: Semantic nets were used to find the relationships among the objects by spreading the activation from each of the two nodes and seeing where the activation met. This process is called intersection search. Using this process we can find the connection between Brooklyn and blue.

Representing Non-binary Predicates

Semantic nets are a natural way to represent the relationship that would appear as ground instances of binary predicates in predicate logic.

isa(person, mammal)

instance(Reese, person)

team(Reese, Brooklyn)

Uniform_colour(Reese, blue)

Representing Unary Predicates

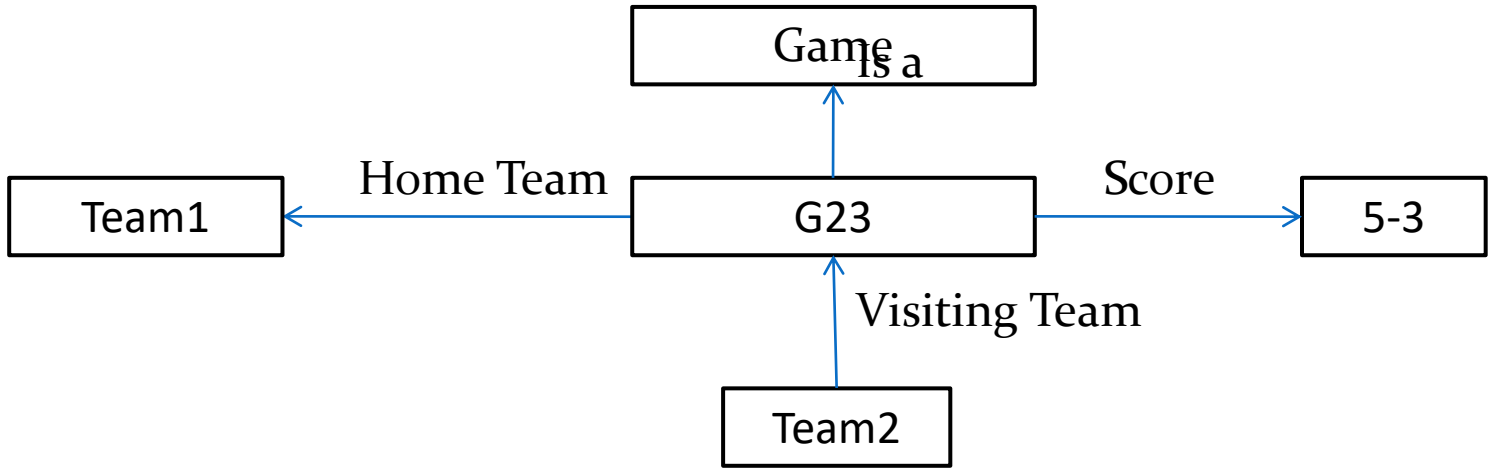
One place predicate can be converted into two place predicates

Example: Man(Marcus) can be converted in to instance(Man, Marcus)

Representing Three or More Place Predicates

Three or more place predicates can also be converted into a binary form by creating new objects representing the entire predicate statemnt.

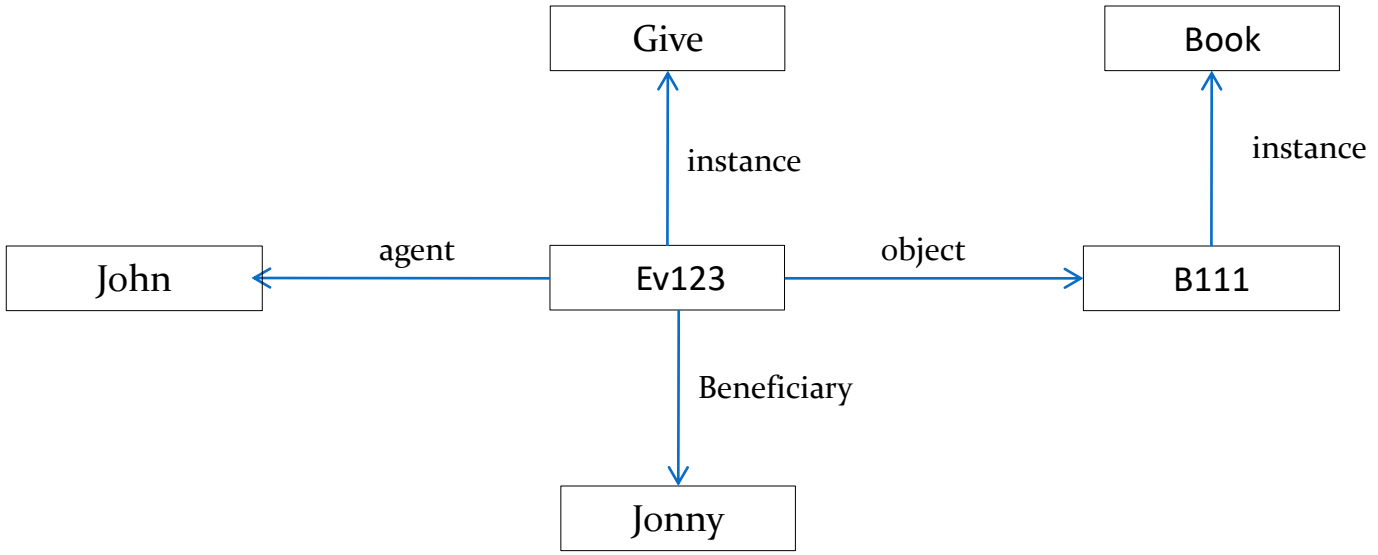
Example: Score(Team1, Team2, 5-3)



Semantic Net representing a sentence

We can represent a sentence in semantic net

Example: John gave the book to Johnny



Frames

Idea: there exist a great deal of evidence that people do not analyze new situations from the scratch and then build new knowledge to describe those situations. Instead they have available in memory a large collection of structures representing the previous experience with objects, locations situations and people.

To analyze new experience they invoke appropriate stored structure and then fill them in with details of current events.

A general mechanism designed for computer representation of such common knowledge is called Frame.

According to Minsky a frame is a static data structure used to represent well understood, stereotyped situations. Frame like structures seem to organize our own knowledge of the world. We adjust to ever new situations by calling up information structured by past experience. We then specially fashion or revise the details of these past experience to represent the individual differences for the new situation.

Frame Description : Frames represent knowledge as structured objects. A frame is a collection labeled slots with their values that together describe a sterotyped object, act or events in the world.

The internal designed structure make them useful in specific kinds of problem solving task.

Name of the Frame	
Slot	**Slot's Value**

Representing Knowledge Using Frames

A frame describe a class of objects and it consists of a collection of slots that describe aspects of the objects. These slots are filled by frame describe by particular object.

Associated with each slot may be a set of conditions that must be met any fills for it.

Each slot is filled with a default value sothat in the absence of a specific information to the contrary, things can be assumed to be as they usually are.

Procedural information may be associated with particular slots

Conceptual Dependency (CD)

Conceptual Dependency: It is a theory about how to represent the meaning of natural language sentences in a way that

i. Facilitate the drawing of inference from sentences
ii. independent of the language in which the sentences were originally stated.

Representing Knowledge using CD

The CD representation is built out of conceptual primitives that can be combined to form the meanings of words in any particular language.

CD provides both a structure and a specific set of primitives out of which representation of any particular piece of information can be constructed.

Primitive Acts

ATRANS -- Transfer of an abstract relationship. E*xample: give*.

PTRANS -- Transfer of the physical location of an object. E*xample: go*.

PROPEL -- Application of a physical force to an object. E*xample: push*.

MTRANS -- Transfer of mental information. E*xample: tell*.

MBUILD-- Construct new information from old. E*xample: decide*.

SPEAK -- Utter a sound. E*xample say*.

ATTEND -- Focus a sense on a stimulus. E*xample: listen, watch*.

MOVE -- Movement of a body part by owner. E*xample: punch, kick*.

GRASP -- Actor grasping an object. E*xample: clutch*.

INGEST -- Actor ingesting an object. E*xample: eat*.

EXPEL-- Expulsion of something out of body of an animal in to the world : cry

Six primitive conceptual categories to provide building blocks

PP -- Real world objects.

ACT -- Real world actions.

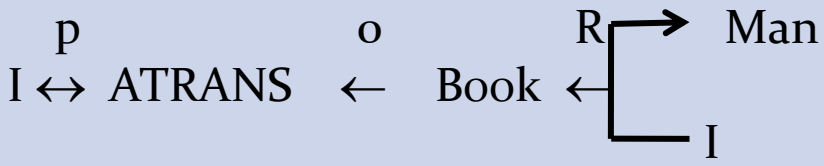
PA -- Attributes of objects.

AA -- Attributes of actions.

T -- Times.

LOC-- Locations.

Example: I gave the man *a book*



→ Arrows indicate the direction of dependency.

↔ Double arrows indicate *two-way* links between the actor (PP) and action (ACT).

p indicate past tense

o object case relation

R recipient-donor relation

Conceptual Dependency not only provide a structure through which knowledge can be represented but also a specific set of building block from which representation can be built.

Second set of building block is set of allowable dependencies among conceptualization described in a sentence. There are four primitive conceptual categories

ACT's– Actions

PP's – Objects (picture producer)

AA's – Modifier of actions (action aider)

PA's – Modifier of PP's (Picture aider)

Example: *John ran* p

Rule1: PP ⟺ ACT's John ⟺ PTRANS

Scripts

Definition A script is a structured representation describing a stereotyped sequence of events in a particular context. Scripts consist of a set slots. Associated with each slot

A script is composed of following components:

- Entry conditions that must be true for the script to be called.
- Results or facts those are true once the script has terminated.
- Props or the “things” that make up the content of the script.
- Roles are the actions that the individual participants perform.
- Scenes which present temporal aspects of the script.

Scripts: an example

Script Restaurant

Props
- •Tables
- •Menu
- •F = Food
- •Check
- •Money

Roles
- •P = Customer
- •O = Waiter
- •V = Cook
- •K = Cashier
- •S = Owner

Entry conditions
- •P is hungry
- •P has money

Results
- •P has less money
- •P is not hungry
- •P is pleased (optional)
- •S has more money

Scene 1: Entering

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

Scene 2: Ordering

(Menu on table)
O brings menu)
P PTRANS menu to P

(S asks for menu)
S MTRANS signal to O
O PTRANS O to table
P MTRANS "need menu" to O
O PTRANS O to menu

O PTRANS O to table
O ATRANS menu to P

P MTRANS food list to P
* P MBUILD choice of F
P MTRANS signal to O
O PTRANS O to table
P MTRANS 'I want F' to O

O PTRANS O to V
O MTRANS (ATRANS F) to V

V MTRANS 'no F' to O
O PTRANS O to P
O MTRANS 'no F' to P
(go back to *) or
(go to Scene 4 at no pay path)

V DO (prepare F script)
to Scene 3

Scene 3: Eating

V ATRANS F to O
O ATRANS F to P
P INGEST F

Option: Return to Scene 2 to order more; otherwise, go to Scene 4

Scene 4: Exiting

P MTRANS to O
(O ATRANS check to P)

O MOVE write check
O PTRANS O to P
O ATRANS check to P
P ATRANS tip to O
P PTRANS P to K
P ATRANS money to K
P PTRANS P to out of restaurant

No pay path

Schank un Abelson, 1977