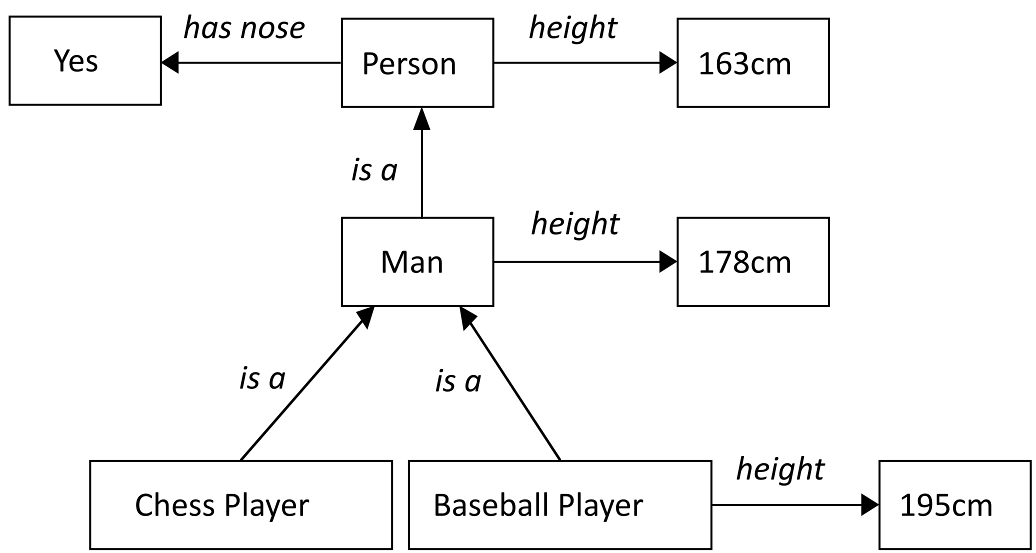
Semantic Networks

Inheritance and Defaults

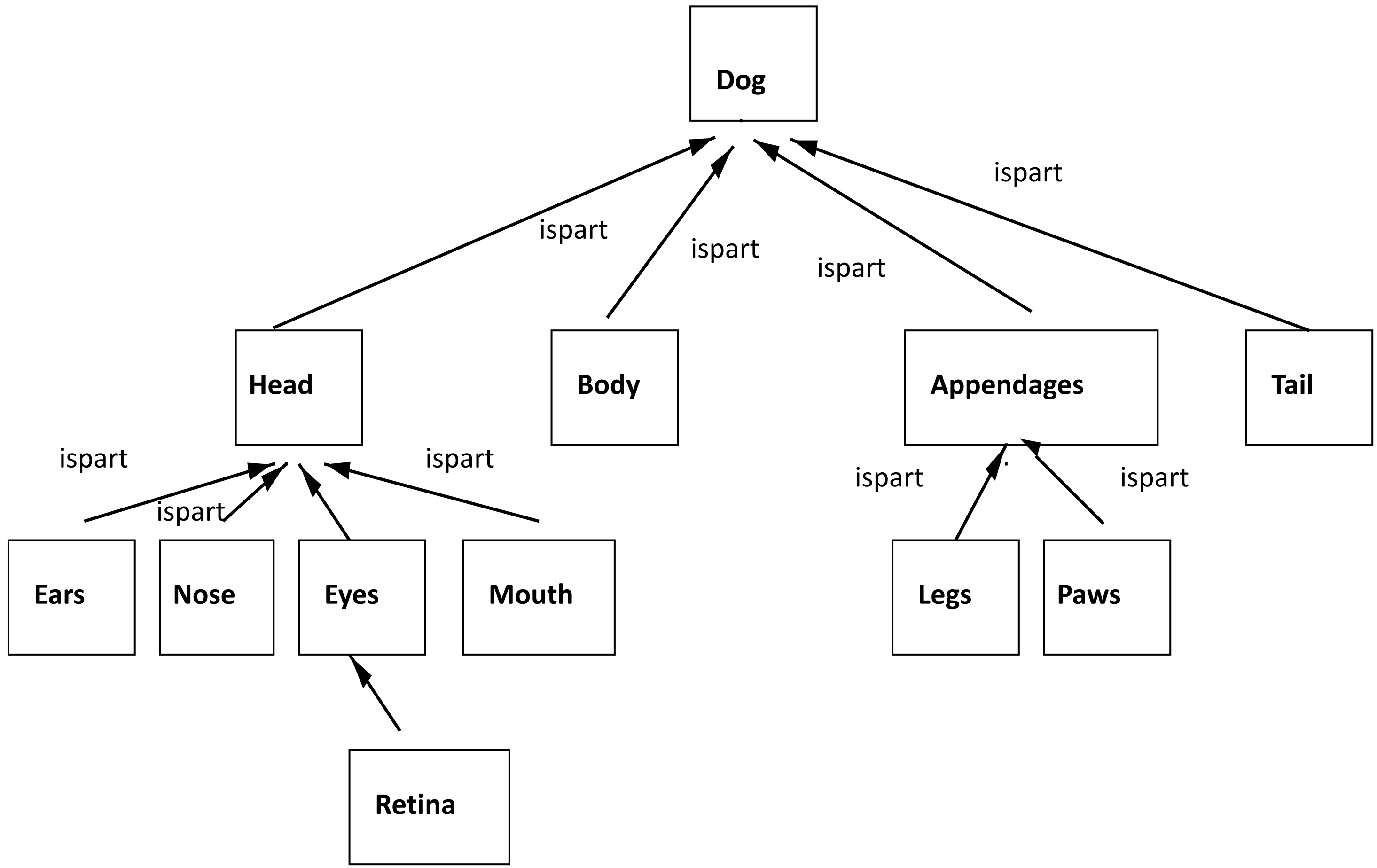
The figure shows the hierarchy of default values example.

- **I*s-a* corresponds to subset ⊆**



An IS-PART Hierarchy

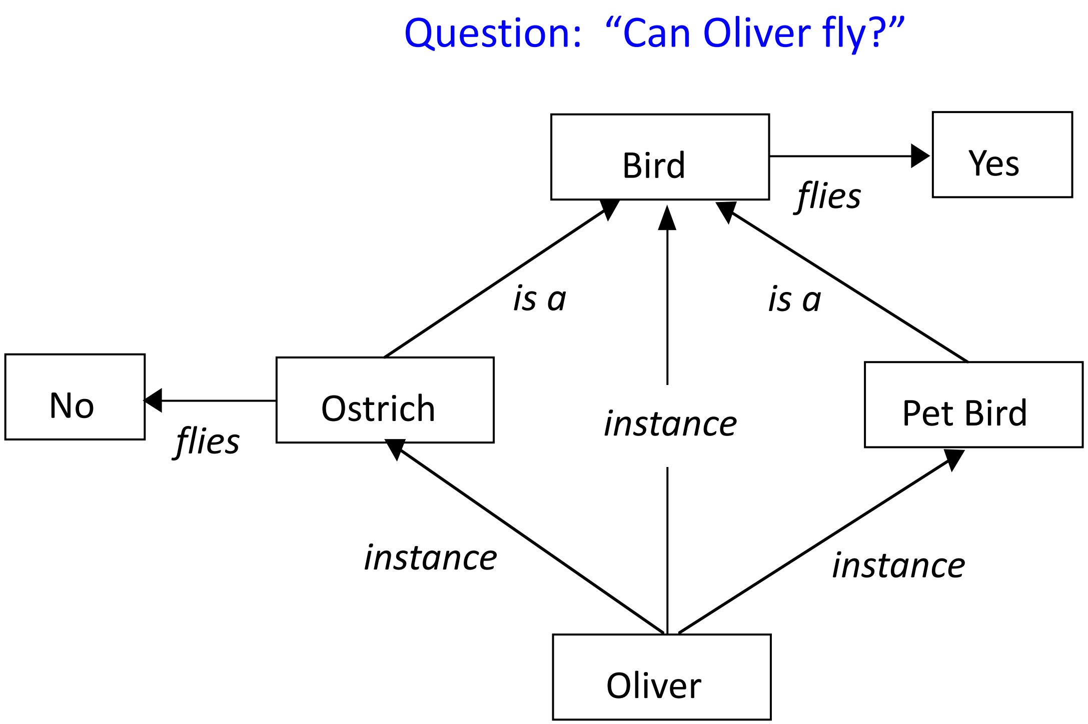
- ***Is-part* related to physical composition**



Naturally, where we choose to stop the hierarchy depends on what we want to represent.

Tangled Hierarchies

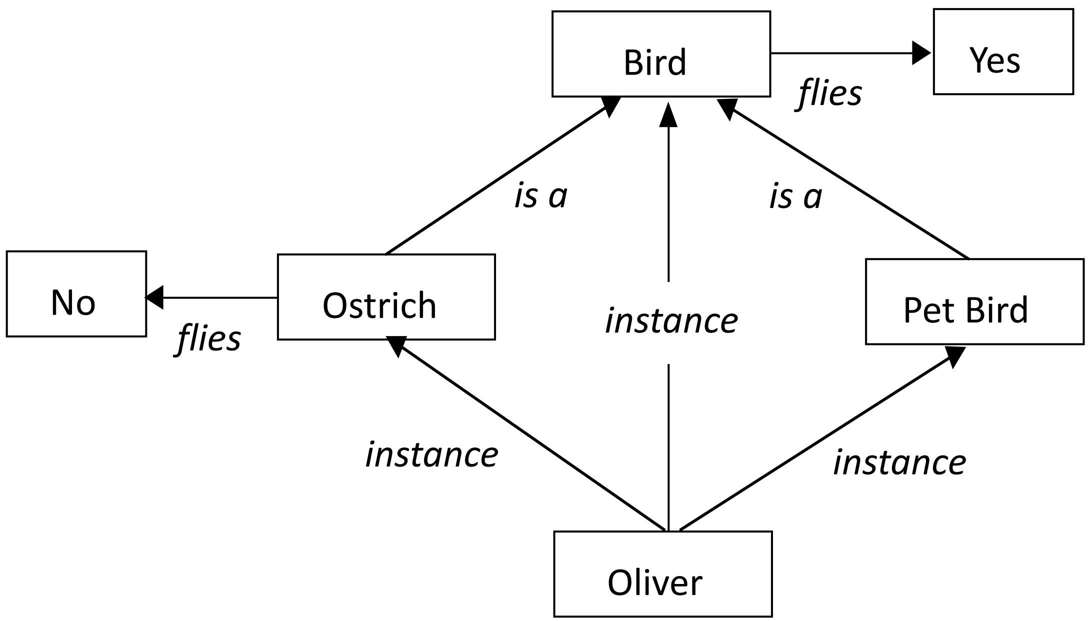
Tangled hierarchies results into inheritance conflict as shown below.



Tangled Hierarchies

Tangled hierarchies results into inheritance conflict as shown below.

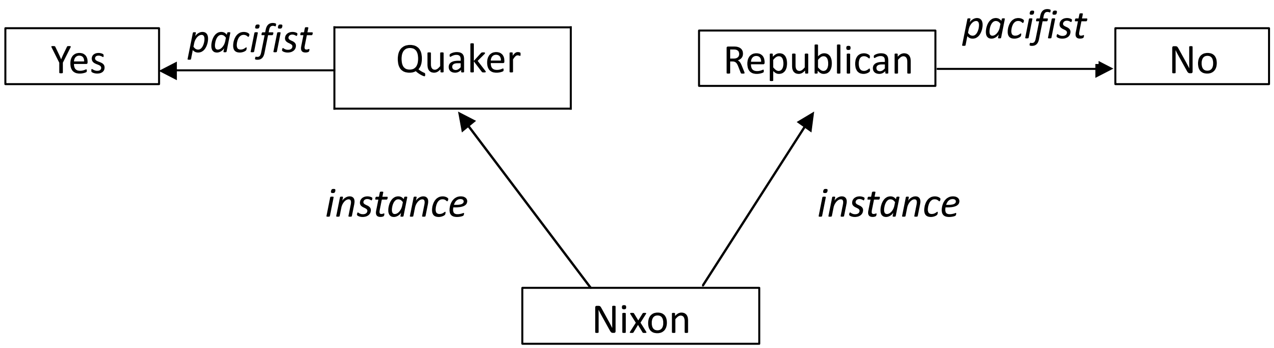
Question: "Can Oliver fly?"



Solution: a specific "flies no" for all individual instances of an ostrich.

Multiple Inheritance

Figure shows the example for the multiple inheritance.



Pacifist: a person who believes that war and violence are unjustifiable.

Conflicts: “Is Nixon a pacifist?”

In practice, we aim to build semantic networks in which all such conflicts are either over-ridden, or resolved appropriately.

Encoding N-Ary Relations

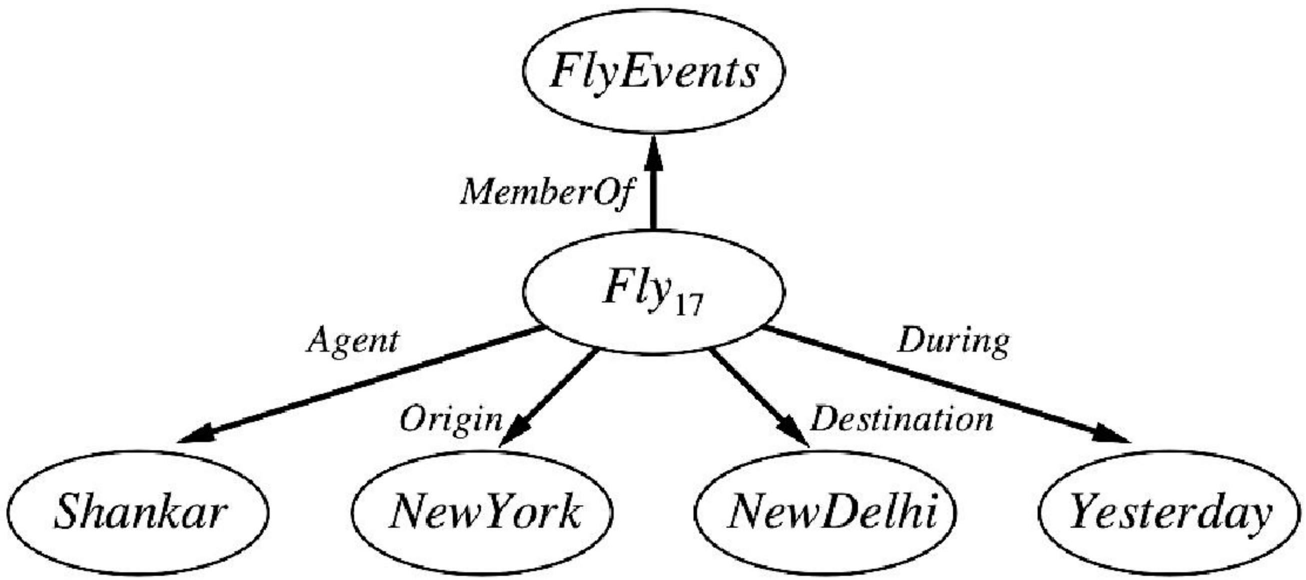
How to represent n-ary relations?

"Fly_{17}" for $Fly(Shankar, NewYork, NewDelhi, Yesterday)$

Encoding N-Ary Relations

How to represent n-ary relations?

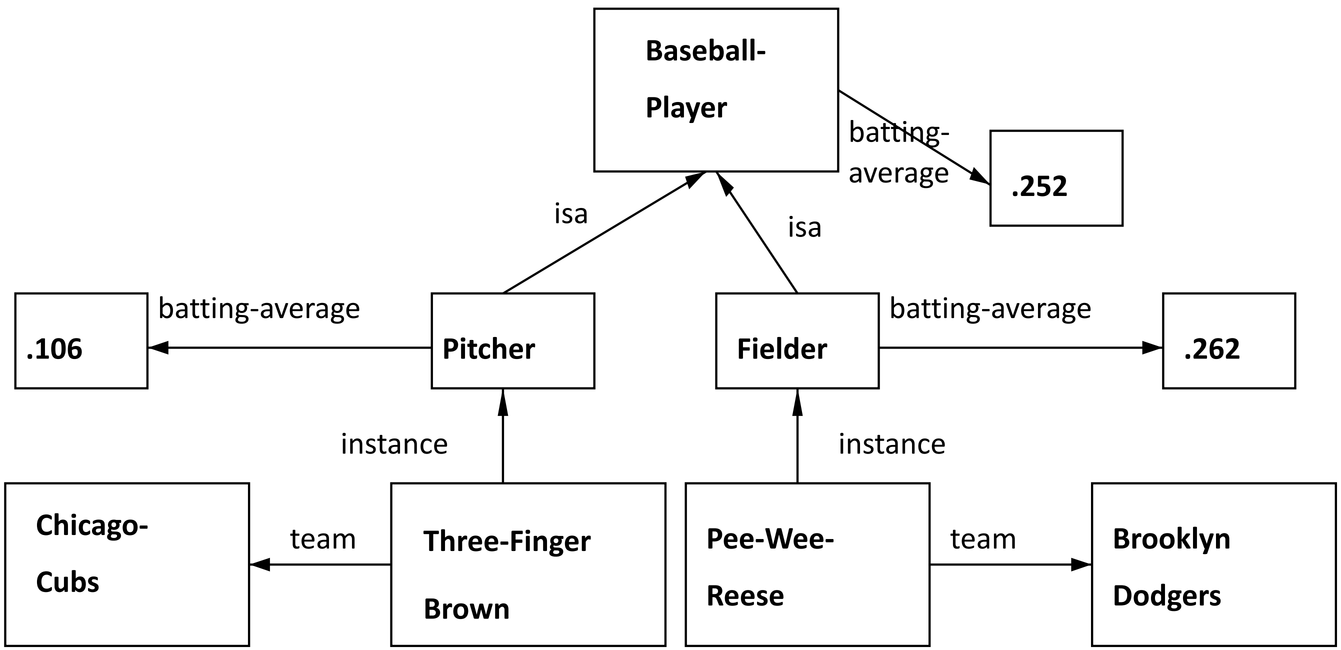
"Fly_{17}" for $Fly(Shankar, NewYork, NewDelhi, Yesterday)$



Intersection Search

The process to find relationships between objects by spreading ***activation*** from each of two nodes and seeing where the activations met.

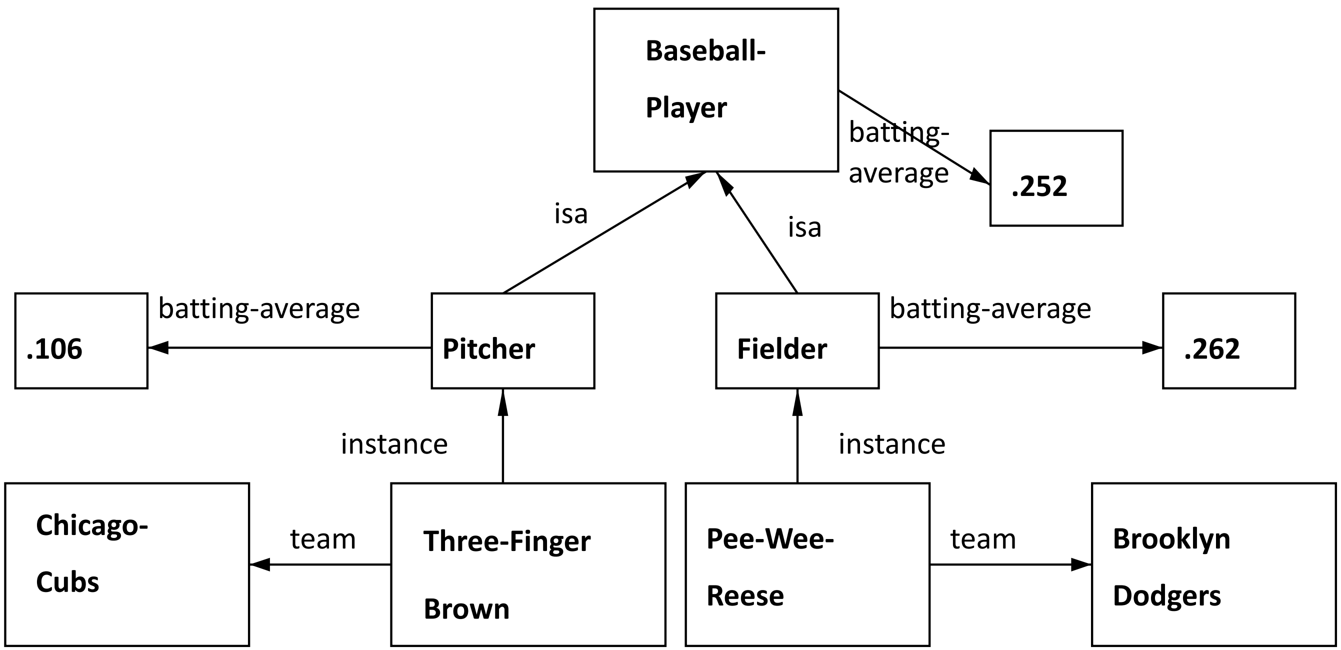
Question: “What is the relation between Chicago cubs and Brooklyn Dodgers?”



Intersection Search

The process to find relationships between objects by spreading ***activation*** from each of two nodes and seeing where the activations met.

Question: "What is the relation between Chicago cubs and Brooklyn Dodgers?"



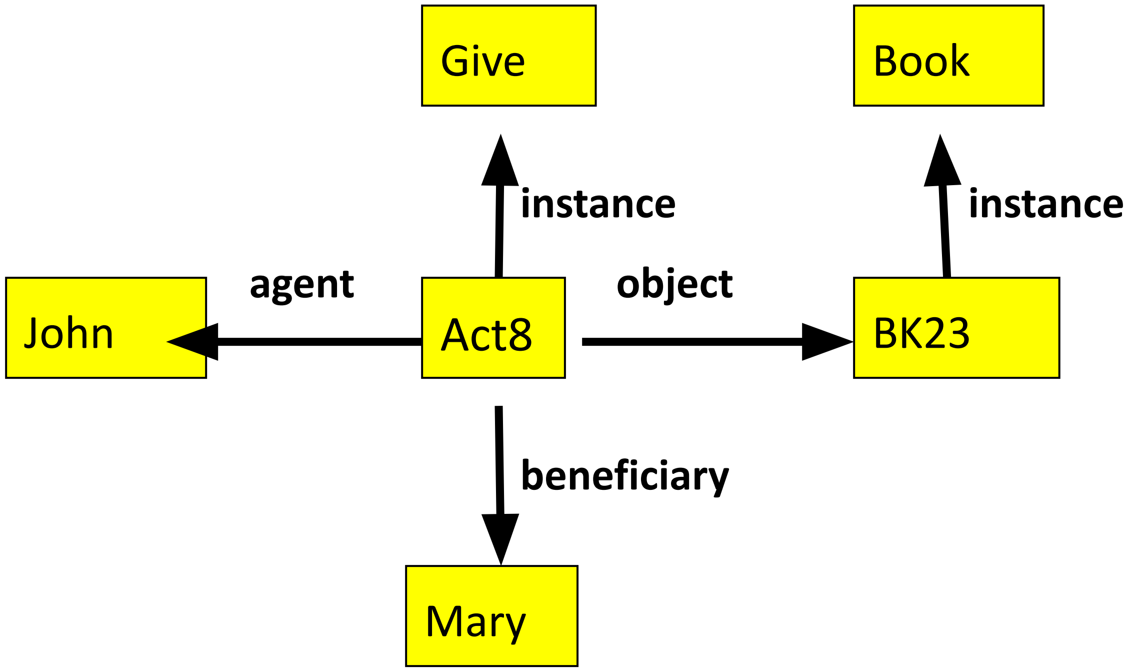
Answer: "They are both teams of baseball players."

A Semantic Net Representing a Sentence

“John gave the book to Mary.”

A Semantic Net Representing a Sentence

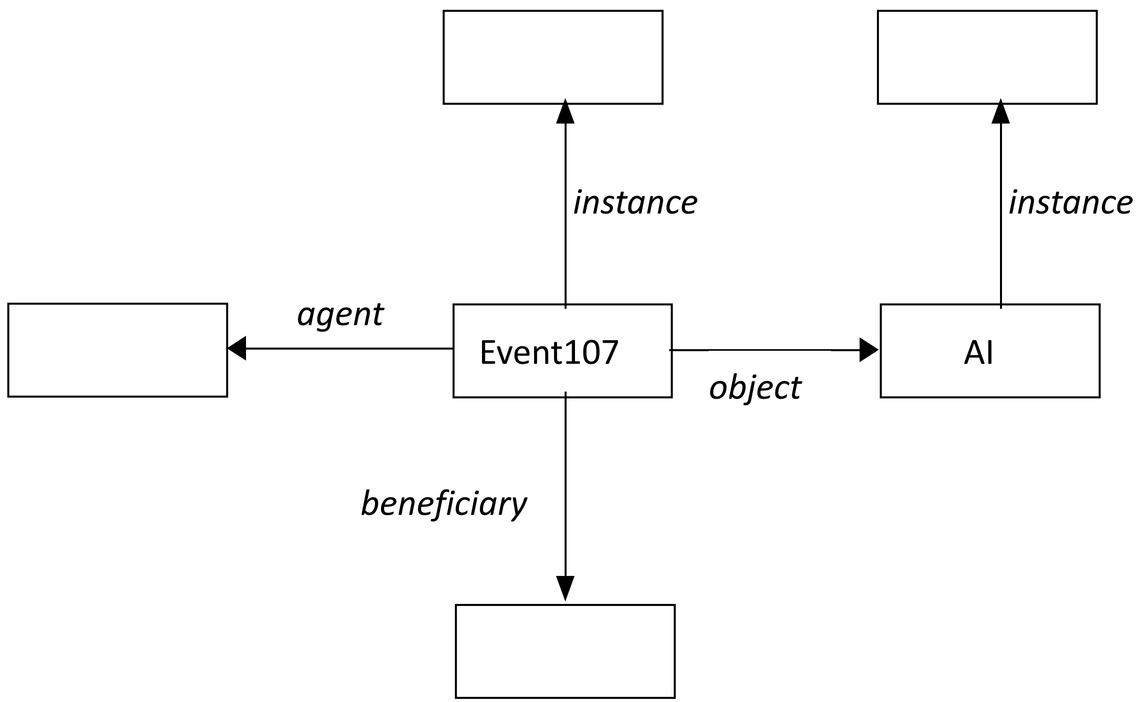
"John gave the book to Mary."



I*nstance* corresponds to element ∈

A Semantic Net Representing a Sentence

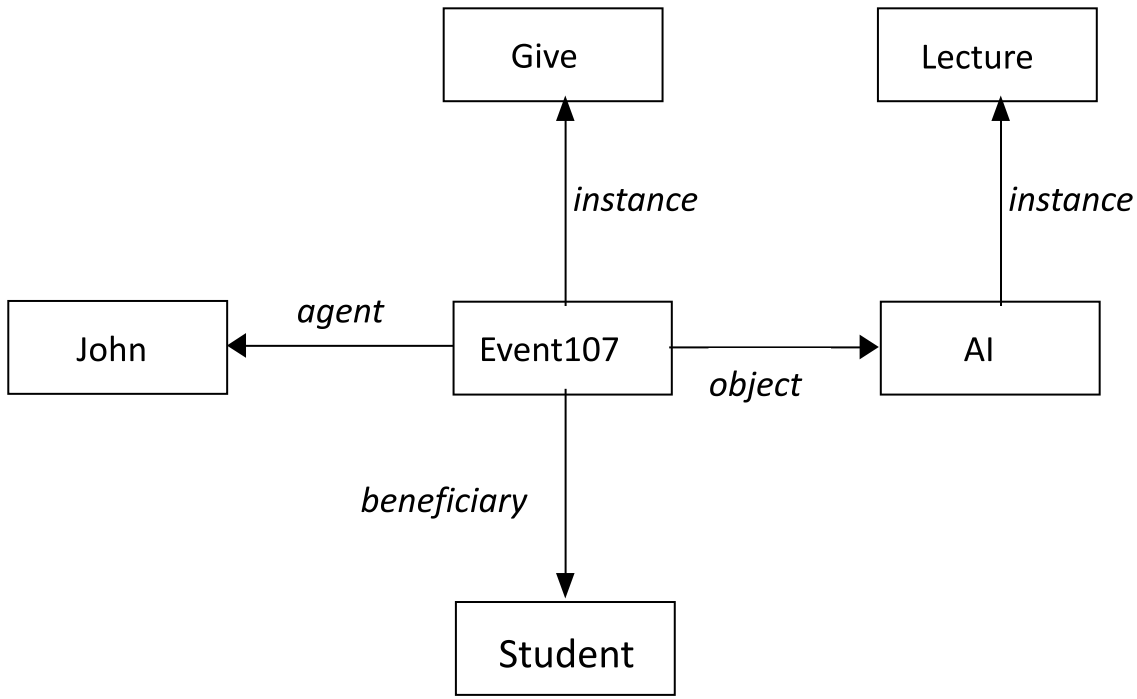
"John gave lecture AI to his students"



In fact, several of the earliest semantic networks were English-understanding programs.

A Semantic Net Representing a Sentence

"John gave lecture AI to his students"



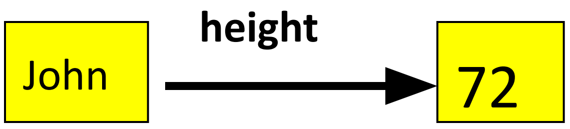
In fact, several of the earliest semantic networks were English-understanding programs.

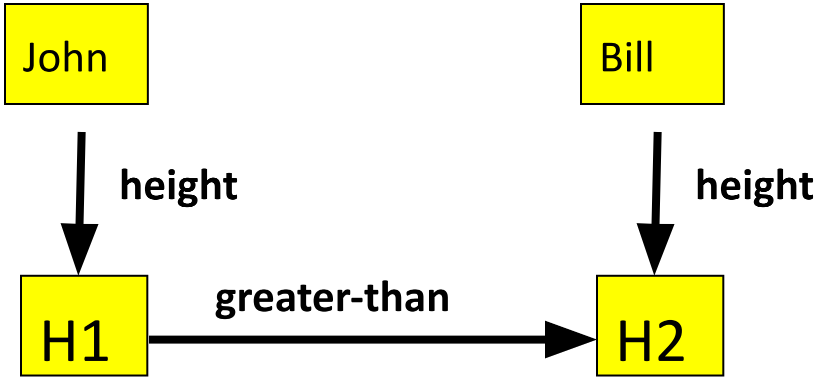
Comparison

John height is greater-than Bill height

Comparison

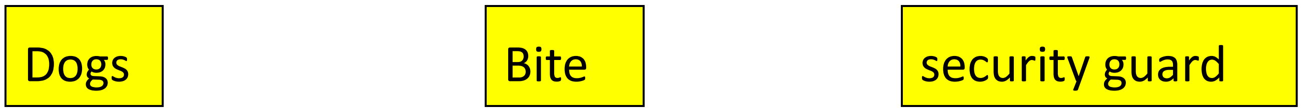
John height is greater-than Bill height





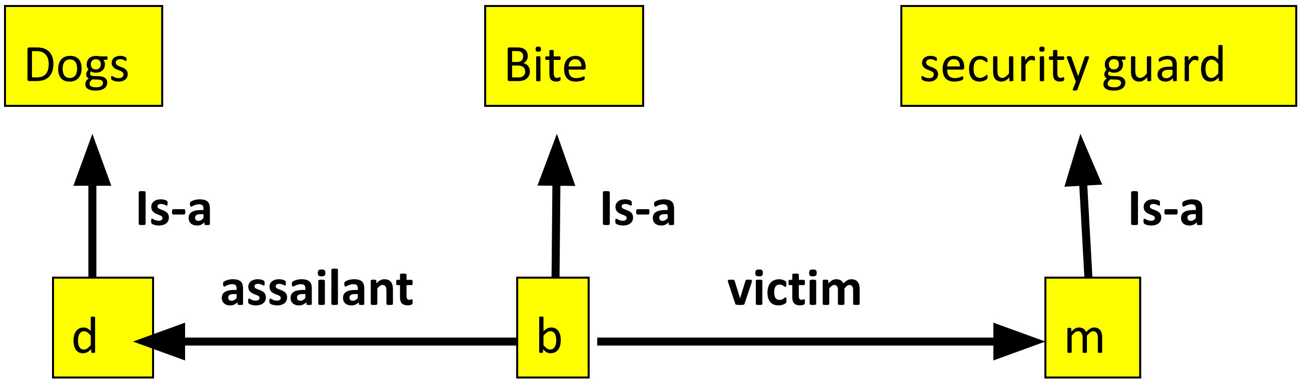
Semantic Nets

The dog bit the security guard.



Semantic Nets

The dog bit the security guard.



Semantic Networks Summary

- Semantic networks can be used to analyze large texts and identify the main themes and topics (e.g., of social media posts).
- Inheritance detection straightforward
- Ability to represent default values for categories
- Semantic networks are used in natural language processing applications, e.g., WordNet based word-sense disambiguation.
- The Knowledge Graph proposed by Google in 2012 is very similar to application of semantic network in search engine.

Semantic Networks Summary

- The notion of a semantic network is extremely general, which can be a problem.
- Limited expressive power: cannot represent negation and nested function symbols, etc.

Frames

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
A.I. LABORATORY

June 1974

Artificial Intelligence
Memo No. 306

A FRAMEWORK FOR REPRESENTING KNOWLEDGE*

MARVIN MINSKY

ABSTRACT

This is a partial theory of thinking, combining a number of classical and modern concepts from psychology, linguistics, and AI. Whenever one encounters a new situation (or makes a substantial change in one's

Frames as a Knowledge Representation

- S*emantic networks* represents labelled connections between entities. They become increasingly complex and can not solve various problems.
- Frames assign more structure to nodes and links to keep track of nodes and their relations, helps to model complex problems.
- A complete frame based representation will consist of a whole hierarchy or network of frames connected together by appropriate links/pointers.

Frames and Slots

hotel room
is-a: room
location: hotel
contains: {hotel chair, hotel bed, hotel phone, …}
⋮

hotel chair
is-a: chair
location: hotel room
height: 20-40cm
legs: 4
comfortable: yes
use: for sitting on
⋮

- Above are examples of class frames. We can also have instance frames for individuals.
- Each piece of information about a particular frame is held in a slot.

Slots

- Slot contain one or more facets, where facets can take following form.
  - Facts or Data
    - Values (called facets)
  - Procedures (also called procedural attachments)
    - IF-NEEDED: if a value of a facet is not known and we want to acquire it.
    - IF-ADDED: if a new value has been added to a facet.
  - Default Values
    - For Data
    - For Procedures
  - Pointer to other frames (inheritance) or Other frames

Frame Boy

Slot	Value	Type
BOY	_	(This Frame)
ISA	Person	(parent frame)
SEX	Male	(instance value)
AGE	Under 12 yrs.	(procedural attachment - sets constraint)
HOME	A Place	(frame)
NUM_LEGS	Default = 2	(default, inherited from Person frame)

Frame Alex

Slot	Value	Type
ALEX	_	(This Frame)
NAME	Alex	(key value)
ISA	Boy	(parent frame)
SEX	**Male**	**(inheritance value)**
BIRTHDATE	8/4/2000	(instance value)
AGE	IF-NEEDED: Subtract(current,BIRTHDATE);	(procedural attachment)
HOME	100 Main St.	(instance value)
FAVORITE_FOOD	Spaghetti	(instance value)
NUM_LEGS	1	(exception)

Frame Alex

Slot	Value	Type
ALEX	_	(This Frame)
NAME	Alex	(key value)
ISA	Boy	(parent frame)
SEX	Male	(inheritance value)
BIRTHDATE	**8/4/2000**	**(instance value)**
AGE	**IF-NEEDED: Subtract(current,BIRTHDATE);**	**(procedural attachment)**
HOME	100 Main St.	(instance value)
FAVORITE_FOOD	Spaghetti	(instance value)
NUM_LEGS	1	(exception)

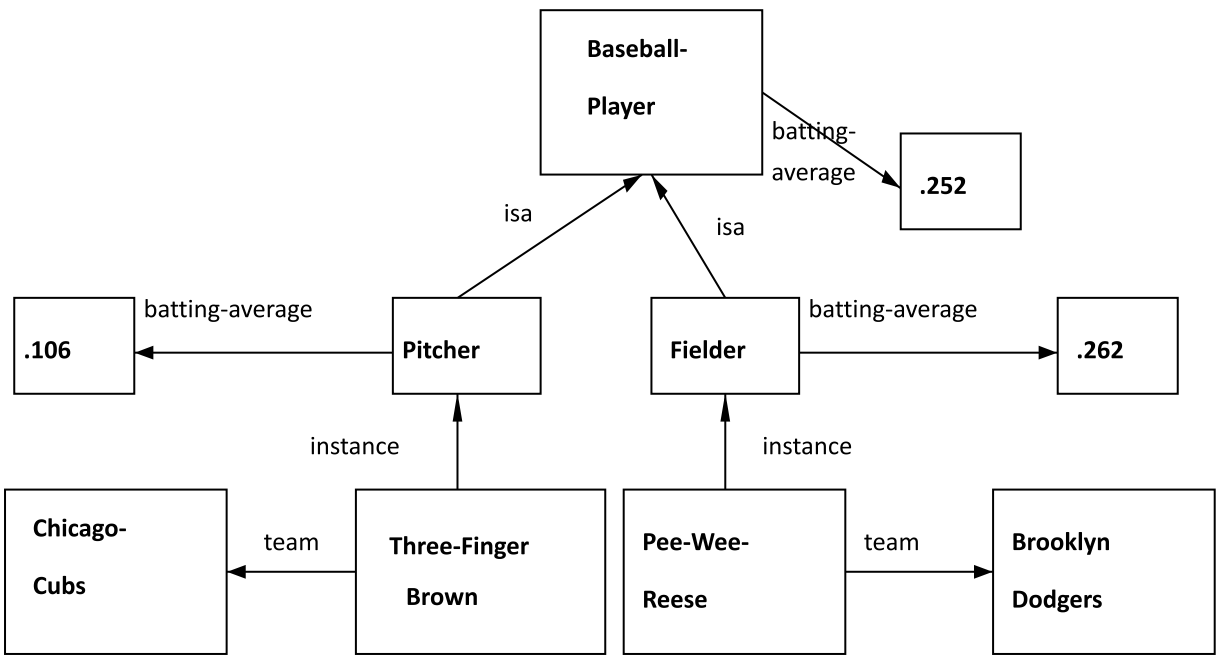
Frame Alex

Slot	Value	Type
ALEX	_	(This Frame)
NAME	Alex	(key value)
ISA	Boy	(parent frame)
SEX	Male	(inheritance value)
BIRTHDATE	8/4/2000	(instance value)
AGE	IF-NEEDED: Subtract(current,BIRTHDATE);	(procedural attachment)
HOME	100 Main St.	(instance value)
FAVORITE_FOOD	Spaghetti	(instance value)
NUM_LEGS	**1**	**(exception)**

Semantic Networks and Frames

- The distinction between frames and semantic networks is actually rather blurred.
- Frames were originally derived from semantic networks and are therefore part of structure-based knowledge representations.
- The more structure a system has, the more likely it is to be termed a frame system rather than a semantic network.
- The simplest type of frame with the ideas of inheritance and default values is a semantic network.

Semantic Networks and Frames



Baseball Player
is-a: Adult Male
batting average: .252
bats: equal to handed
team:
:

Fielder
is-a: Baseball player
batting average: .262

Pee-Wee-Reese
instance: Baseball player
team: Brooklyn Dodgers

Object-oriented languages and Frames

- Frame languages have a significant overlap with object-oriented languages.

Frame terminology	**OO terminology**
Frame	Object class
Slot	Object property or attribute
Method	Method

Slots as Fully-Fledged Objects

- Frame based representations allows the **descriptive slot fillers**. The frame system interpreter will know how to process such frames.
- Frames become much more powerful when their slots contains instructions **(procedures)** for computing things from information in other slots or in other frames. This type of frames are now generally referred to as Scripts.

Logic and Frames

$\forall$ x mammal(x) $\Rightarrow$ has_part(x, head)

```
MAMMAL:
    subclass:    ANIMAL
    has_part:    head
    *furry:        yes

ELEPHANT
    subclass:    MAMMAL
    has_trunk:   yes
    *colour:     grey
    *size:       large
    *furry:      no

Clyde
    instance:    ELEPHANT
    colour:      pink
    owner:       Fred
```

Logic and Frames

$\forall$ x mammal(x) $\Rightarrow$ has_part(x, head)

$\forall$ x elephant(x) $\Rightarrow$ mammal(x)

elephant(clyde)

MAMMAL:	
subclass:	ANIMAL
has_part:	head
*furry:	yes

ELEPHANT	
subclass:	MAMMAL
has_trunk:	yes
*colour:	grey
*size:	large
*furry:	no

Clyde	
instance:	ELEPHANT
colour:	pink
owner:	Fred

Logic and Frames

$\forall$ x mammal(x) $\Rightarrow$ has_part(x, head)

$\forall$ x elephant(x) $\Rightarrow$ mammal(x)

elephant(clyde)

mammal(clyde)

has_part(clyde, head)

```
MAMMAL:
    subclass:    ANIMAL
    has_part:    head
    *furry:      yes

ELEPHANT
    subclass:    MAMMAL
    has_trunk:   yes
    *colour:     grey
    *size:       large
    *furry:      no

Clyde
    instance:    ELEPHANT
    colour:      pink
    owner:       Fred
```

Conceptual Dependency (CD)

Conceptual Dependency (CD)

- Conceptual dependency theory is a model of natural language understanding used in AI.
- Conceptual dependency theory was based on two assumptions:
  - If two sentences have the same meaning, they should be represented the same, regardless of the particular words used.
  - Information implicitly stated in the sentence should be represented explicitly.

Conceptual Dependency Graph

Figure 2. Basic form of a conceptual dependency graph.

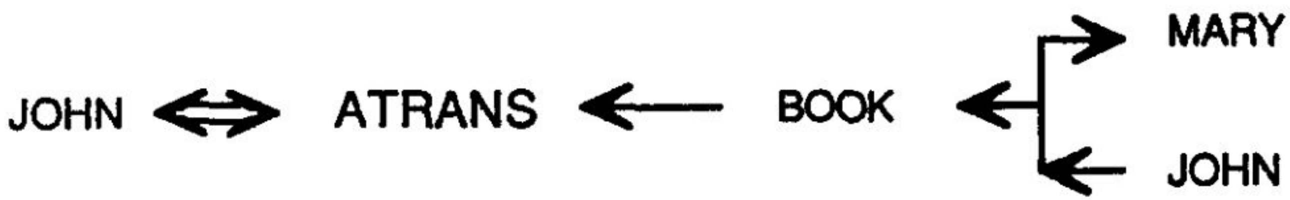


Figure 3. Representation of "John gave Mary a book."

Primitives.

Primitive conceptual categories from which dependency structure is built.

1. ACTs -- Real world actions.
2. AAs -- Attributes of Action/ Modifier of Actions (Action aiders).
3. PPs -- Real world objects (picture producers).
4. PAs -- Attributes of Objects/ Modifier of PPs (picture aiders).

Primitives

Set of Primitives ACTs.

- ATRANS - transfer a relationship (give).
- PTRANS - transfer physical location of an object (go).
- PROPEL - apply physical force to an object (push).
- MOVE - move body part by owner (kick).
- GRASP - grab an object by an actor (grasp).
- INGEST - ingest an object by an animal (eat).
- EXPEL - expel from an animal’s body (cry).

Primitives

Set of Primitives ACTs.

- MTRANS - transfer mental information (tell).
- MBUILD- mentally make new information (decide).
- SPEAK - produce sound (say).
- ATTEND - focus sense organ (listen).

Conceptual Dependency Graph

Figure 2. Basic form of a conceptual dependency graph.

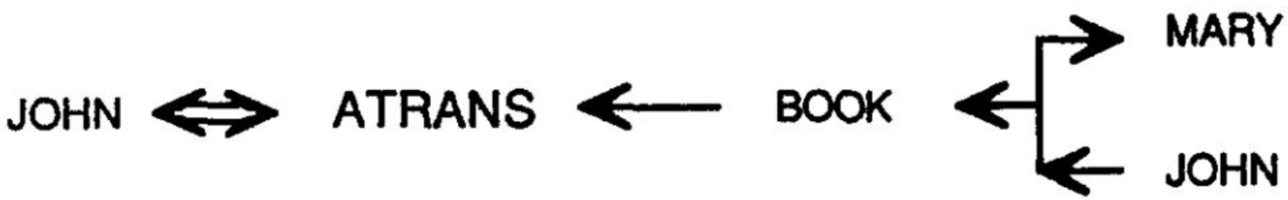
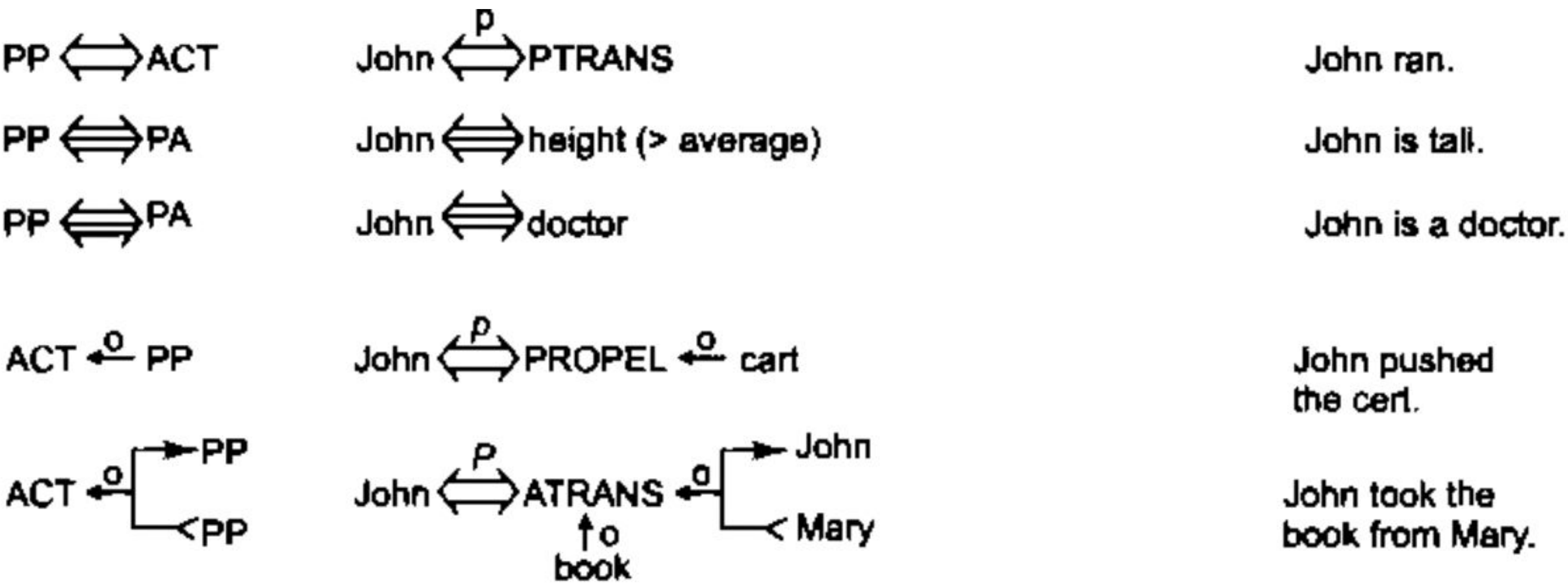


Figure 3. Representation of "John gave Mary a book."

Dependencies of CD



Motivation for Conceptual dependency

- Conceptual dependency mainly deals with representation of semantics of a language.
- CD Representation of sentences is not built using the words in the sentences rather built using conceptual primitives (independent of the language).

CD and Slots

- Each **primitive** will be associated with **a set of slots** based on the set of conceptual dependencies.
- Associated with each slot were **restrictions** as to what sorts of objects could appear in that slot.
- For example, the slots for PTRANS were the following:
  - ACTOR: a HUMAN (or animate object), that initiates the PTRANS
  - OBJECT: a PHYSICAL OBJECT, that is PTRANSed (moved)
  - FROM: a LOCATION, at which the PTRANS begins
  - TO: a LOCATION, at which the PTRANS ends.

Scripts

Scripts Plans Goals and Understanding

An Inquiry into Human Knowledge Structures

Roger Schank
Robert Abelson

Scripts.

- Scripts is a structured representation for representing **procedural** knowledge.
- Unlike frames, instead of describing an object, the script describes a sequence of events (**ordered**) in a particular context.
- Scripts organize a knowledge base in terms of the situations that the system should understand.

Script

The important components for a script are:

- **Entry conditions**: conditions to be satisfied before the event described in script can occur.
- **Results**: Conditions that are true once the script has terminated.

Script: RESTAURANT

Entry conditions:

S is hungry.
S has money.

Resulls:

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

Script

The important components for a script are:

- **Tracks**: Variations on the script. Different tracks may share components of the same script.
- **Props:** Slots that support the content of the script.
- **Roles:** Slots representing the individual participants involved in the event described in the script.

Script:	RESTAURANT
Track:	Coffee Shop
Props:	Tables
	Menu
	F = Food
	Check
	Money
Roles:	S = Customer
	W = Waiter
	C = Cook
	M = Cashier
	O = Owner

Script

The important components for a script are:

- **Scenes:** The sequence of events that occurred.

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

ATRANS - transfer a relationship (give).

PTRANS - transfer physical location of an object (go).

MOVE - move body part by owner (kick).

MTRANS - transfer mental information (tell).

MBUILD- mentally make new information (decide).

ATTEND - focus sense organ (listen).