9. Strong Slot and Filler Structures

Introduction

Individual semantic networks and frame systems may have specialized links and inference procedures, but no hard and fast rules about what kinds of objects and links are good in general for knowledge representation.

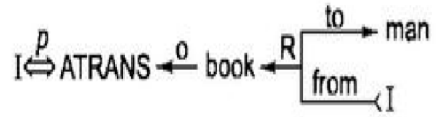
3 Structures: *Conceptual Dependency, Scripts and CYC* embody specific notions of what types of objects and relations are permitted. They stand for powerful theories of how AI programs can represent and use knowledge about common situations.

CONCEPTUAL DEPENDENCY (CD)

It is the theory of how to represent the kind of knowledge about events that is usually contained in natural language sentences. The goal is to represent the knowledge in a way that Facilitates drawing

- inferences from the sentences.
- Is independent of the language in which the sentences are originally stated. CD provides a structure into which nodes representing information can be placed a specific set of primitives at a given level of granularity.

Representation of Conceptual Dependency: ***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.

In CD representation of actions are built from a set of primitive Acts.

ATRANS	Transfer of an abstract relationship (e.g., 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 actor (e.g., clutch)
INGEST	Ingestion 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	Production of sounds (e.g., say)
ATTEND	Focusing of a sense organ toward a. stimulus (e.g., listen)

A second set of CD building blocks is the set of allowable dependencies among the conceptualizations described in a sentence. There are 4 primitive conceptual categories from which dependency structures can be built.

ACTs	Actions
PPs	Objects (picture producers)
AAs	Modifiers of actions (action aiders)
PAs	Modifiers of PPs (picture aiders)

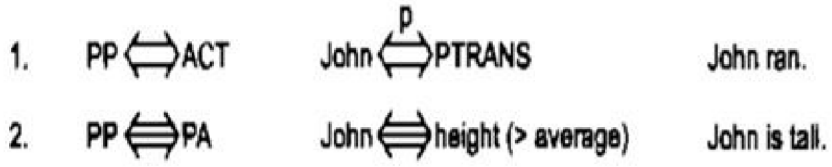
In addition, dependency structures are themselves conceptualizations and can serve as components of larger dependency structures.

The dependencies among conceptualizations correspond to semantic relations among the underlying concepts. The **first column** contains the rules; the **second** contains examples of use and the **third** theircontains an English version of each example. The rules are interpreted as follows:

- **Rule 1**

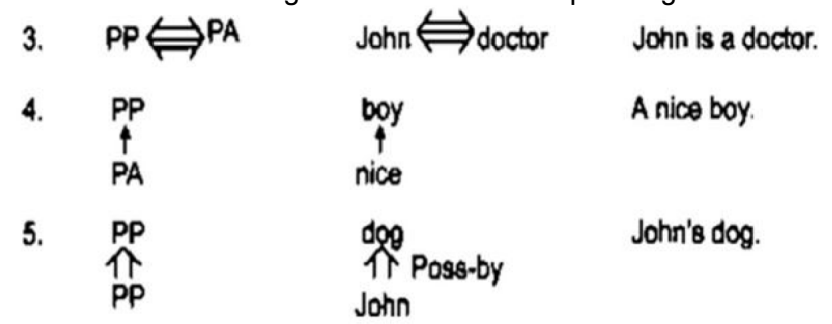
describes the relationship between an actor and the event he or she causes. This is a two-way dependency since neither actor nor event can be considered primary. The letter p above the dependency link indicates past tense.

- **Rule 2** describes the relationship between a PP and a PA that is being asserted to describe it. Many state descriptions, such as height, are represented in CD as numeric scales.



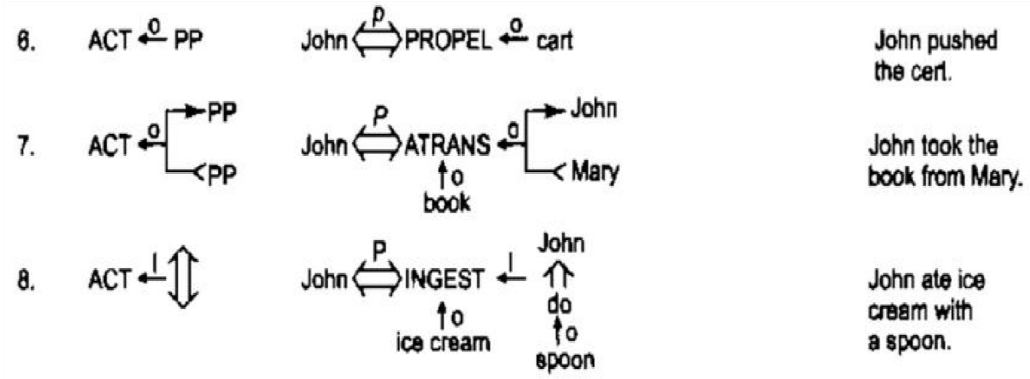
- **Rule 3** describes the relationship between two PPs, one of which belongs to the set defined by the other.
- **Rule 4** describes the relationship between a PP and an attribute that has already been predicated of it. The direction of the arrow is toward the PP being described.
- **Rule 5** describes the relationship between two PPs, 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.

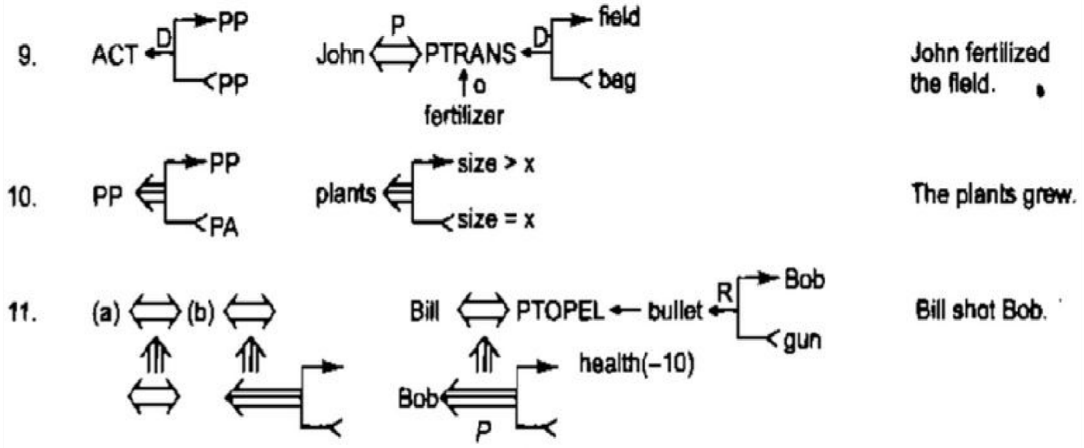


- **Rule 6** describes the relationship between an ACT and the PP that is the object of that ACT. The direction of the arrow is toward the ACT since the context of the specific ACT determines the meaning of the object relation.
- **Rule 7** describes the relationship between an ACT and the source and the recipient of the ACT.
- **Rule 8** describes the relationship between an ACT and the instrument with which it is performed. The instrument must always be a full conceptualization (i.e., it must contain

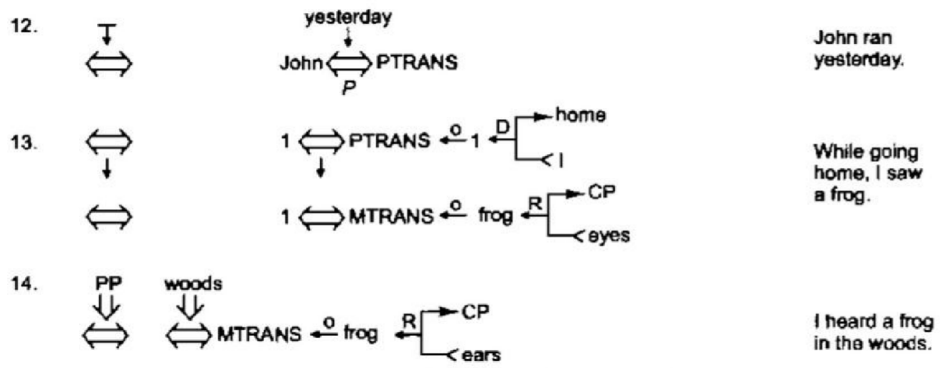
an ACT), not just a single physical object.



- **Rule 9** describes the relationship between an ACT and its physical source and destination.
- **Rule 10** represents the relationship between a PP and a state in which it started and another in which it ended.
- **Rule 11** describes the relationship between one conceptualization and another that causes it. Notice that the arrows indicate dependency of one conceptualization on another and so point in the opposite direction of the implication arrows. The two forms of the rule describe the cause of an action and the cause of a state change.



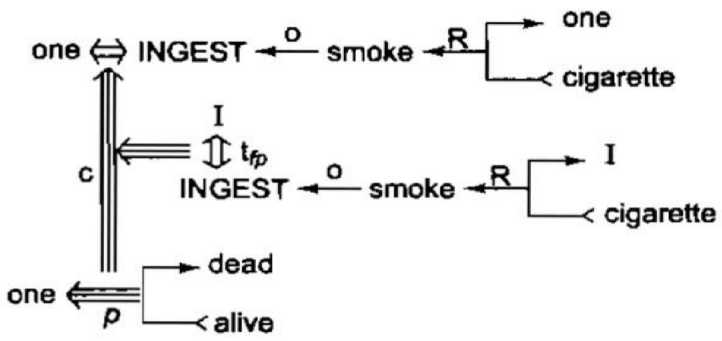
- **Rule 12** describes the relationship between a conceptualization and the time at which the event it describes occurred.
- **Rule 13** describes the relationship between one conceptualization and another that is the time of the first. The example for this rule also shows how CD exploits a model of the human information processing system.
- **Rule 14** describes relationship between a conceptualization and the place at which it occurred.



Conceptualizations representing events can be modified in a variety of ways to supply the information normally indicated in language by the tense, mood or aspect of a verb form. The set of conceptual tenses includes

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

Example: ***Since smoking can kill you, I stopped.***



Using Conceptual Tenses

Advantages of CD:

- Using these primitives involves fewer inference rules.
- Many inference rules are already represented in CD structure.
- The holes in the initial structure help to focus on the points still to be established.

Disadvantages of CD:

- Knowledge must be decomposed into fairly low level primitives.
- Impossible or difficult to find correct set of primitives.
- A lot of inference may still be required.

- Representations can be complex even for relatively simple actions. Consider:
  Dave bet Frank five pounds that Wales would win the Rugby World Cup.
  Complex representations require a lot of storage

Applications of CD:

- **MARGIE** (*Meaning Analysis, Response Generation and Inference on English*) -- model natural language understanding.
- **SAM** (*Script Applier Mechanism*) -- Scripts to understand stories. See next section.
- **PAM** (*Plan Applier Mechanism*) -- Scripts to understand stories.

SCRIPTS

We present a mechanism for representing knowledge about common sequences of events.
A **script** is a structure that describes stereotyped sequence of events in a particular event. A script consists of a set of slots. Associated with each slot may be some information about what kinds of values it may contain as well as a default value to be used if no other information is available.

A script is a structure that prescribes a set of circumstances which could be expected to follow on from one another. It is similar to a thought sequence or a chain of situations which could be anticipated. It could be considered to consist of a number of slots or frames but with more specialized roles.

Scripts provide an ability for default reasoning when information is not available that directly states that an action occurred. So we may assume, unless otherwise stated, that a diner at a restaurant was served food, that the diner paid for the food, and that the dinner was served by a waiter/waitress.

Scripts are beneficial because:

- Events tend to occur in known runs or patterns.
- Causal relationships between events exist.
- Entry conditions exist which allow an event to take place
- Prerequisites exist upon events taking place. E.g. when a student progresses through a degree scheme or when a purchaser buys a house.

The important components of the script are:

Entry Conditions	these must be satisfied before events in the script can occur.
Results	Conditions that will be true after events in script occur.
Props	objects involved in events.
Roles	events.
Track	Variations on the script. Different tracks may share components of the same script.
Scenes	The sequence of events that occur. Events are represented in conceptual dependency form.

s representing
s involved in the

Scripts are useful in describing certain situations such as robbing a bank. This might involve:

- Getting a gun.
- Hold up a bank.
- Escape with the money.

- Here the ***Props*** might be
  - Gun, *G*.
  - Loot, *L*.
  - Bag, *B*
  - Getaway car, *C*.
- The ***Roles*** might be:
  - Robber, *S*.
  - Cashier, *M*.
  - Bank Manager, *O*.
  - Policeman, *P*.
- The ***Entry*** *Conditions* might be:
  - *S* is poor.
  - *S* is destitute.
- The ***Results*** might be:
  - *S* has more money.
  - *O* is angry.
  - *M* is in a state of shock.
  - *P* is shot.

Script: ROBBERY — *Track: Successful Snatch*

Props:
G = Gun,
L = Loot,
B= Bag,
C = Get away car.

Roles:
R = Robber,
M = Cashier,
O = Bank Manager,
P = Policeman.

Entry Conditions:
R is poor.
R is destitute.

Results:
R has more money.
O is angry.
M is in a state of shock.
P is shot.

Scene 1: Getting a gun

R PTRANS R into Gun Shop
R MBUILD R choice of G
R MTRANS choice.
R ATRANS buys G

(go to scene 2)

Scene 2 Holding up the bank

R PTRANS R into bank
R ATTEND eyes M, O and P
R MOVE R to M position
R GRASP G
R MOVE G to point to M
R MTRANS "Give me the money or ELSE" to M
P MTRANS "Hold it Hands Up" to R
R PROPEL shoots G
P INGEST bullet from G
M ATRANS L to M
M ATRANS L puts in bag, B
M PTRANS exit
O ATRANS raises the alarm

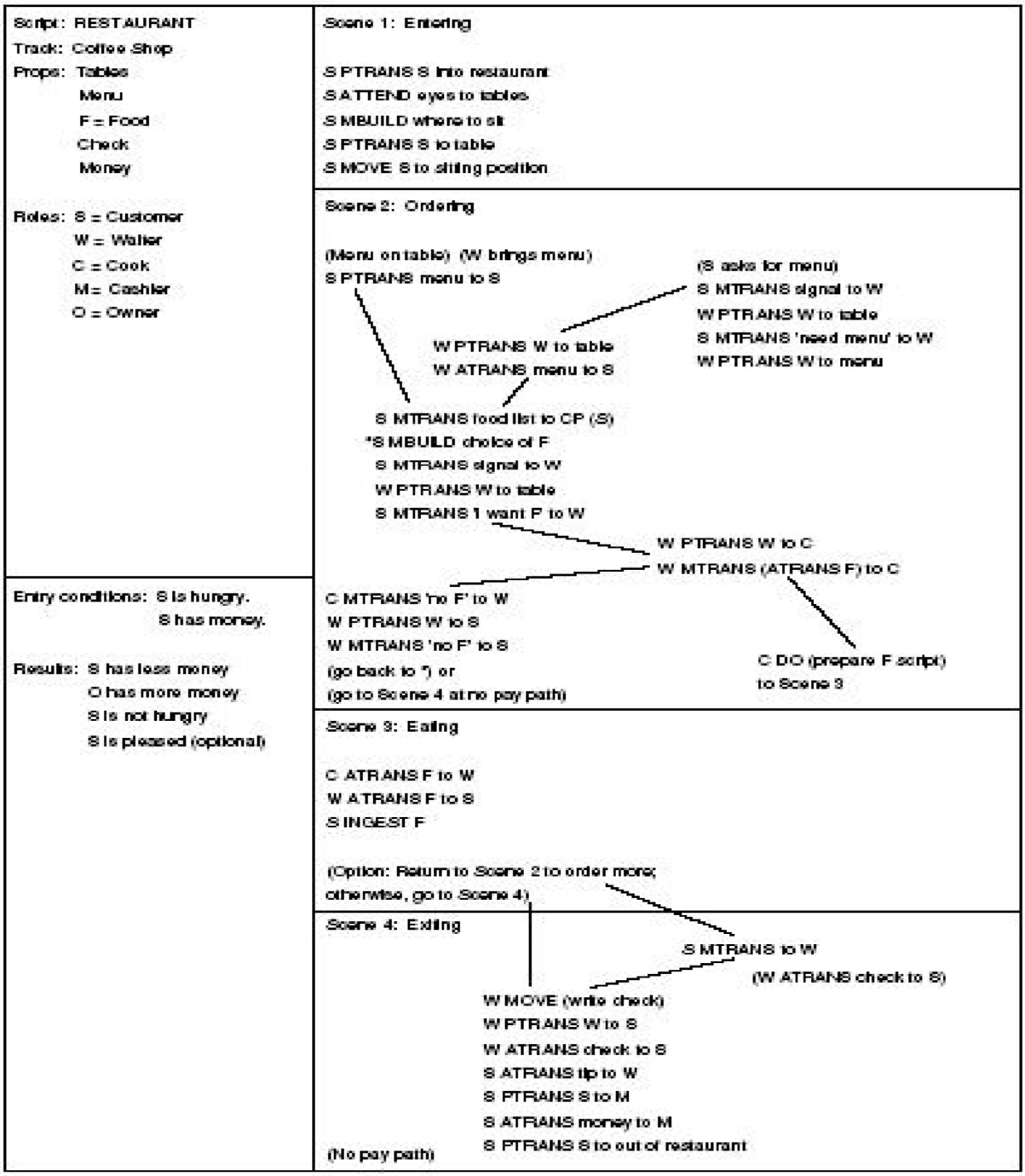
(go to scene 3)

Scene 3: The getaway

M PTRANS C

Restaurant Script

The script does not contain typical actions although there are options such as whether the customer was pleased or not. There are multiple paths through the scenes to make for a robust script what would a "going to the movies" script look like? Would it have similar props, actors, scenes? How about "going to class"?



Advantages of Scripts:

- Ability to predict events.
- A single coherent interpretation may be build up from a collection of observations.

Disadvantages:

- Less general than frames.
- May not be suitable to represent all kinds of knowledge.

CYC

CYC is a very large knowledge base project aimed at capturing human commonsense knowledge. The goal of CYC is to encode the large body of knowledge that is so obvious that it is easy to forget to state it explicitly. Such a knowledge base could then be combined with specialized knowledge bases to produce systems that are less brittle than most of the ones available today.

Why build large knowledge bases?

- **Brittleness**-- Specialized knowledge bases are *brittle*. Hard to encode new situations and non-graceful degradation in performance. Commonsense based knowledge bases should have a firmer foundation.
- **Form and Content**- Knowledge representation may not be suitable for AI. Commonsense strategies could point out where difficulties in content may affect the form.
- **Shared Knowledge** -- Should allow greater communication among systems with common bases and assumptions.

Building an immense knowledge base is a staggering task. There are two possibilities for acquiring this knowledge automatically:

1. **Machine Learning:** In order for a system to learn a great deal, it must already know a great deal. In particular, systems with a lot of knowledge will be able to employ powerful analogical reasoning.
2. **Natural Language Understanding:** Humans extend their own knowledge by reading books and talking with other humans.

How is CYC coded?

- By hand.
- Special CYCL language:
    - LISP like.
    - Frame based
    - Multiple inheritance
    - Slots are fully fledged objects.
    - Generalized inheritance -- any link not just *isa* and *instance*.

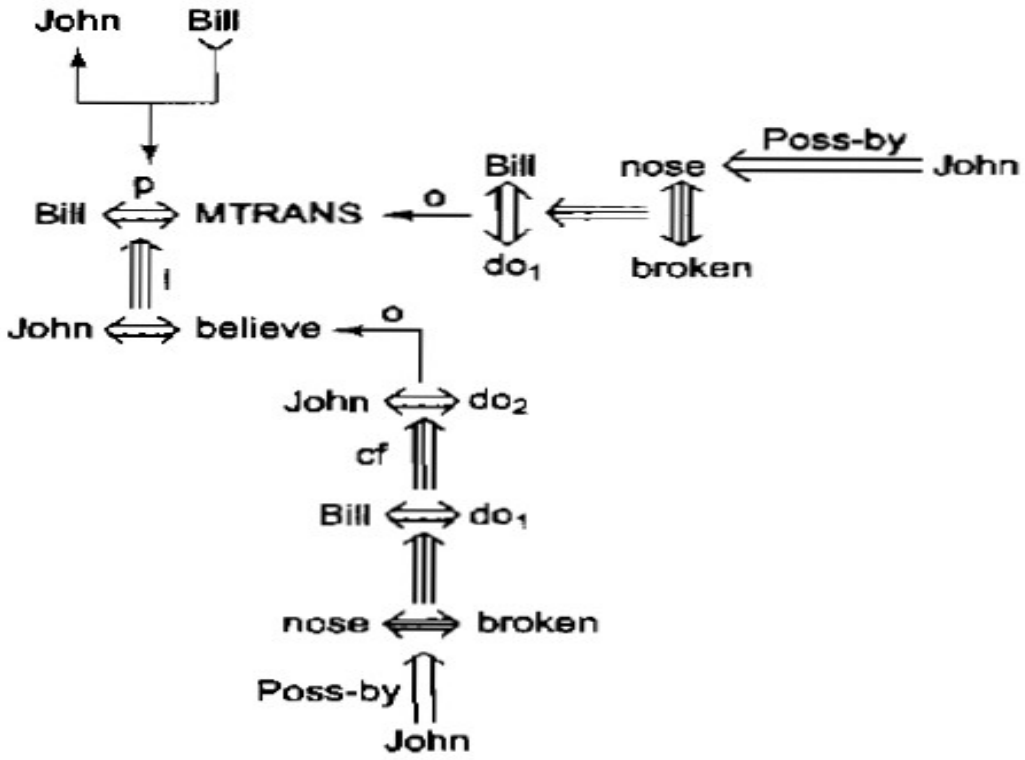
How do we do this?

- We require special implicit knowledge or commonsense such as:
    - We only die once.
    - You stay dead.
    - You cannot learn of anything when dead.
    - Time cannot go backwards.

Strong Slot and Filler Structures (Continued)

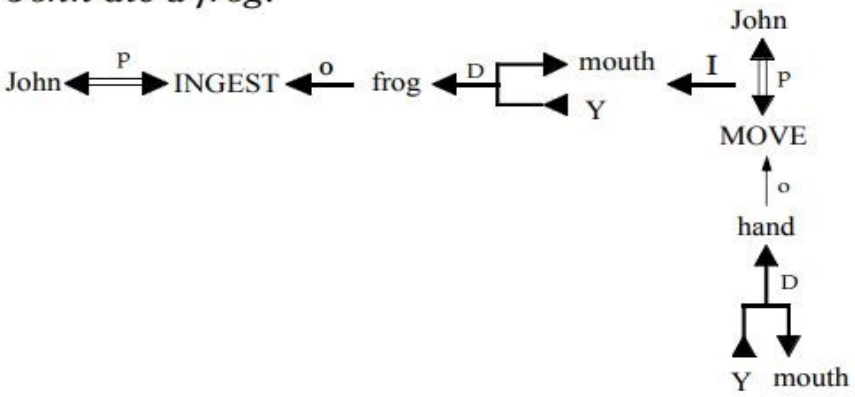
Conceptual Dependency Examples:

Consider the sentence, “Bill threatened John with a broken Nose”.

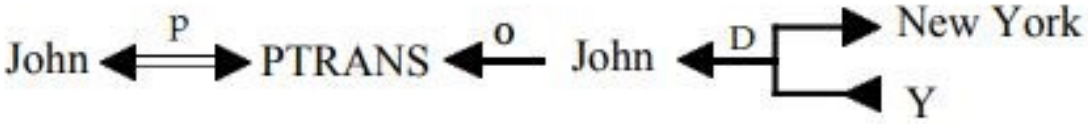


The CD representation of the information contained in the sentence is shown above. It says that Bill informed John that he (Bill) will do something to break John’s nose. Bill did this so that John will believe that if he (John) does something (different from what Bill will do to break his nose), then Bill will break John’s nose. In this representation, the word believe has been used to simplify the example. But the idea behind believe can be represented in CD as MTRANS of a fact into John’s memory. The actions do1 and do2 are dummy placeholders that refer to some as yet unspecified actions.

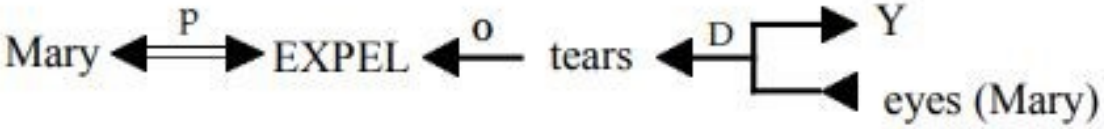
John ate a frog.



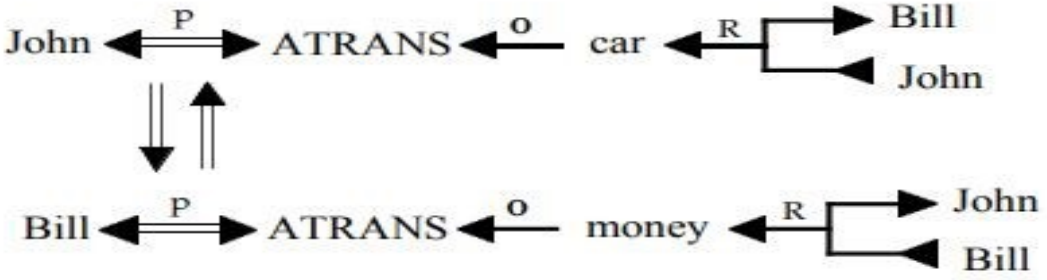
John went to New York.



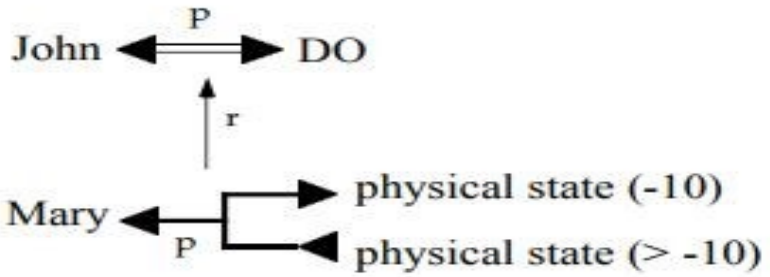
Mary cried.



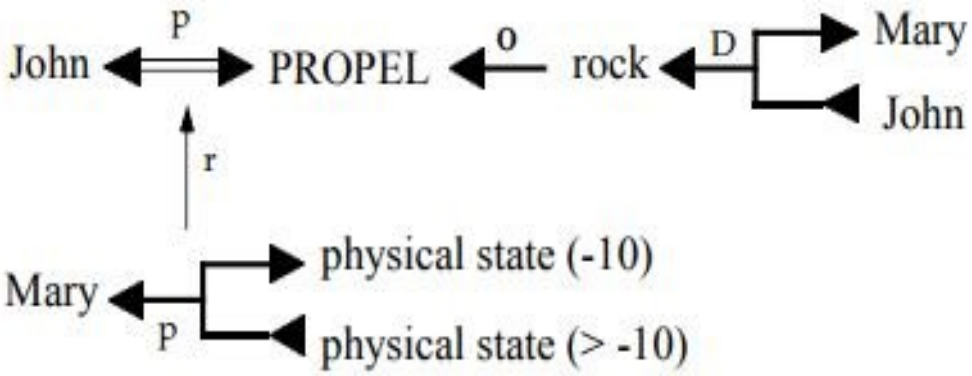
John sold his car to Bill.



John killed Mary.



John killed Mary by throwing a rock at her.



CYCL

- CYCs knowledge is encoded in a representation language called CYCL.
- CYCL is a frame based system that incorporates most of the techniques.
- Generalizes the notion of inheritance so that properties can be inherited along any link, not just isa and instance.
- CYCL contains a constraint language that allows the expression of arbitrary first-order logical expressions.

```
Mary
  likes:                    ???
  constraints:              (LispConstraint)
LispConstraint
  slotConstrained:          (likes)
  slotValueSubsumes:
    (TheSetOf X (Person allInstances)
       (And (programsIn X LispLanguage)
          (Not (ThereExists Y (Languages allInstances)
             (And (Not (Equal Y LispLanguage))
                (programsIn X Y)))))))
  propagationDirection: forward
Bob
  programsIn:               (LispLanguage)
Jane
  programsIn:               (LispLanguage CLanguage)
```

Frames and Constraint Expressions in CYC