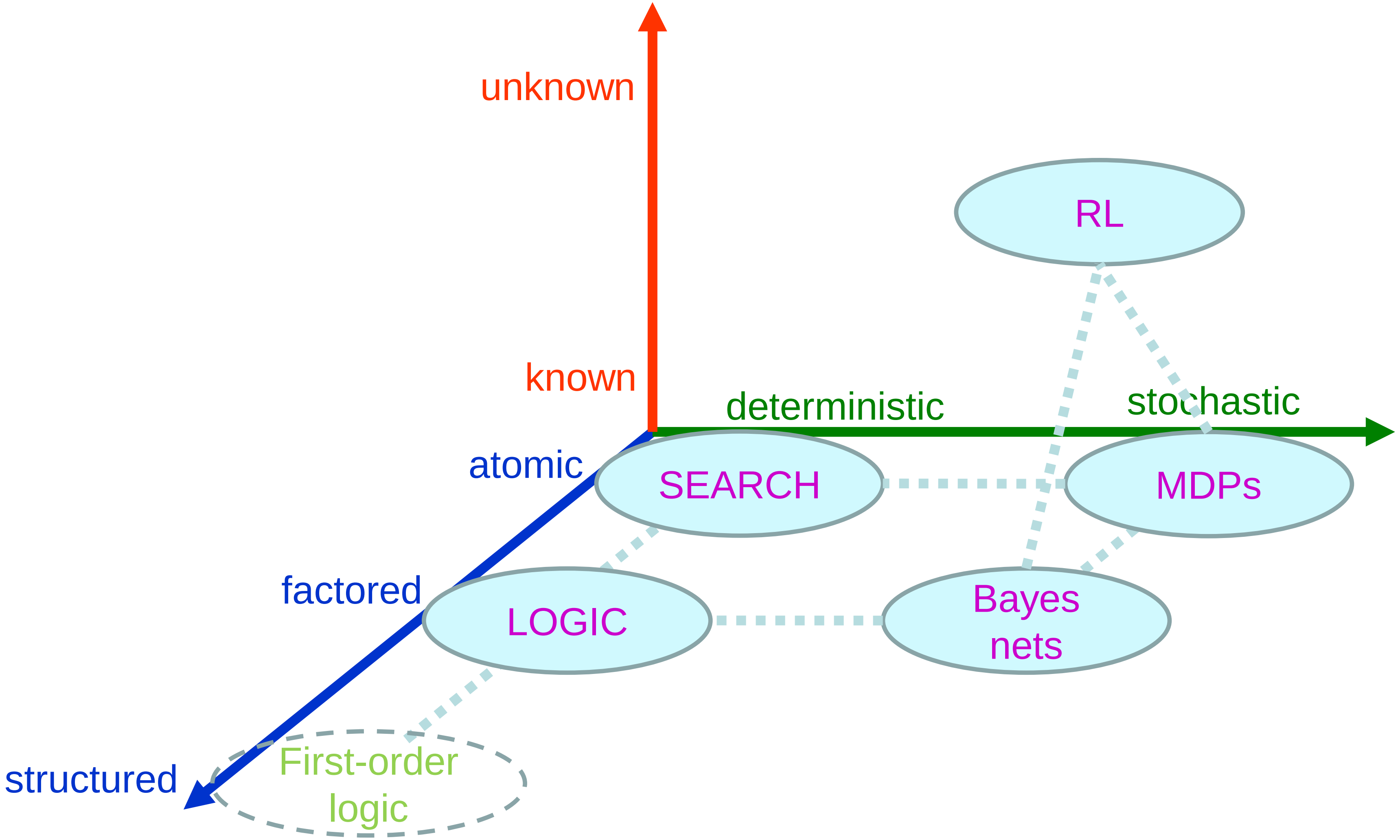
Artificial Intelligence

Introduction to Logic



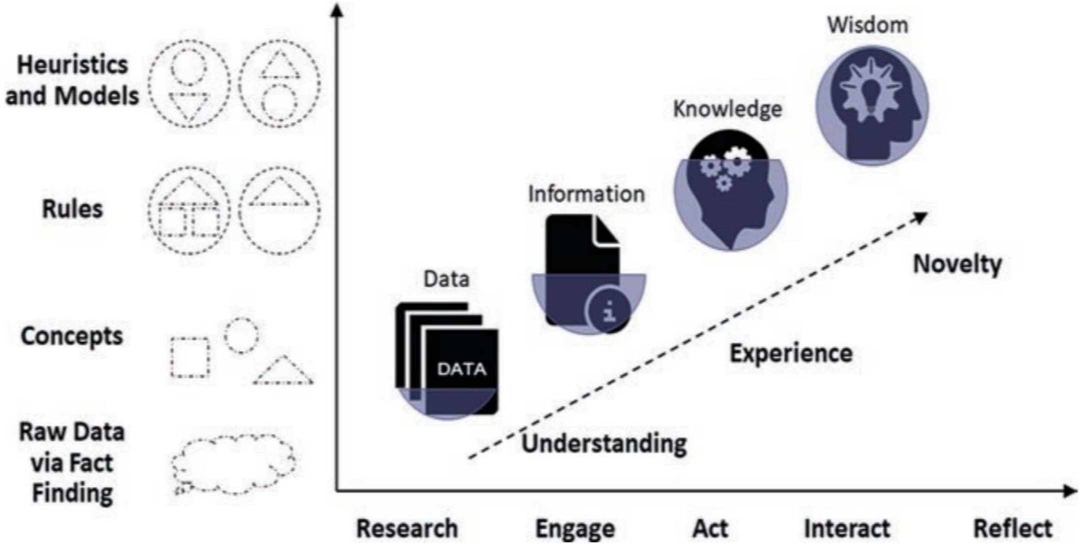
Outline

1. Knowledge
2. Data to Intelligence
3. Knowledge Base
4. Knowledge Base Agent
5. Knowledge Representation
   - Logical
   - Semantic Net
   - Frames
   - Production Rules

Knowledge

- The fact or condition of knowing something with familiarity gained through experience or association.
- Knowing something via seeing, hearing, touching, feeling, and tasting .
    - Acquaintance with or understanding of a science, art, or technique
    - The fact or condition of being aware of something
    - The range of one's information or understanding
    - The fact or condition of having information or of being learned
    - The sum of what is known : the body of truth, information, and principles acquired by mankind

Convergence from Data to Intelligence



Data is viewed as collection of *disconnected facts.* **:** Example : It is raining.

Information emerges when *relationships among facts* are established and understood; Provides answers to *"who", "what", "where", and "when".* **:** Example : The temperature dropped 15 degrees and then it started raining.

Knowledge emerges when *relationships among patterns* are identified and understood; Provides answers as *"how"* . **:** Example : If the humidity is very high and the temperature drops substantially, then atmospheres is unlikely to hold the moisture, so it rains.

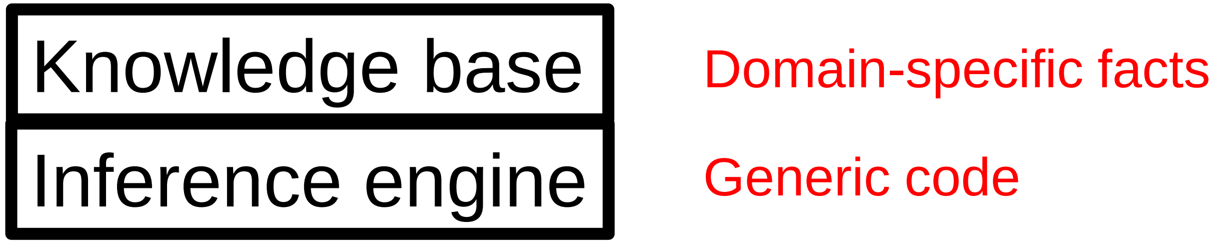
Wisdom is the pinnacle of understanding, uncovers the *principles of relationships that describe patterns.* Provides answers as *"why"* . **:** Example : Encompasses understanding of all the interactions that happen between raining, evaporation, air currents, temperature gradients and changes.

Knowledge Base

- Knowledge base = set of sentences in a formal language
- Declarative approach to building an agent (or other system):
  - ***Tell*** it what it needs to know (or have it ***Learn*** the knowledge)
  - Then it can ***Ask*** itself what to do—answers should follow from the KB
- For Example:
  - **Tell**: Father of Dipak is Ramesh
  - **Tell**: Jyoti is Dipak sister
  - **Tell**: Dipak father is the same as Jyoti sister father
  - **Ask**: Who is Jyoti father?

Knowledge, contd.

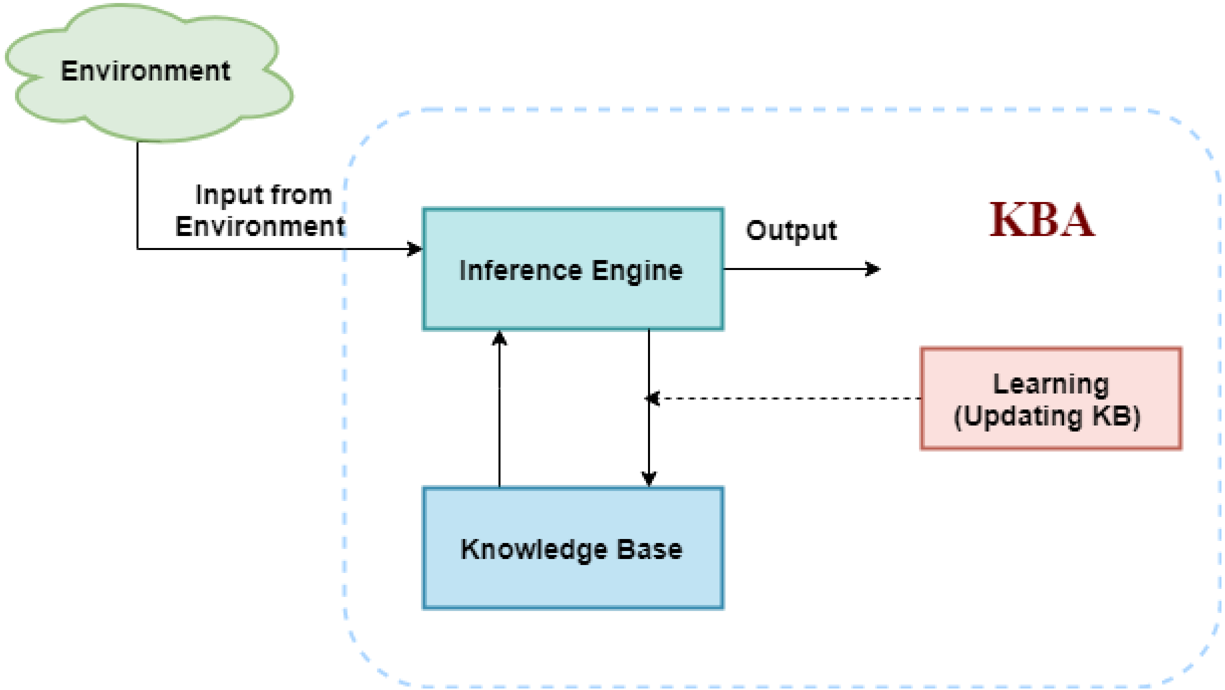
- Knowledge base = set of sentences in a formal language
- Declarative approach to building an agent (or other system):
  - ***Tell*** it what it needs to know (or have it ***Learn*** the knowledge)
  - Then it can ***Ask*** itself what to do—answers should follow from the KB
- Agents can be viewed at the ***knowledge level***
  i.e., what they ***know***, regardless of how implemented
- A single inference algorithm can answer any answerable question



Knowledge base — Domain-specific facts

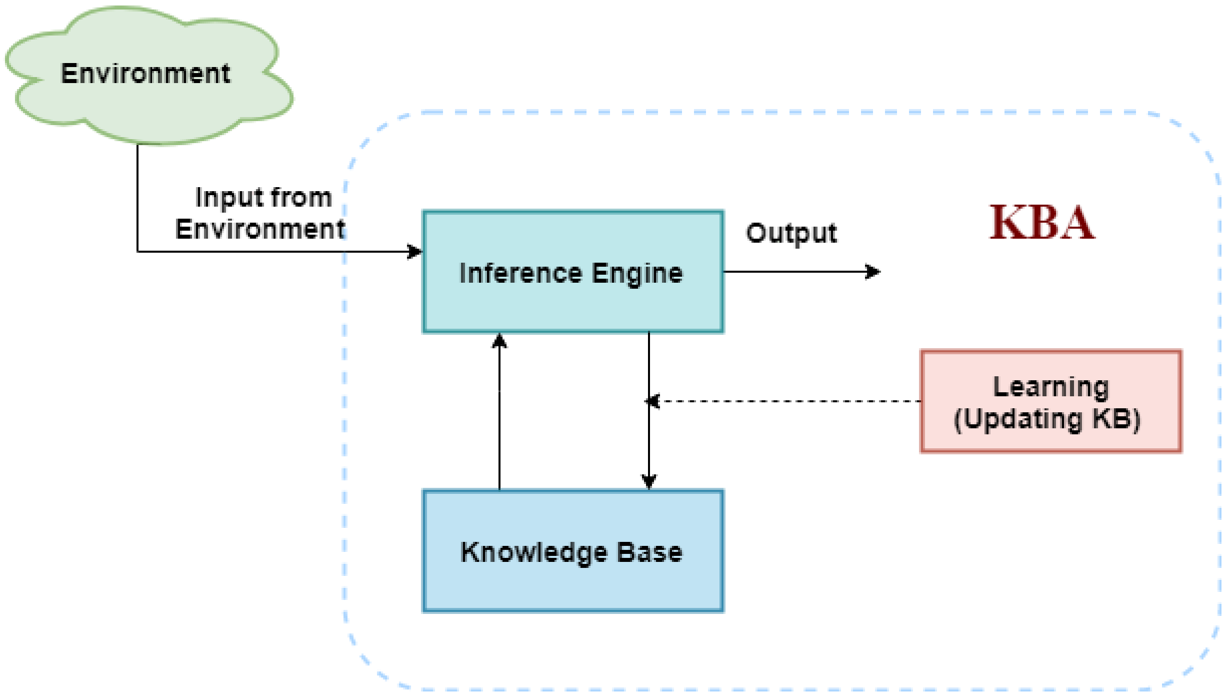
Inference engine — Generic code

Knowledge Base Agent in AI



- An intelligent agent needs **knowledge** about the real world for **taking decisions** and **reasoning** to act efficiently.
- Knowledge-based agents are those agents who have the capability of
  - Maintaining an internal state of knowledge
  - Reason over that knowledge
  - Update their knowledge after observations
  - Take actions.
- These agents can represent the world with some formal representation and act intelligently.

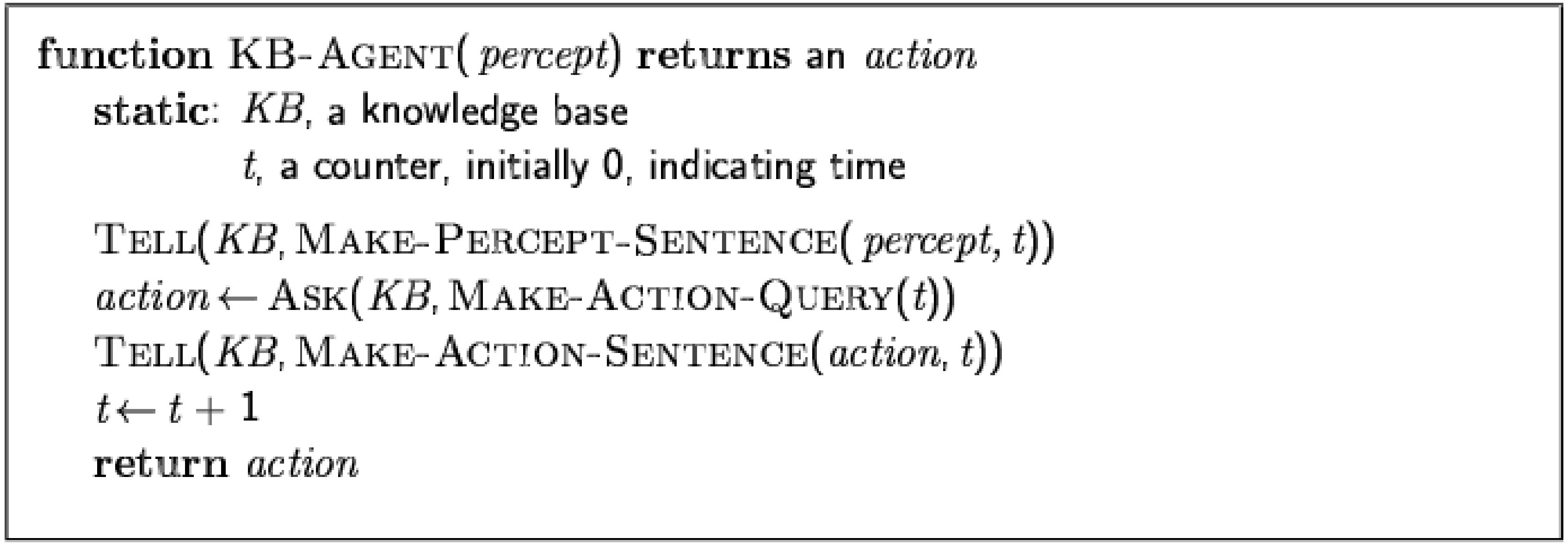
Knowledge Base Agent in AI



- **Knowledge-base:**
  - It is a collection of sentences, expressed in a language which is called a knowledge representation language.
  - The Knowledge-base of KBA stores fact about the world.
- **Inference Engine:**
  - Deriving new sentences from old.
  - Allows us to add a new sentence to the knowledge base.
  - Inference system applies logical rules to the KB to deduce new information.
  - Inference system generates new facts so that an agent can update the KB.

A simple knowledge-based agent

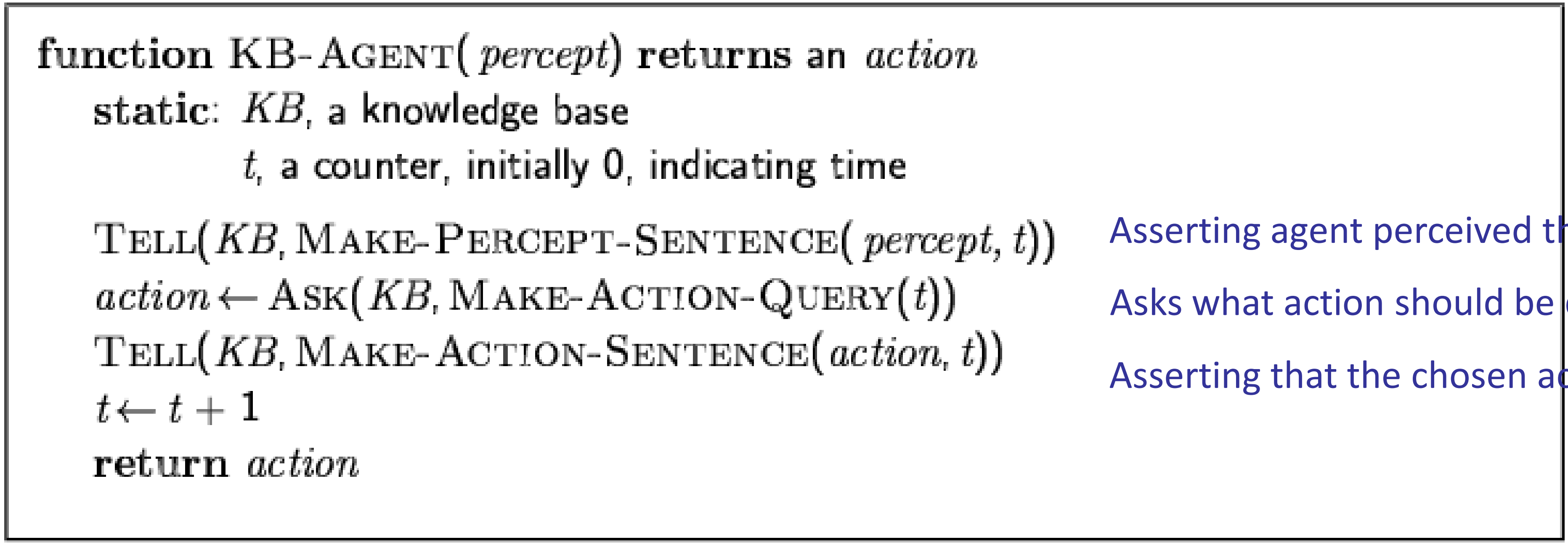
- The agent must be able to:
  - Represent states, actions, etc.
  - Incorporate new percepts
  - Update internal representations of the world
  - Deduce hidden properties of the world
  - Deduce appropriate actions



```
function KB-AGENT(percept) returns an action
    static: KB, a knowledge base
            t, a counter, initially 0, indicating time

    TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
    action ← ASK(KB, MAKE-ACTION-QUERY(t))
    TELL(KB, MAKE-ACTION-SENTENCE(action, t))
    t ← t + 1
    return action
```

A simple knowledge-based agent



```
function KB-AGENT(percept) returns an action
    static: KB, a knowledge base
            t, a counter, initially 0, indicating time

    TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
    action ← ASK(KB, MAKE-ACTION-QUERY(t))
    TELL(KB, MAKE-ACTION-SENTENCE(action, t))
    $t \leftarrow t + 1$
    return action
```

Asserting agent perceived the given percept at the given time

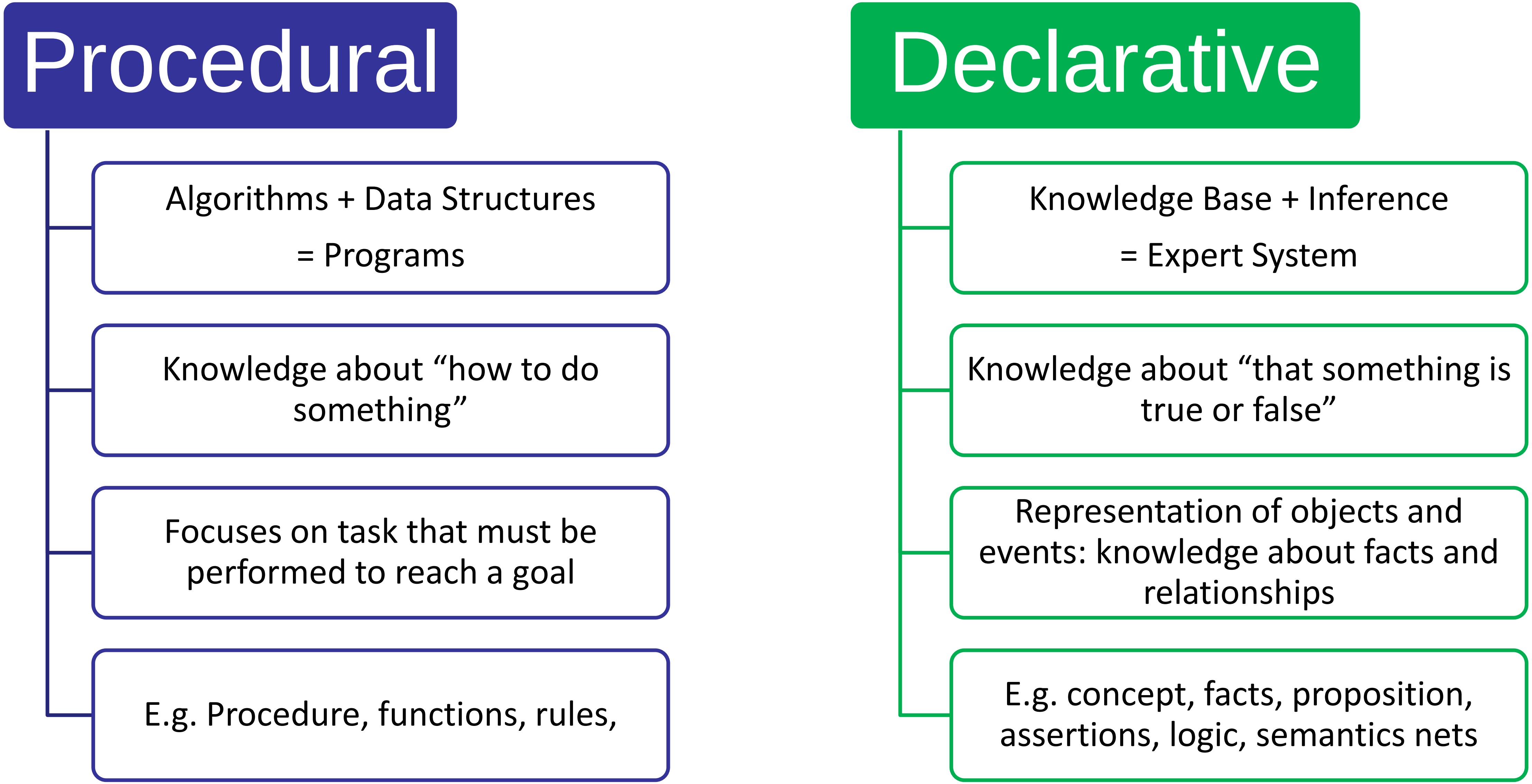
Asks what action should be done at the current time

Asserting that the chosen action was executed

- Agent program
  - TELLs the knowledge base what it perceives.
  - ASKs the knowledge base what action it should perform
  - TELLs the knowledge base which action was chosen

- Declarative Approach:
  - TELLing what need to know
  - TELL sentence one by one until agent know how to operate in environment

Approaches to Designing a Knowledge-Based Agent



Procedural

- Algorithms + Data Structures = Programs
- Knowledge about “how to do something”
- Focuses on task that must be performed to reach a goal
- E.g. Procedure, functions, rules,

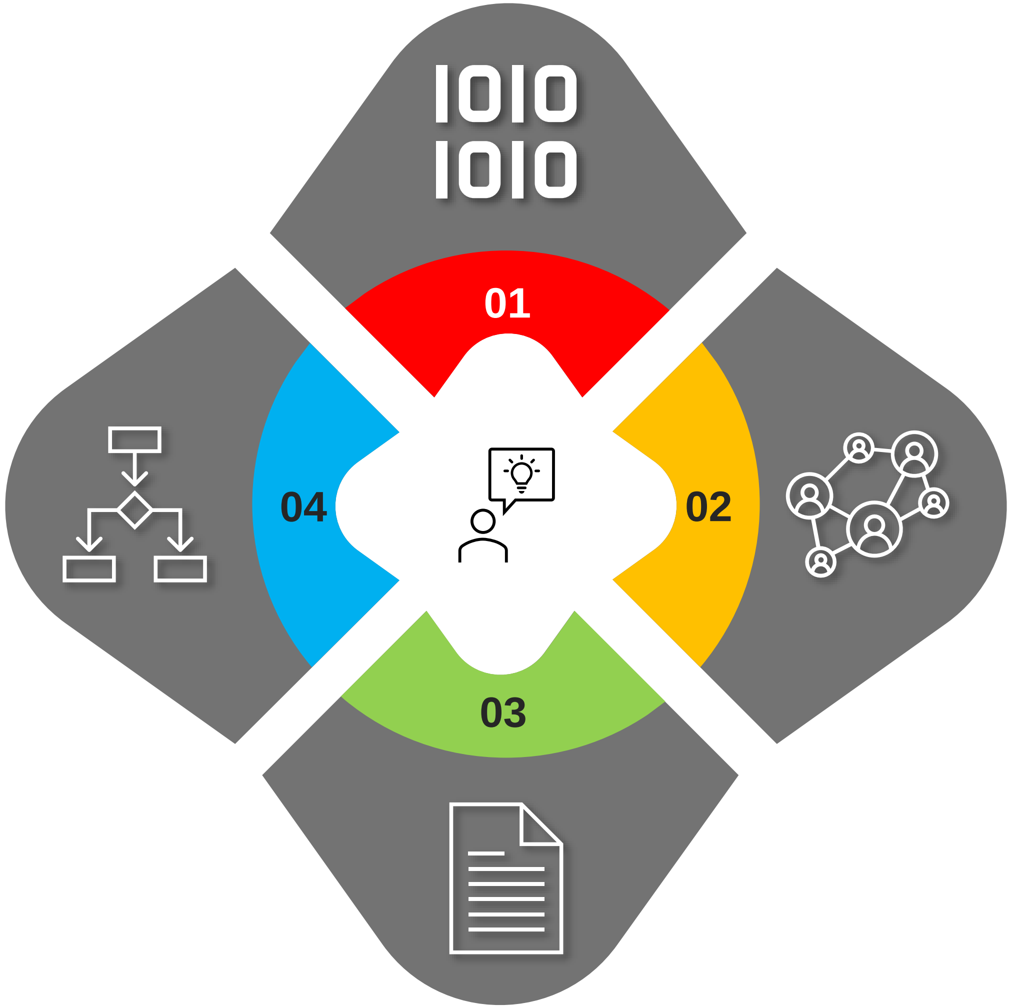
Declarative

- Knowledge Base + Inference = Expert System
- Knowledge about “that something is true or false”
- Representation of objects and events: knowledge about facts and relationships
- E.g. concept, facts, proposition, assertions, logic, semantics nets

Techniques of Knowledge Representation

Production Rules

Conditions and Relative Actions



Logical

Propositional Logic
First Order Logic

Frames

Record like structure which consists of a collection of attributes and its values

Semantic Network

Graphical Representation

Logical Representation

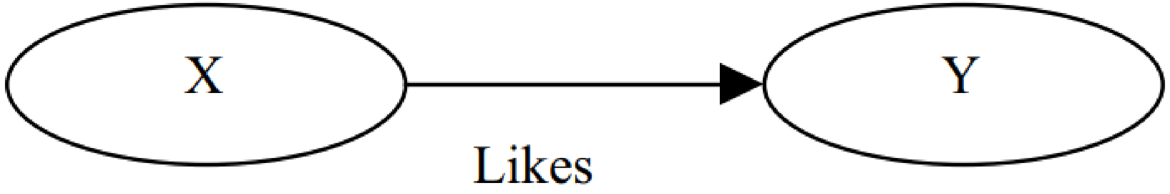
knowledge	propositional	first-order
Paul is a man	a	man(Paul)
Bill is a man	b	man(Bill)
men are mortal	c	$(\forall x)\ (\mathrm{man}(x) \supset \mathrm{mortal}(x))$

Structure Knowledge Representation

- Alternative approach for knowledge representation and reasoning.
- Knowledge Base Significance:
  - Easily represented by modular fashion
  - Explanation tracing becomes easier
  - Provision for representing monotonic, non-monotonic and default logic
  - Fragments of a piece of knowledge can be accessed concurrently by many modules
  - Provides a scope for massive parallelism in the reasoning process.
- Types:
  - Semantic Nets
  - Frames
  - Conceptual dependencies
  - Script

Semantic Network

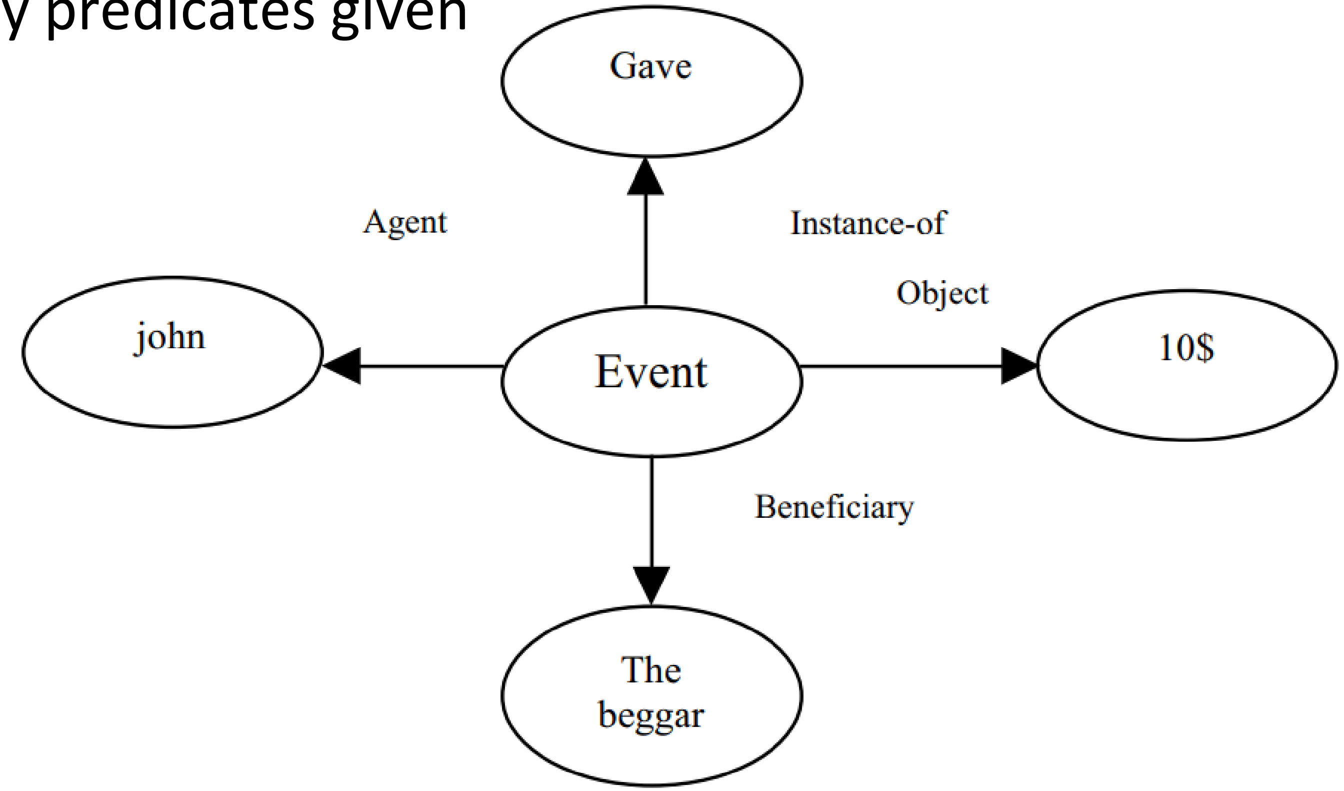
- Way of representing relationships between objects and ideas.
- Consist fact in Binary(or *n*-ary) predicate: entity – predicate – entity
  - Entity (Network nodes): any objects such as people, location, event
  - Predicate (Network edge): relationship type between the two entity, such as likes, friend_of, lives_in, is_a, part_of
  - Exampe: Likes (X, Y)



Semantic Network

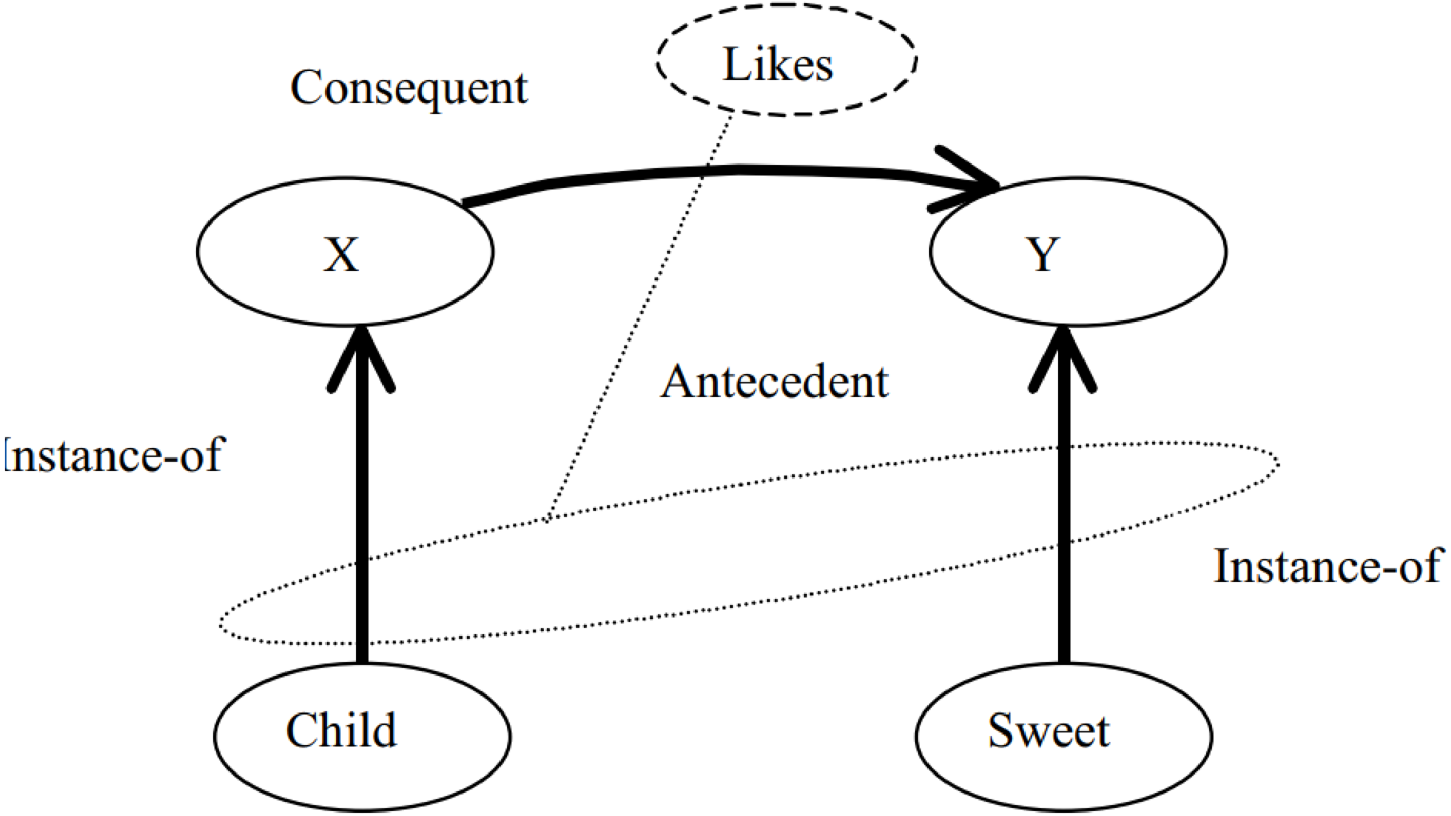
- Ternary predicate:
  - Gave(john, the-beggar, 10$)

  - Represented as collection of 4 Binary predicates given
    - Agent (john, event)
    - Beneficiary (the-beggar, event)
    - Object (10$, event)
    - Instance-of (give, event)



Semantic Network

- Capability of representation of quantifiers
  - ∀X ∃Y (Child (X) ∧ Sweet(Y) ⇒ Likes(X,Y))
  - Represented in Binary Predicate
  - ∀X ∃Y (Instance-of (X, child) ∧ Instance-of (Y, sweet) ⇒ Likes (X, Y))



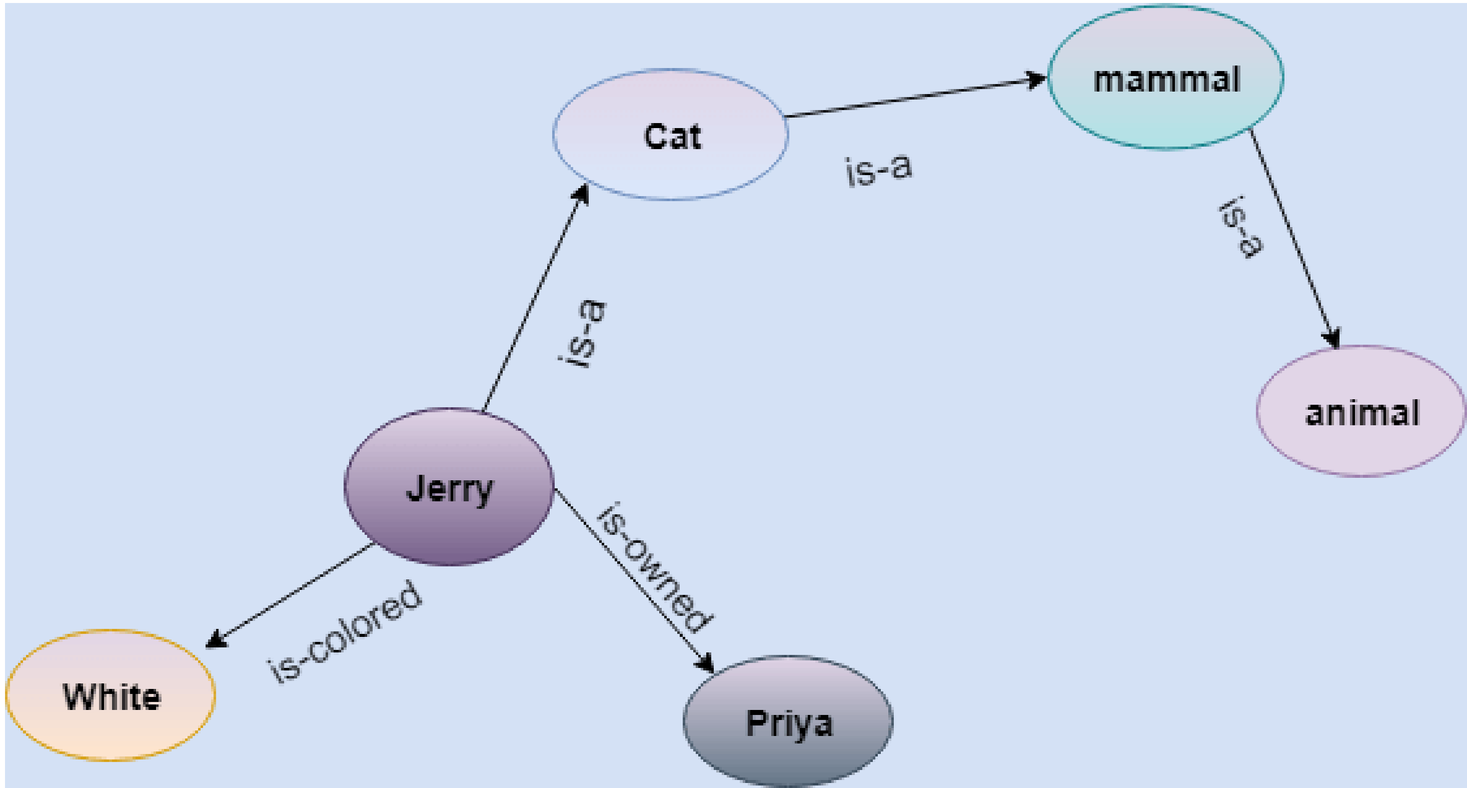
Inheritance in Semantic Network

- A binary relation $x < y$ is a partial order
  - when $x < y$ and $y < z$ holds good
- If the concept in a semantic net has a partial order, we call the network an inheritance system.
- While drawing the network, we, however, always omit the edge representing transitive inheritance.
- if there is a directed edge from node u → v and v → w, the edge u → w is obvious and thus omitted.

Semantic Network

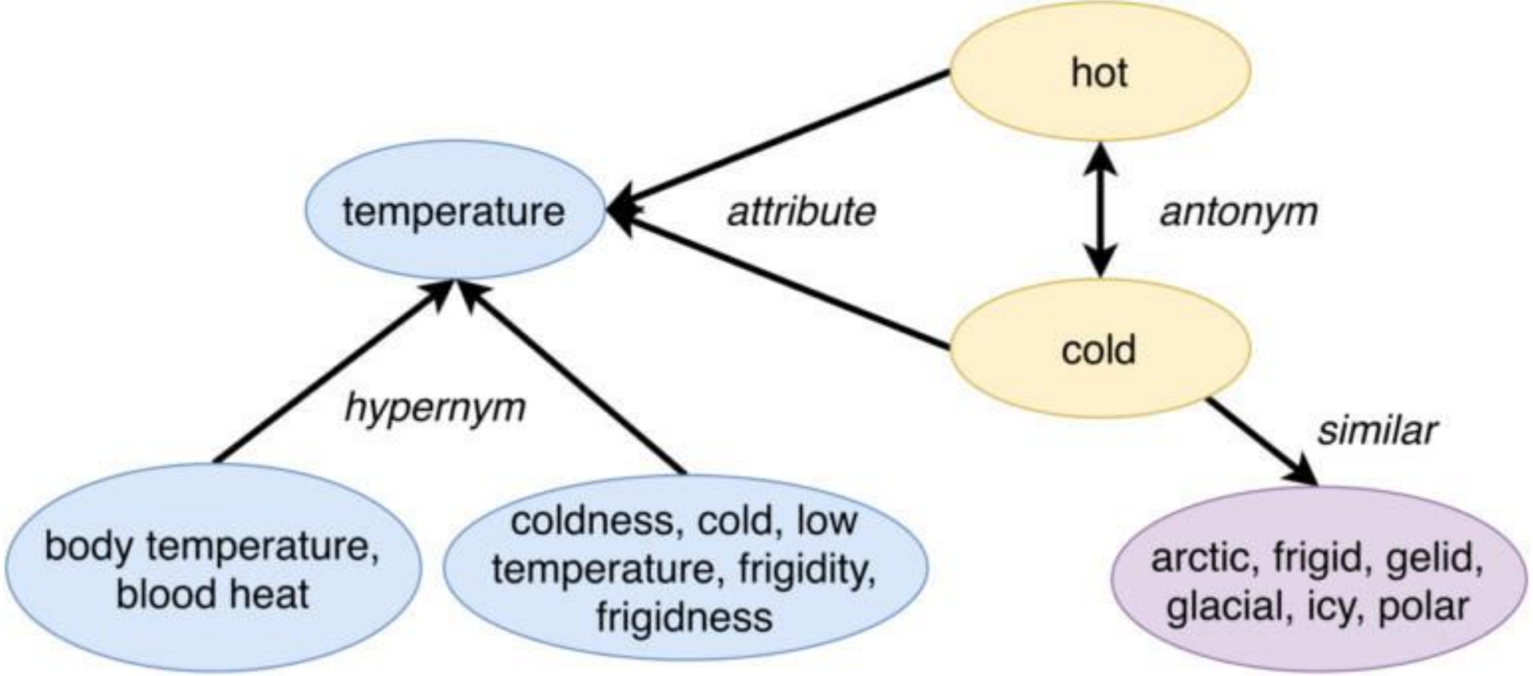
- Represent knowledge in the form of graphical networks.
- Nodes representing objects
- Arcs describe the relationship between those objects.
- It categorizes the object in different forms and links those objects.
- This representation consist of two types of relations:
  - IS-A relation (Inheritance)
  - Kind-of-relation

1. Jerry is a cat.
2. Jerry is a mammal
3. Jerry is owned by Priya.
4. Jerry is brown colored.
5. All Mammals are animal.



Application of Semantic Network

- WordNet:
  - large lexical database of English.
  - Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.
  - To help computers understand the connections between different words in the English language


hot
temperature
attribute
antonym
cold
hypernym
similar
body temperature, blood heat
coldness, cold, low temperature, frigidity, frigidness
arctic, frigid, gelid, glacial, icy, polar

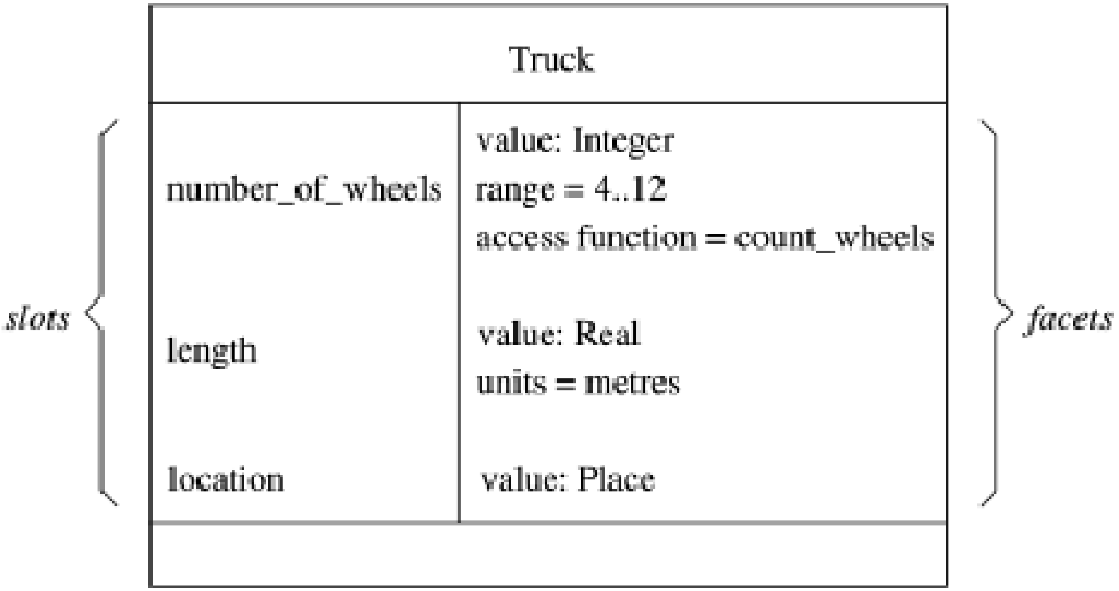
Application of Semantic Network

- SciCrunch:
  - collaboratively edited knowledge base about scientific resources.
  - provide a common source of data to the research community and the data about Research Resource Identifiers (RRIDs)

- Knowledge Graph:
  - knowledge base that uses a graph-structured data model or topology to integrate data.
  - Store interlinked descriptions of entities
    - Google Knowledge Graph
    - WikiData
    - DBPedia

Frame Representation

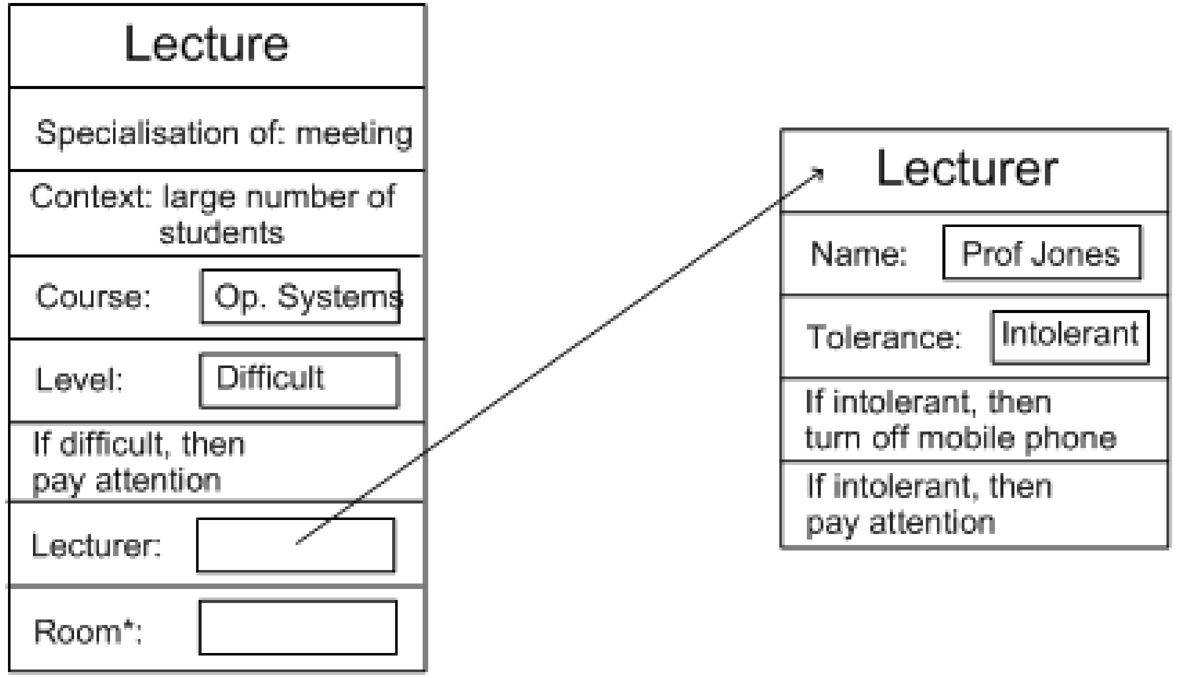
- A frame is a **record** like structure.
- Consists of a **collection of attributes** and **values** to describe an entity in the world.
- It consists of a collection of slots and slot values of any type and size.
- Slots have names and values which are called facets.



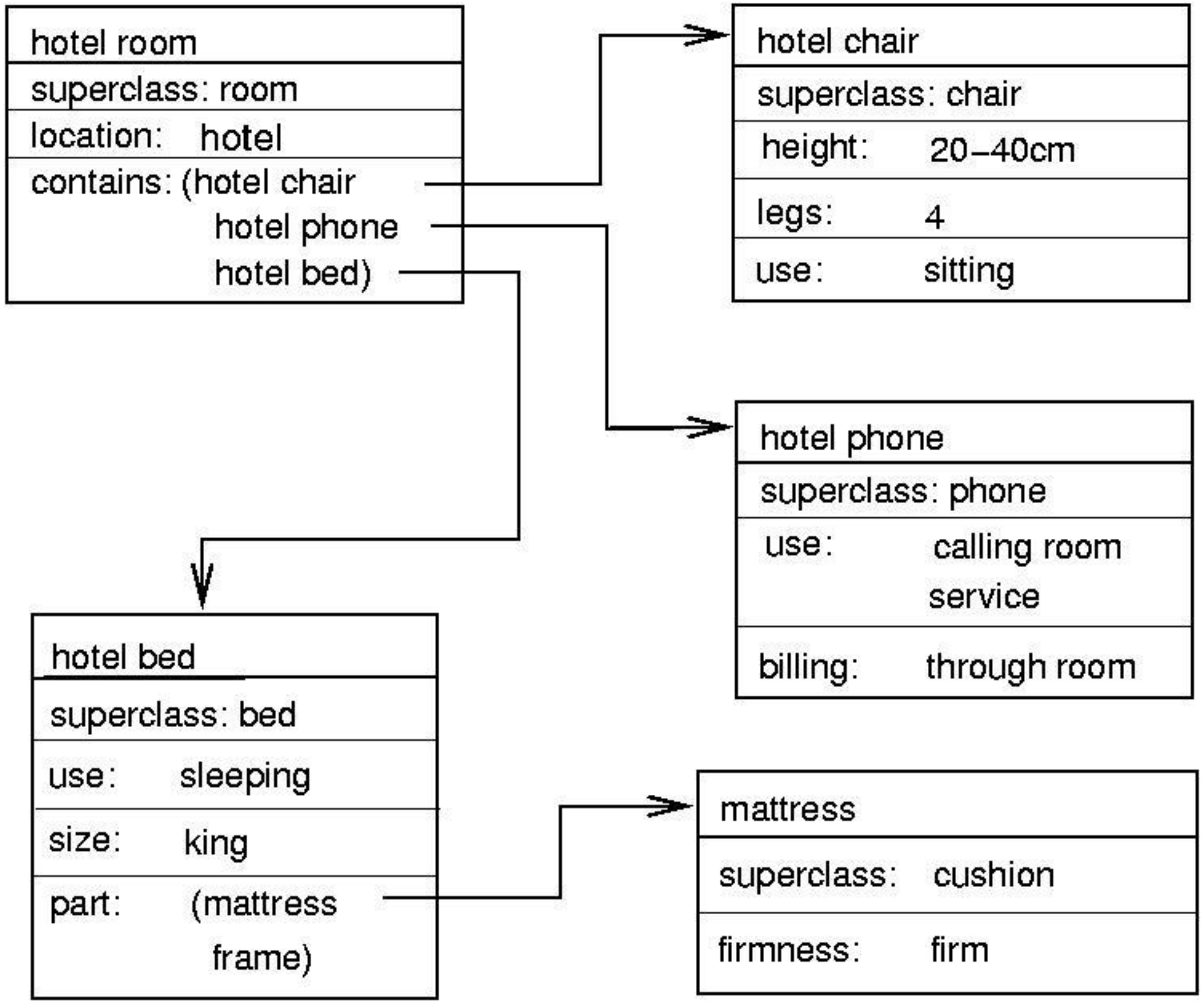
Truck	
number_of_wheels	value: Integer range = 4..12 access function = count_wheels
length	value: Real units = metres
location	value: Place

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


Lecture
Specialisation of: meeting
Context: large number of students
Course:
Op. Systems
Level:
Difficult
If difficult, then pay attention
Lecturer:
Room*:
Lecturer
Name:
Prof Jones
Tolerance:
Intolerant
If intolerant, then turn off mobile phone
If intolerant, then pay attention

Example: Frame Representation



hotel room
superclass: room
location: hotel
contains: (hotel chair
hotel phone
hotel bed)

hotel chair
superclass: chair
height: 20–40cm
legs: 4
use: sitting

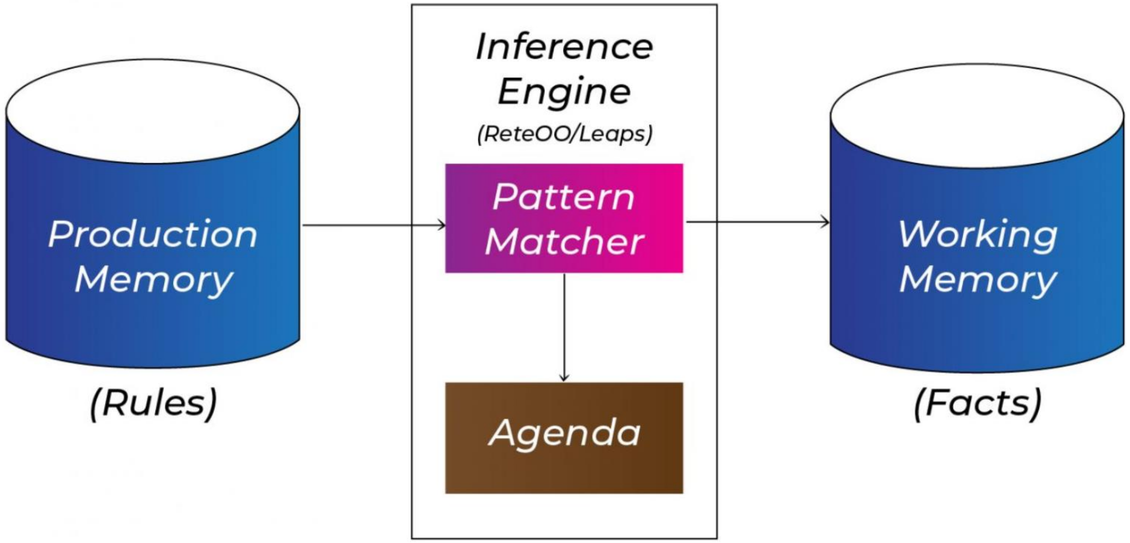
hotel phone
superclass: phone
use: calling room service
billing: through room

hotel bed
superclass: bed
use: sleeping
size: king
part: (mattress
frame)

mattress
superclass: cushion
firmness: firm

Production Base System

- Agent checks for the **condition** and if the condition exists then production rule fires and corresponding action is carried out.
- The condition part of the rule determines which rule may be applied to a problem.
- The action part carries out the associated problem-solving steps.
- This complete process is called a recognize-act cycle.
- The production rules system consists of three main parts:
    - The set of production rules
    - Working Memory
    - The recognize-act-cycle



IF (at bus stop AND bus arrives) THEN action (get into the bus)
IF (on the bus AND paid AND empty seat) THEN action (sit down).
IF (on bus AND unpaid) THEN action (pay charges).
IF (bus arrives at destination) THEN action (get down from the bus).

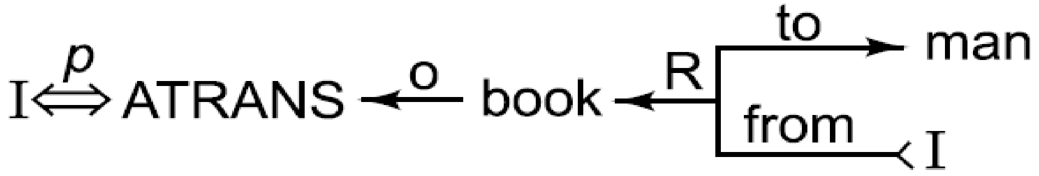
Other Representation

- Control Dependency
- Script

Conceptual Dependency (CD)

- CD theory was developed by Schank in 1973 to 1975 to represent the meaning of NL sentences.
  - It helps in drawing inferences
  - It is independent of the language
- CD representation of a sentence 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.

- “I gave the man a book.”



- Where the symbols have the following meanings:
  - **Arrows** indicate direction of dependency.
  - **Double arrow** indicates two way link between actor and action.
  - **p** indicates past tense.
  - **ATRANS** is one of the primitive acts used by the theory. It indicates transfer of possession.
  - **o** indicates the object case relation.
  - **R** indicates the recipient case relation.

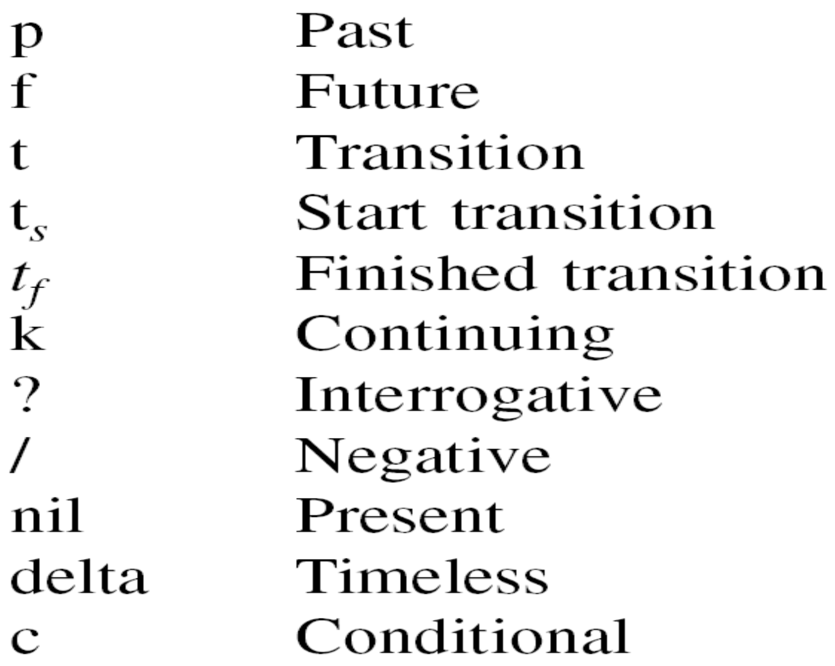
Primitive Acts of CD theory

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)

Conceptual category

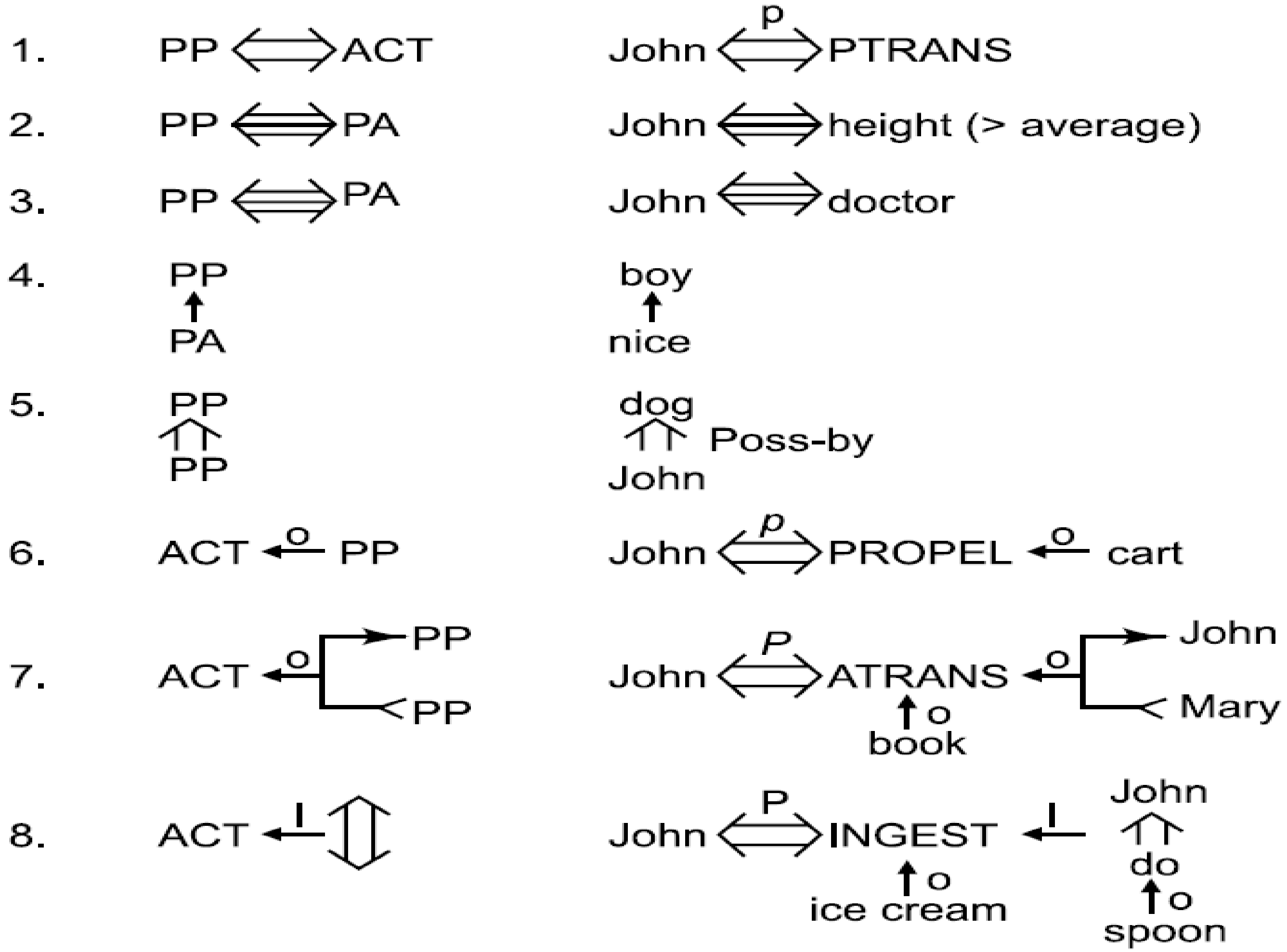
- There are four conceptual categories
  - ACT Actions {one of the CD primitives}
  - PP Objects {picture producers}
  - AA Modifiers of actions {action aiders}
  - PA Modifiers of PP’s {picture aiders}

- Conceptual Tenses



p	Past
f	Future
t	Transition
t_s	Start transition
t_f	Finished transition
k	Continuing
?	Interrogative
/	Negative
nil	Present
delta	Timeless
c	Conditional

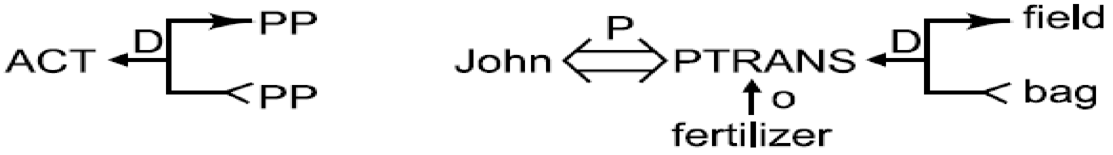
The Dependencies of CD



1.	PP $\Leftrightarrow$ ACT	John $\overset{p}{\Leftrightarrow}$ PTRANS	John ran.
2.	PP $\Lleftarrow\!\Rrightarrow$ PA	John $\Lleftarrow\!\Rrightarrow$ height (> average)	John is tall.
3.	PP $\Leftrightarrow$ PA	John $\Leftrightarrow$ doctor	John is a doctor.
4.	PP ↑ PA	boy ↑ nice	A nice boy.
5.	PP ⇑ PP	dog ⇑ Poss-by John	John’s dog.
6.	ACT $\xleftarrow{o}$ PP	John $\overset{p}{\Leftrightarrow}$ PROPEL $\xleftarrow{o}$ cart	John pushed the cart.
7.	ACT $\xleftarrow{o}$ → PP < PP	John $\overset{P}{\Leftrightarrow}$ ATRANS $\xleftarrow{o}$ → John < Mary ↑ o book	John took the book from Mary.
8.	ACT $\xleftarrow{I}$ ⇕	John $\overset{P}{\Leftrightarrow}$ INGEST $\xleftarrow{I}$ John ⇑ do ↑ o spoon ↑ o ice cream	John ate ice cream with a spoon.

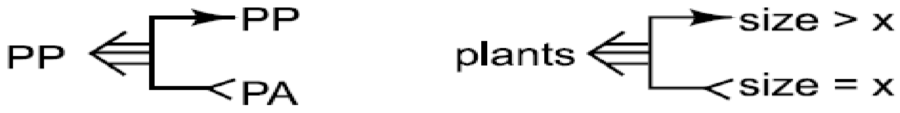
The Dependencies of CD

9.



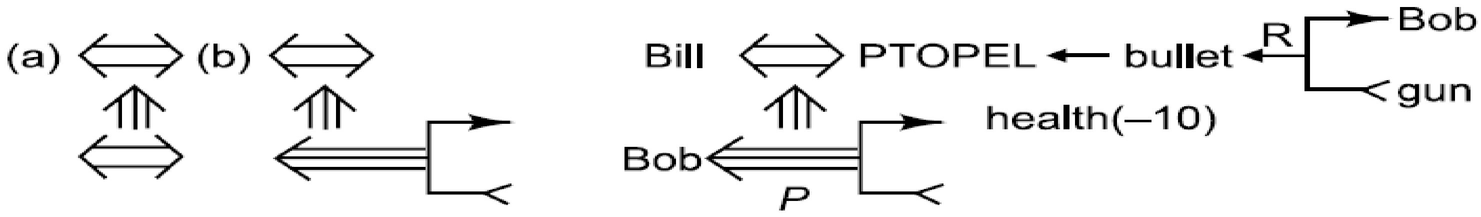
John fertilized the field.

10.



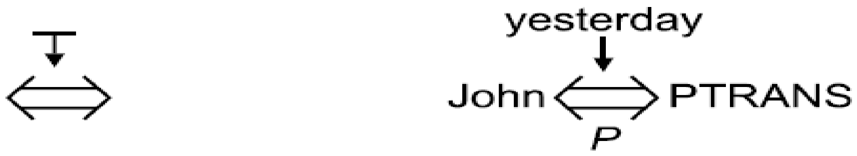
The plants grew.

11.



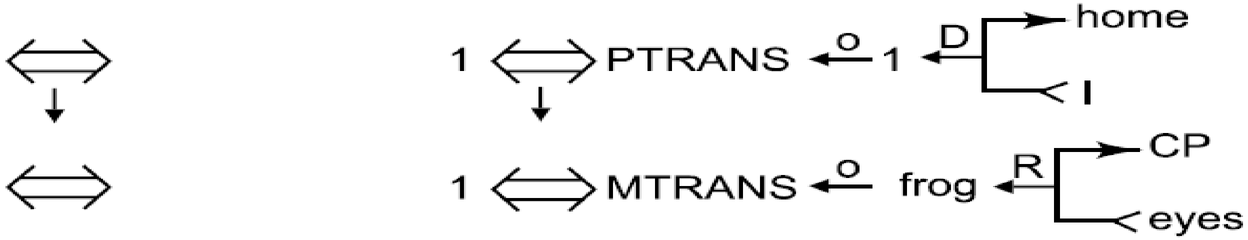
Bill shot Bob.

12.



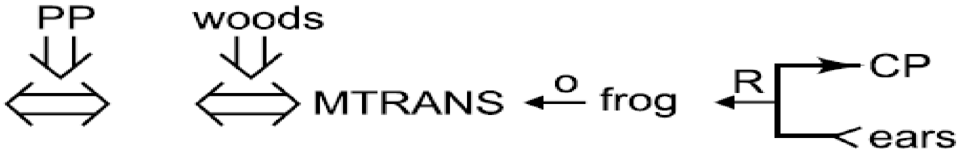
John ran yesterday.

13.



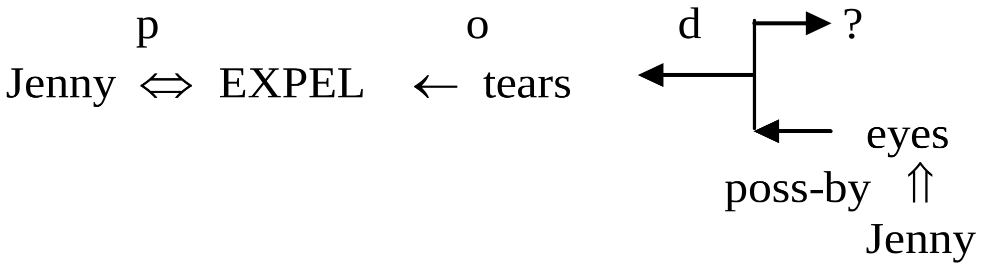
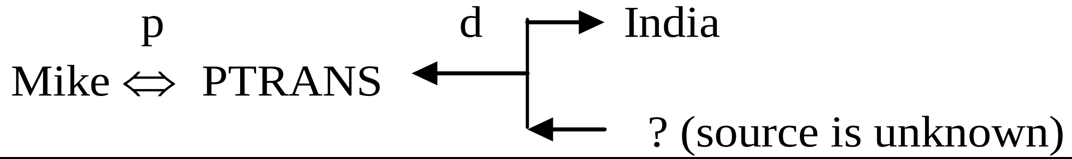
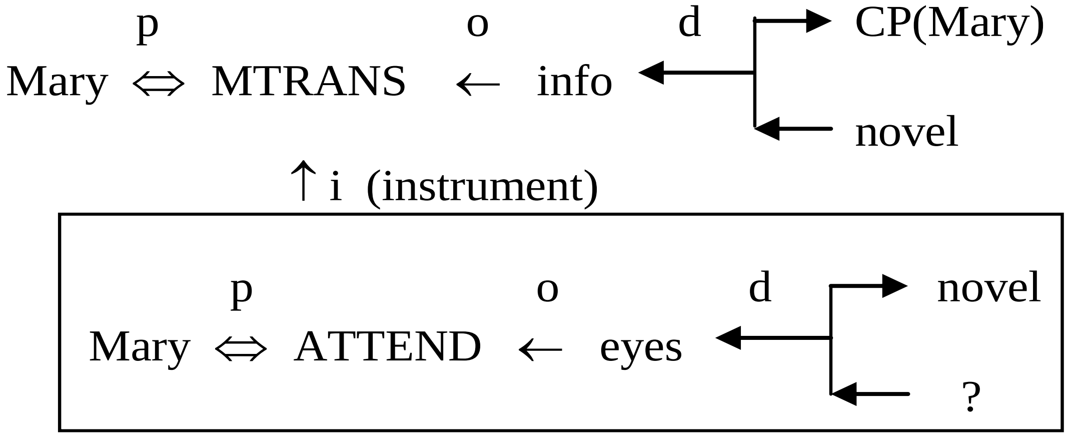
While going home, I saw a frog.

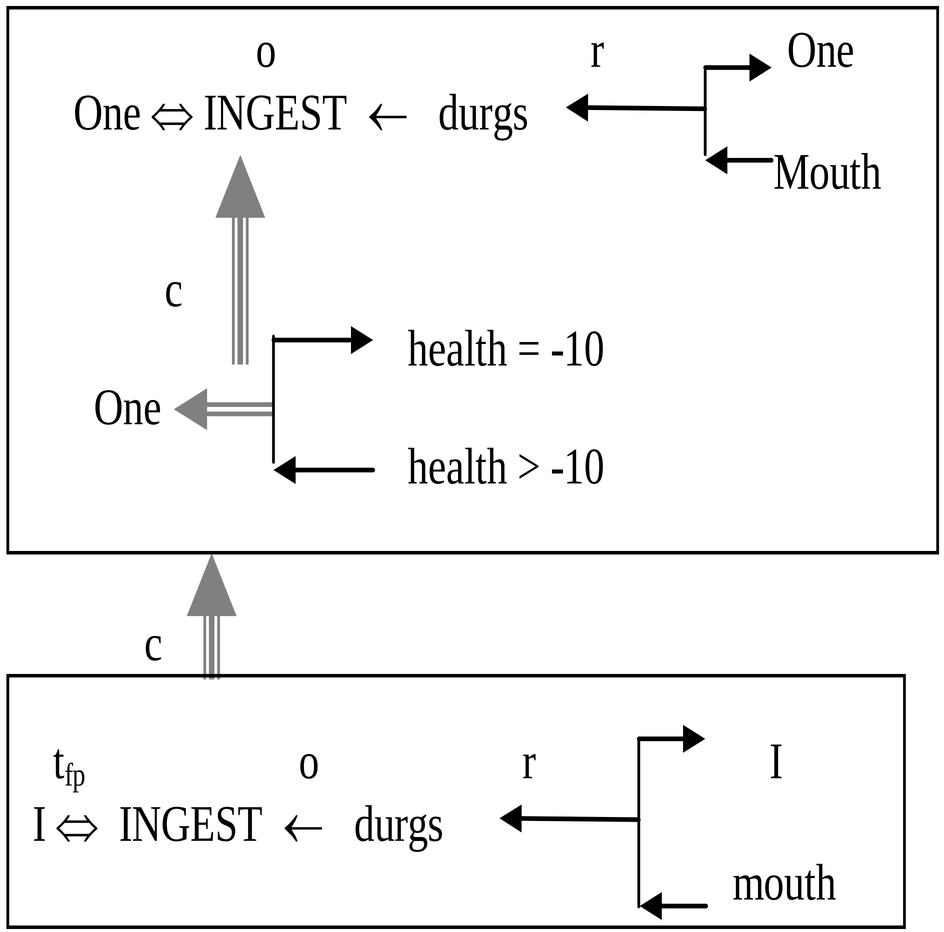
14.



I heard a frog in the woods.

Generation of CD representations

Sentences	CD Representations
Jenny cried	 p o d ? Jenny ⇔ EXPEL ← tears eyes poss-by ⇑ Jenny
Mike went to India	 p d India Mike ⇔ PTRANS ? (source is unknown)
Mary read a novel	 p o d CP(Mary) Mary ⇔ MTRANS ← info novel ↑ i (instrument) p o d novel Mary ⇔ ATTEND ← eyes ?

Sentence	CD Representation
Since drugs can kill, I stopped.	 o r One One ⇔ INGEST ← durgs Mouth c health = -10 One health > -10 c t_{fp} o r I I ⇔ INGEST ← durgs mouth

Problems with CD Representation

- It is difficult to
  - construct original sentence from its corresponding CD representation.
  - CD representation can be used as a general model for knowledge representation, because this theory is based on representation of events as well as all the information related to events.
- Rules are to be carefully designed for each primitive action in order to obtain semantically correct interpretation.
- Many verbs may fall under different primitive ACTs, and it becomes difficult to find correct primitive in the given context.
- The CD representation becomes complex requiring lot of storage for many simple actions.
- For example, the sentence “John bet Mike that Indian cricket team will win incoming world cup” will require huge CD structure.

Script Structure

- Scripts were introduced by Schank and Abelson introduced in 1977 that used CD framework.
- The scripts are useful in describing certain stereotyped situations such as going to theater
- It consists of set of slots containing default values along with some information about the type of values similar to frames.
- It differs from FS as the values of the slots in scripts must be ordered and have more specialized roles.
- In real world situations, we see that event tends to occur in known patterns because of clausal relationship to the occurrence of events

The Component of a Script

Entry conditions: Conditions that must, in general, be satisfied before the events described in the script can occur.

Result: Conditions that will, in general, be true after the events described in the script have occurred.

Props: Slots representing objects that are involved in the event described in the script.

Roles: Slots representing people who are involved in the events described in the script.

Track: The specific variation on a more general pattern that is represented by the particular script.

Scenes: The actual sequences of events that occur.

Script : Play in theater	**Various Scenes**
Track: Play in Theater **Props:** • Tickets • Seat • Play **Roles:** • Person (who wants to see a play) – P • Ticket distributor – TD • Ticket checker – TC **Entry Conditions:** • P wants to see a play • P has a money **Results:** • P saw a play • P has less money • P is happy (optional if he liked the play)	***Scene 1: Going to theater*** • P PTRANS P into theater • P ATTEND eyes to ticket counter
	Scene 2: Buying ticket • P PTRANS P to ticket counter • P MTRANS (need a ticket) to TD • TD ATRANS ticket to P
	Scene 3: Going inside hall of theater and sitting on a seat • P PTRANS P into Hall of theater • TC ATTEND eyes on ticket POSS_by P • TC MTRANS (showed seat) to P • P PTRANS P to seat • P MOVES P to sitting position
	Scene 4: Watching a play • P ATTEND eyes on play • P MBUILD (good moments) from play
	Scene5: Exiting • P PTRANS P out of Hall and theater

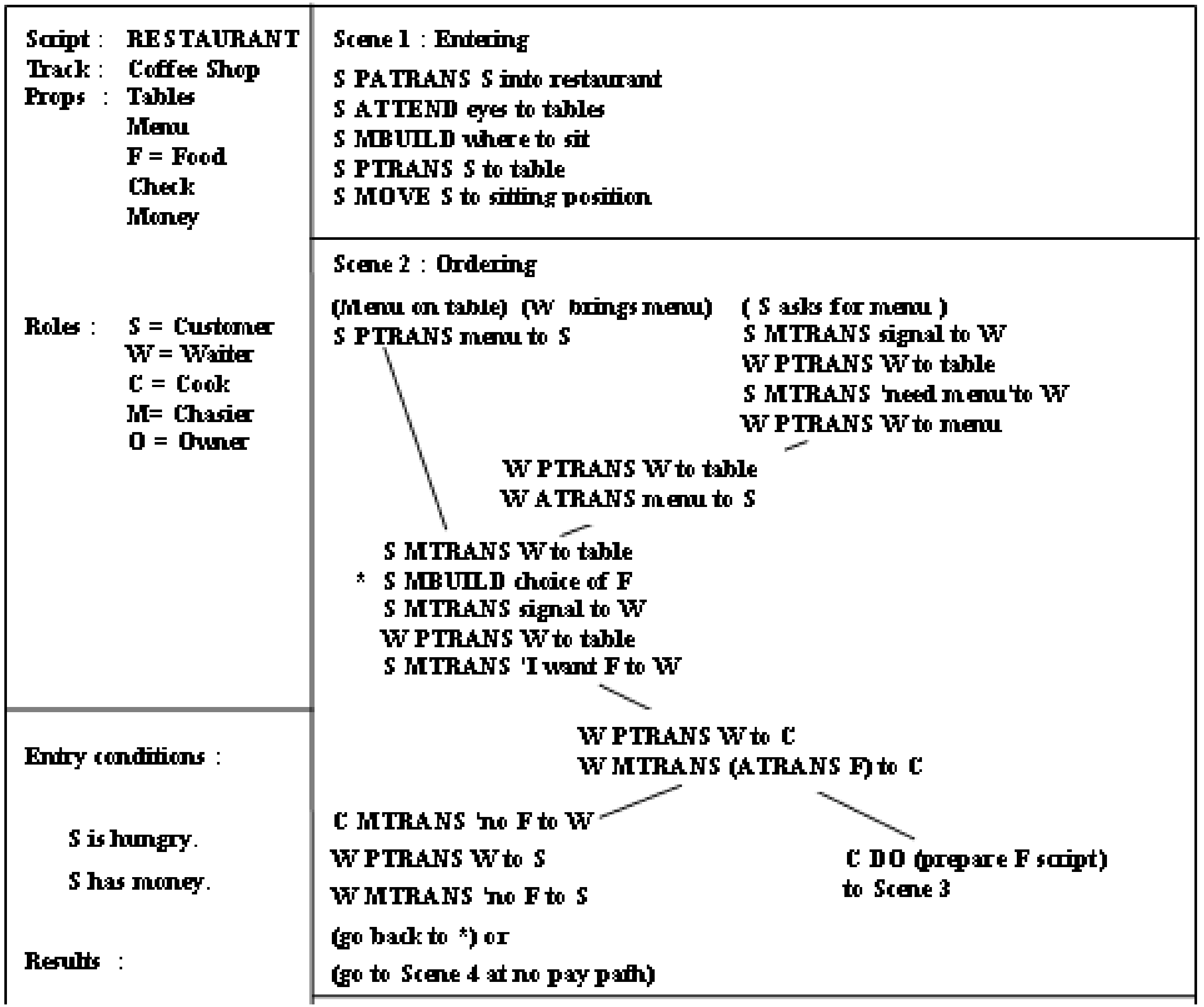
Script Invocation

- It must be activated based on its significance.
- If the topic is important, then the script should be opened.
- If a topic is just mentioned, then a pointer to that script could be held.
- For example, given “John enjoyed the play in theater”, a script “Play in Theater” suggested above is invoked.
- All implicit questions can be answered correctly.
- Here the significance of this script is high.
  - Did john go to theater?
  - Did he buy ticket?
  - Did he have money?
- If we have a sentence like “John went to theater to pick his daughter”, then invoking this script will lead to many wrong answers.
  - Here significance of the script theater is less.
- Getting significance from the story is not straightforward. However, some heuristics can be applied to get the value.

Advantages / Disadvantages of Script

- 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 combined in a way that is appropriate for the situation.

The Restaurant Script



Script : RESTAURANT
Track : Coffee Shop
Props : Tables
Menu
F = Food
Check
Money

Roles : S = Customer
W = Waiter
C = Cook
M= Chasier
O = Owner

Entry conditions :

S is hungry.

S has money.

Results :

Scene 1 : Entering

S PATRANS 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 asks for menu)

S PTRANS menu to S

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

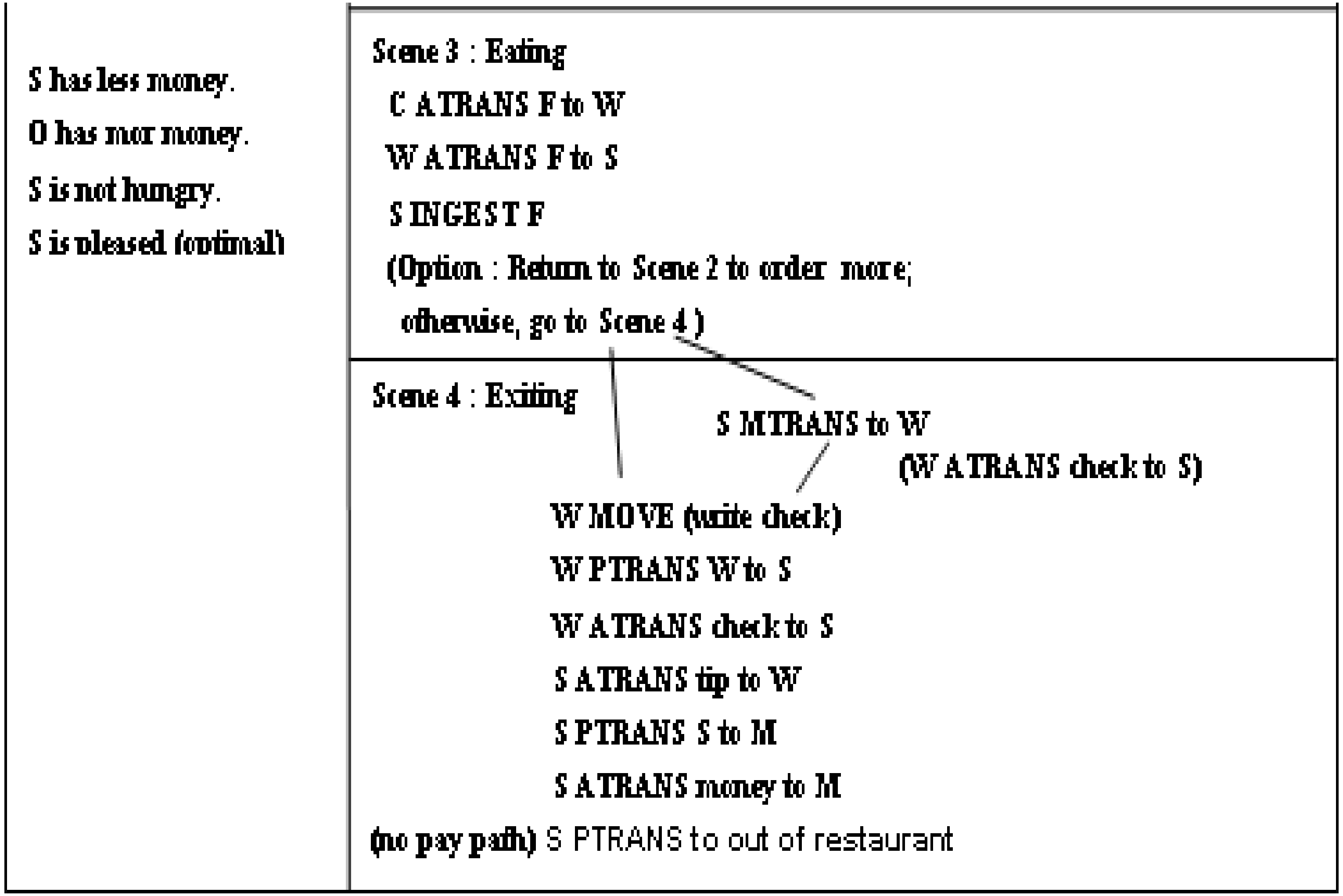
S MTRANS W to table
* S MBUILD choice of F
S MTRANS signal to W
W PTRANS W to table
S MTRANS 'I want F to W

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

The Restaurant Script



S has less money.

O has mor money.

S is not hungry.

S is pleased (optimal)

Scene 3 : Eating

C ATRANS F to W

W ATRANS F to S

S INGEST F

(Option : Return to Scene 2 to order more;

otherwise, go to Scene 4)

Scene 4 : Exiting

S MTRANS to W

(W ATRANS check to S)

W MOVE (write check)

W PTRANS W to S

W ATRANS check to S

S ATRANS tip to W

S PTRANS S to M

S ATRANS money to M

(no pay path) S PTRANS to out of restaurant

Triggering and Using Scripts

Susan passed her favorite restaurant on her way to the museum. She really enjoyed the new Picasso exhibit.

John went out to a restaurant last night. He ordered steak. When he paid for it, he noticed that he was running out of money. He hurried home since it had started to rain.

Triggering and Using Scripts

Susan went out to lunch. She sat down at a table and called the waitress. The waitress brought her a menu and she ordered a hamburger.

John went to a restaurant. He was shown to his table. He ordered a large steak. He sat there and waited for a long time. He got mad and left.