In [1]:

Categories and Objects

The organization of objects into categories is a vital part of knowledge representation. Although interaction with the world takes place at the level of individual objects, much reasoning takes place at the level of categories.

There are two choices for representing categories in first-order logic: predicates and objects.

> We can use the predicate Basketball(b), or we can reify the category as an object, Basketballs. We could then say `Member(b,Basketballs)`, which we will abbreviate as `b ∈ Basketballs`, to say that b is a member of the category of basketballs.

We say `Subset(Basketballs,Balls)`, abbreviated as `Basketballs ⊂ Balls`, to say that Basketballs is a subcategory of Balls. We will use **subcategory, subclass, and subset** interchangeably.

Subcategory organize knowledge through **inheritance**. If we say that all instances of the category Food are edible, and if we assert that Fruit is a subclass of Food and Apples is a subclass of Fruit, then we can infer that every apple is edible. We say that the individual apples inherit the property of edibility, in this case from their membership in the Food category.

Subclass relations organize categories into a taxonomic hierarchy or taxonomy.

> First-order logic makes it easy to state facts about categories. Here are some example facts:

- An object is a member of a category.
  `BB9 ∈ Basketballs`
- A category is a subclass of another category.
  `Basketballs ⊂ Balls`
- All members of a category have some properties.
  `(x ∈ Basketballs) ⇒ Spherical(x)`
- Members of a category can be recognized by some properties.
  `Orange(x) ∧ Round(x) ∧ Diameter(x)=9.5" ∧ x ∈ Balls ⇒ x ∈ Basketballs`
- A category as a whole has some properties.
  `Dogs ∈ DomesticatedSpecies`

We say that two or more categories are **disjoint** if they have no members in common. We say that the classes undergrad and graduate student form an **exhaustive decomposition** of university students. A exhaustive decomposition of disjoint sets is known as a **partition**.

> Examples of these three concepts:

```
Disjoint({Animals,Vegetablesg})
ExhaustiveDecomposition({Americans,Canadians,Mexicans}, NorthAmericans)
Partition({Animals, Plants, Fungi, Protista, Monera}, LivingThings)
```

(***Note that the ExhaustiveDecomposition of NorthAmericans is not a Partition, because some people have dual citizenship.***)

Categories can also be defined by providing necessary and sufficient conditions for membership.

> For example, a bachelor is an unmarried adult male:

$x \in Bachelors \Leftrightarrow Unmarried(x) \land x \in Adults \land x \in Males$.

Physical composition

We use the general `PartOf` relation to say that one thing is part of another. Objects can be grouped into PartOf hierarchies, similar to Subset hierarchy:

```
PartOf(Bucharest,Romania)
PartOf(Romania,EasternEurope)
PartOf(EasternEurope,Europe)
PartOf(Europe,Earth)
```

The PartOf relation is transitive and reflexive; that is,

$PartOf(x,y) \land PartOf(y,z) \Rightarrow PartOf(x,z)$
$PartOf(x,x)$

Categories of composite objects are often characterized by structural relations among parts.

> For example, a biped is an object with exactly two legs attached to a body:

$Biped(a) \Rightarrow \exists\ l1,l2,b\ Leg(l1) \land Leg(l2) \land Body(b) \land PartOf(l1,a) \land PartOf(l2,a) \land PartOf(b,a) \land Attached(l1,b) \land Attached(l2,b) \land l1 \neq l2 \land [\forall\ l3\ Leg(l3) \land PartOf(l3,a) \Rightarrow (l3=l1 \lor l3=l2)]$

Measures

```
Diameter(Basketball12)=Inches(9.5)                    ListPrice(Basketball12)=$(19)
Weight(BunchOf({Apple1,Apple2,Apple3})) = Pounds(2)
```
$d \in Days \Rightarrow Duration(d)=Hours(24)$

Intrinsic Vs Extrinsic properties

Any part of a butter-object is also a butter-object: $b \in Butter\ wedge\ PartOf(p,b) \Rightarrow p \in Butter$ Butter melts at around 30 degrees centigrade: $b \in Butter \Rightarrow MeltingPoint(b,Centigrade(30))$

Some properties are intrinsic: they belong to the very substance of the object, rather than to the object as a whole.

When you cut an instance of stuff in half, the two pieces retain the intrinsic properties—things like density, boiling point, flavor, Extrinsic color, ownership, and so on.

On the other hand, their extrinsic properties—weight, length, shape, and so on—are not retained under subdivision.

A category of objects that includes in its definition only intrinsic properties is then a **substance**, or **mass noun**; a class that includes any extrinsic properties in its definition is a **count noun**.

Stuff and **Thing** are the most general substance and object categories, respectively.

Events

Actions: things that happen, such as Shoot; and fluents: aspects of the world that change, such as HaveArrow come under the category Events.

Example predicates	from the event calculus
T(f,t1,t2)	Fluent f is true for all times between t1 and t2
Happens(e,t1,t2)	Event e starts at time t1 and ends at t2
Initiates(e,f,t)	Event e causes fluent f to become true at time t
Terminates(e,f,t)	Event e causes fluent f to cease to be true at time t
Initiated(f,t1,t2)	Fluent f become true at some point between t1 and t2
Terminated(f,t1,t2)	Fluent f cease to be true at some point between t1 and t2
t1 < t2	Time point t1 occurs before time t2

Assume an event happens between time t1 and t3, and at t2 somewhere in that time interval the event changes the value of fluent f , either initiating it (making it true) or terminating it (making it false). Then at time t4 in the future, if no other intervening event has changed the fluent (either terminated or initiated it, respectively), then the fluent will have maintained its value.

Formally, the axioms are:

$$Happens(e,t1,t3) \wedge Initiates(e,f,t2) \wedge Terminated(f, t2,t4) \wedge t1 \le t2 \le t3 \le t4 \Rightarrow T(f,t2,t4)$$

$$Happens(e,t1,t3) \wedge Terminates(e,f,t2) \wedge \neg\, Initiated(f,t2, t4) \wedge t1 \le t2 \le t3 \le t4 \Rightarrow \neg\, T(f,t2,t4)$$

where *Terminated* and *Initiated* are defined by:

$$Terminated(f,t_1,t_5) \Leftrightarrow$$
$$\exists e,t_2,t_3,t_4\ \ Happens(e,t_2,t_4) \wedge Terminates(e,f,t_3) \wedge t_1 \le t_2 \le t_3 \le t_4 \le t_5$$
$$Initiated(f,t_1,t_5) \Leftrightarrow$$
$$\exists e,t_2,t_3,t_4\ \ Happens(e,t_2,t_4) \wedge Initiates(e,f,t_3) \wedge t_1 \le t_2 \le t_3 \le t_4 \le t_5$$

Time

There are two kinds of time intervals: moments and extended intervals. The distinction is that only moments have zero duration:

```
Partition({Moments, ExtendedIntervals}, Intervals)
```

$$i \in Moments \Leftrightarrow Duration(i) = Seconds(0)$$

Semantic Networks

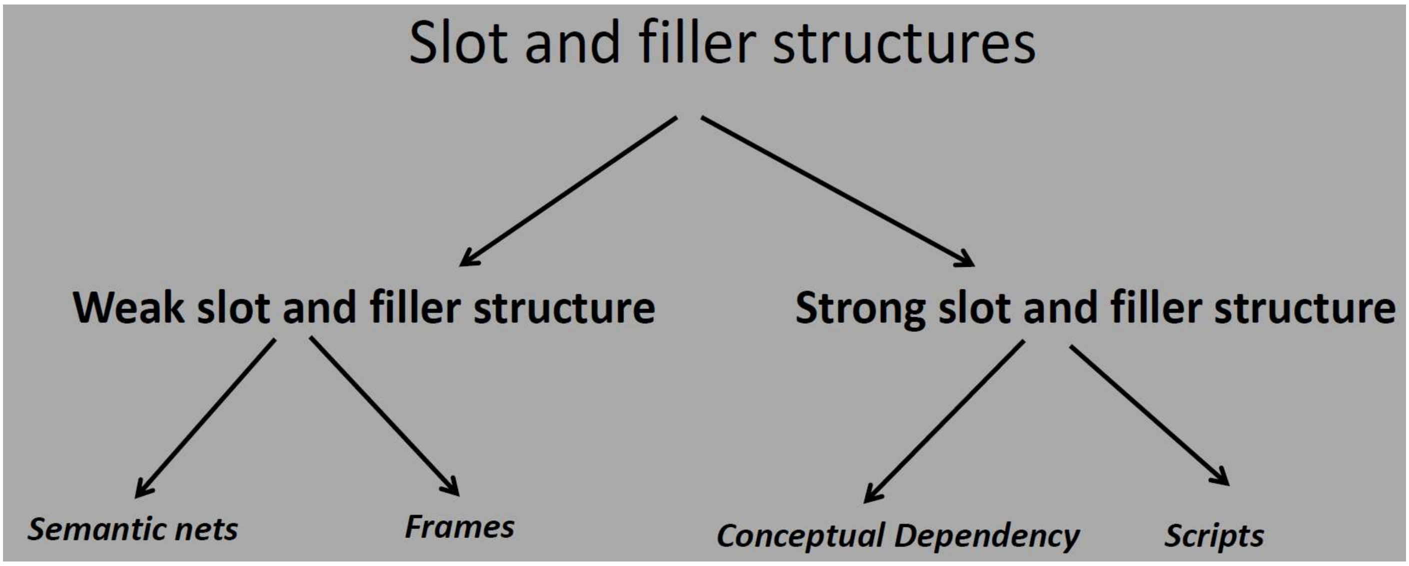
- Semantic network is a knowledge representation model which is in a form of graphical schemes consisting of nodes and links among nodes.
- Nodes in a semantic network can show concepts, objects, features, events, time.
- Links indicates the connection among nodes. The links should be labeled and directed.

where *Terminated* and *Initiated* are defined by:

$$Terminated(f,t_1,t_5) \Leftrightarrow$$
$$\exists e,t_2,t_3,t_4\ \ Happens(e,t_2,t_4) \wedge Terminates(e,f,t_3) \wedge t_1 \le t_2 \le t_3 \le t_4 \le t_5$$
$$Initiated(f,t_1,t_5) \Leftrightarrow$$
$$\exists e,t_2,t_3,t_4\ \ Happens(e,t_2,t_4) \wedge Initiates(e,f,t_3) \wedge t_1 \le t_2 \le t_3 \le t_4 \le t_5$$

Weak slot and filler structures

They are “Knowledge- Poor” or “weak” as very little importance is given to the specific knowledge the structure should contain. Here **attribute means slot and its value is called filler**.

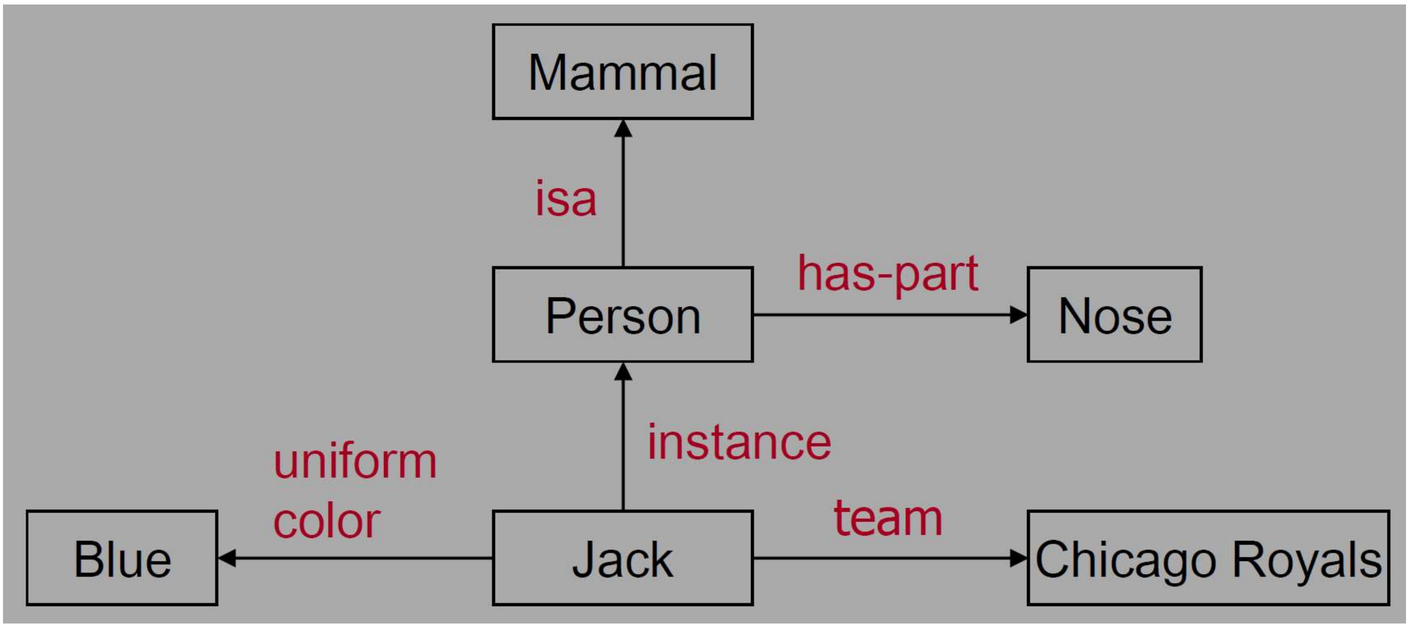


Semantic nets

In semantic nets information is represented as set of nodes connected to each other by a set of labelled arcs.

> Nodes represent: various objects / values of the attributes of object.

> Arcs represent: relationships among nodes.



Representing non binary predicates:

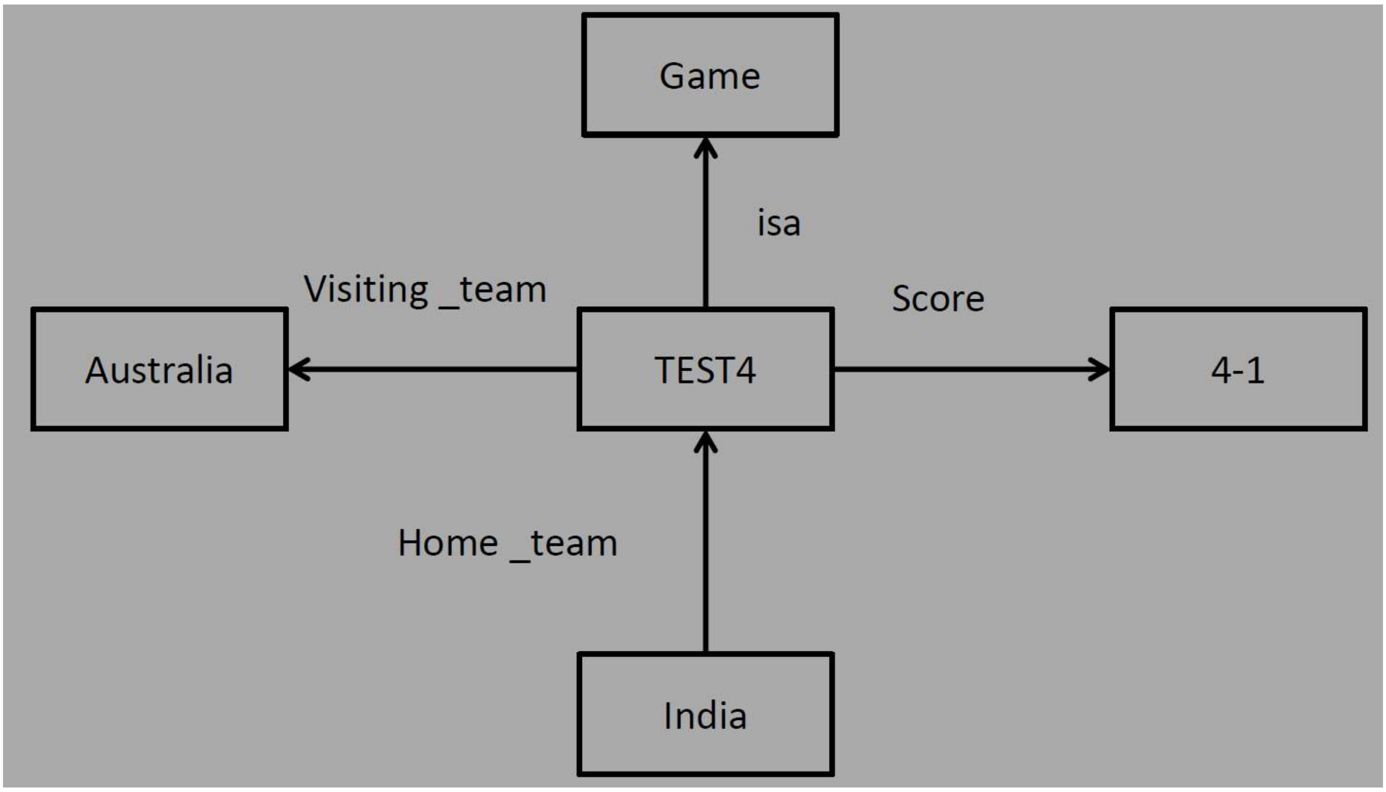
1. Unary

> `Man(marcus)` can be converted into: `instance(marcus,Man)`

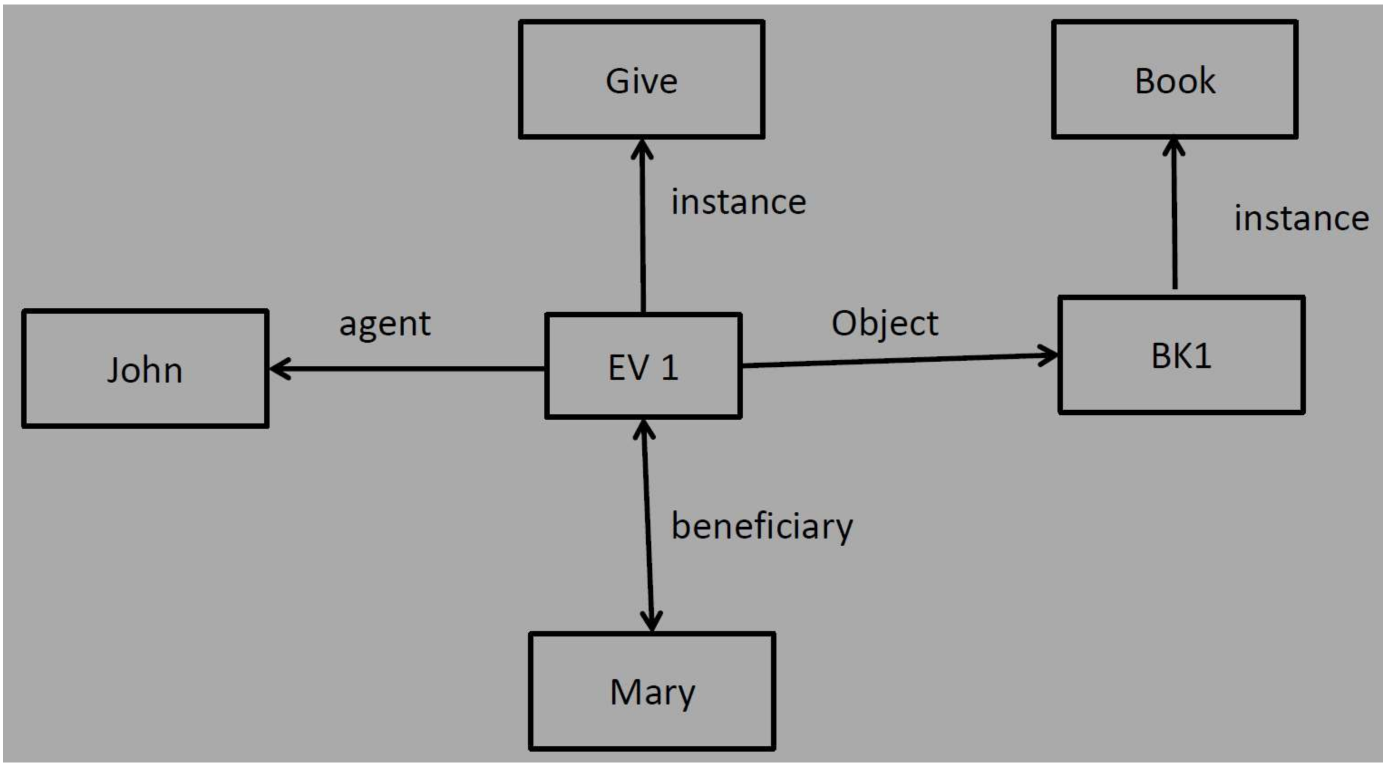
2. Other aritiese. Three or more place predicates can be converted to binary form as follows:

- Create new object representing the entire predicate.
- Introduce binary predicates to describe relation to this new object.

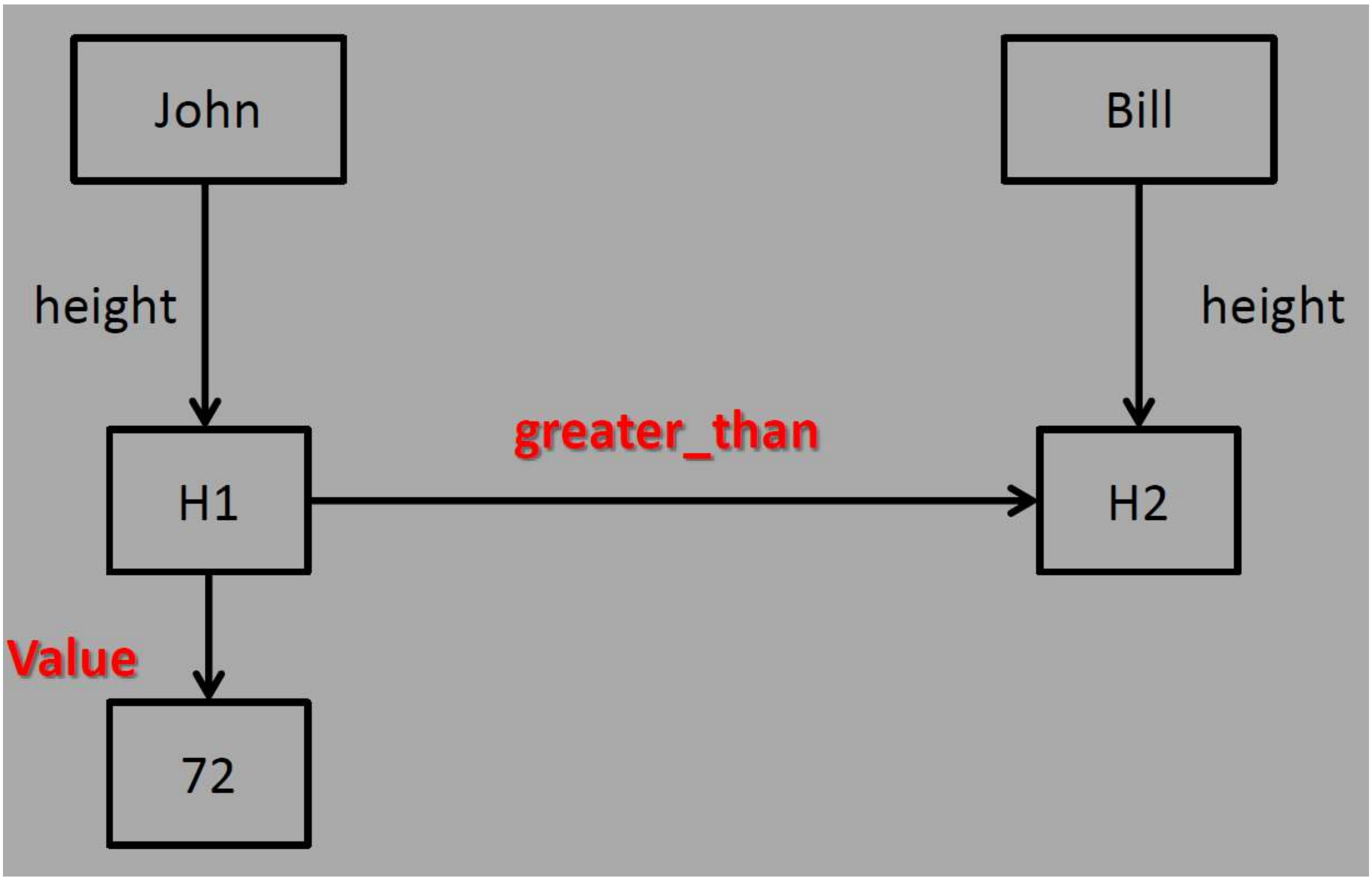
`Score(india,australia,4-1)`



John gave the book to Mary :- `give(john,mary,book)`



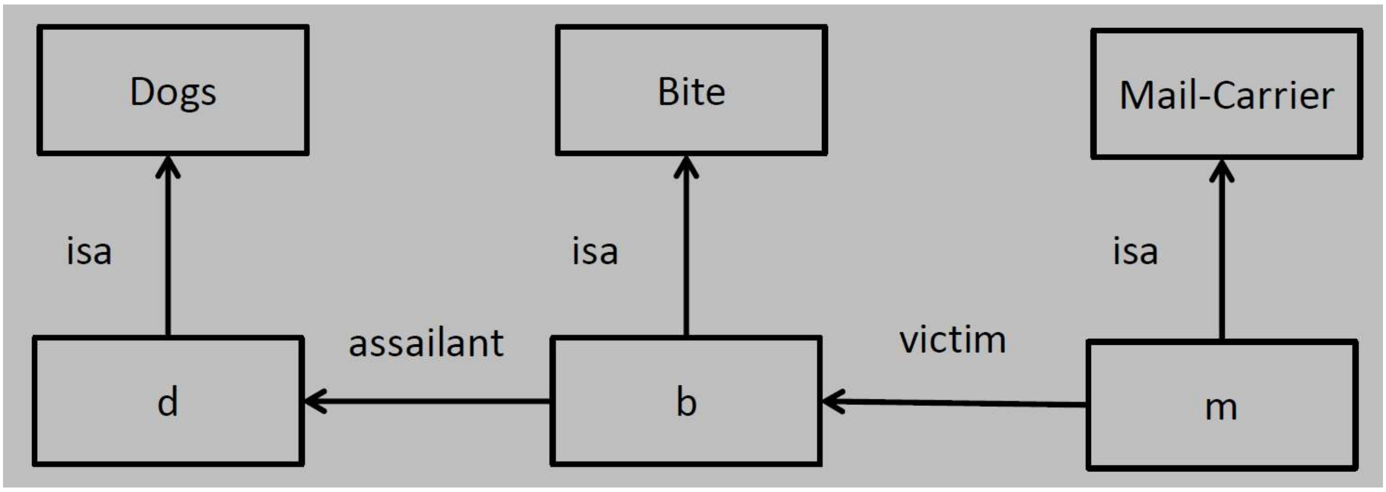
John is taller than Bill :- `taller(john,Bill)`



Partitioned semantic nets

Used to represent quantified expressions in semantic nets. One way to do this is to partition the semantic net into a hierarchical set of spaces each of which corresponds to the scope of one or more variable.

The dog bit the mail carrier (partitioning not required)



Every dog has bitten a mail carrier

$$\forall x : dog(x) \Rightarrow \exists y : mail - carrier(y) \wedge bite(x, y)$$

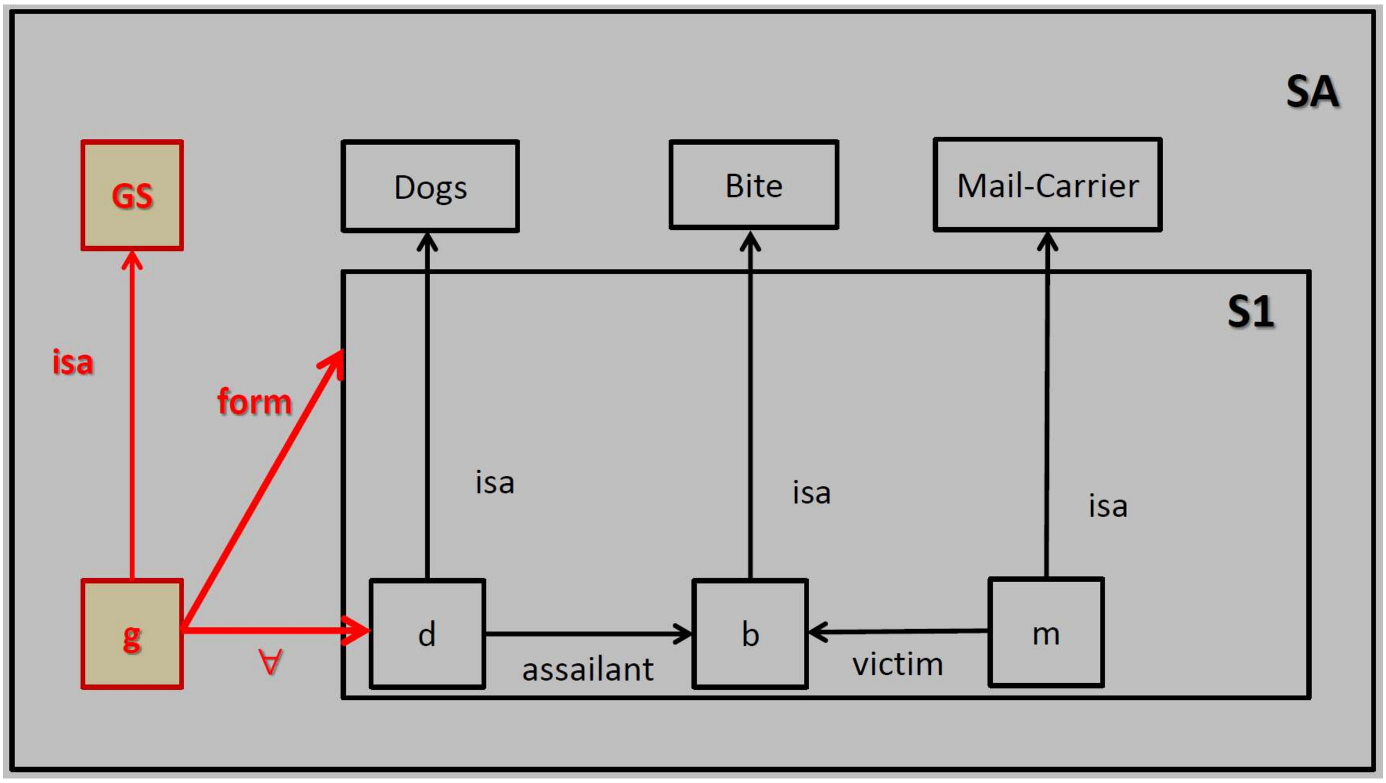
How to represent universal quantifiers?

- Let node **g** stands for assertion given above
- This node is an instance of a special class **GS** of general statements about the world.
- Every element in **GS** has 2 attributes:-

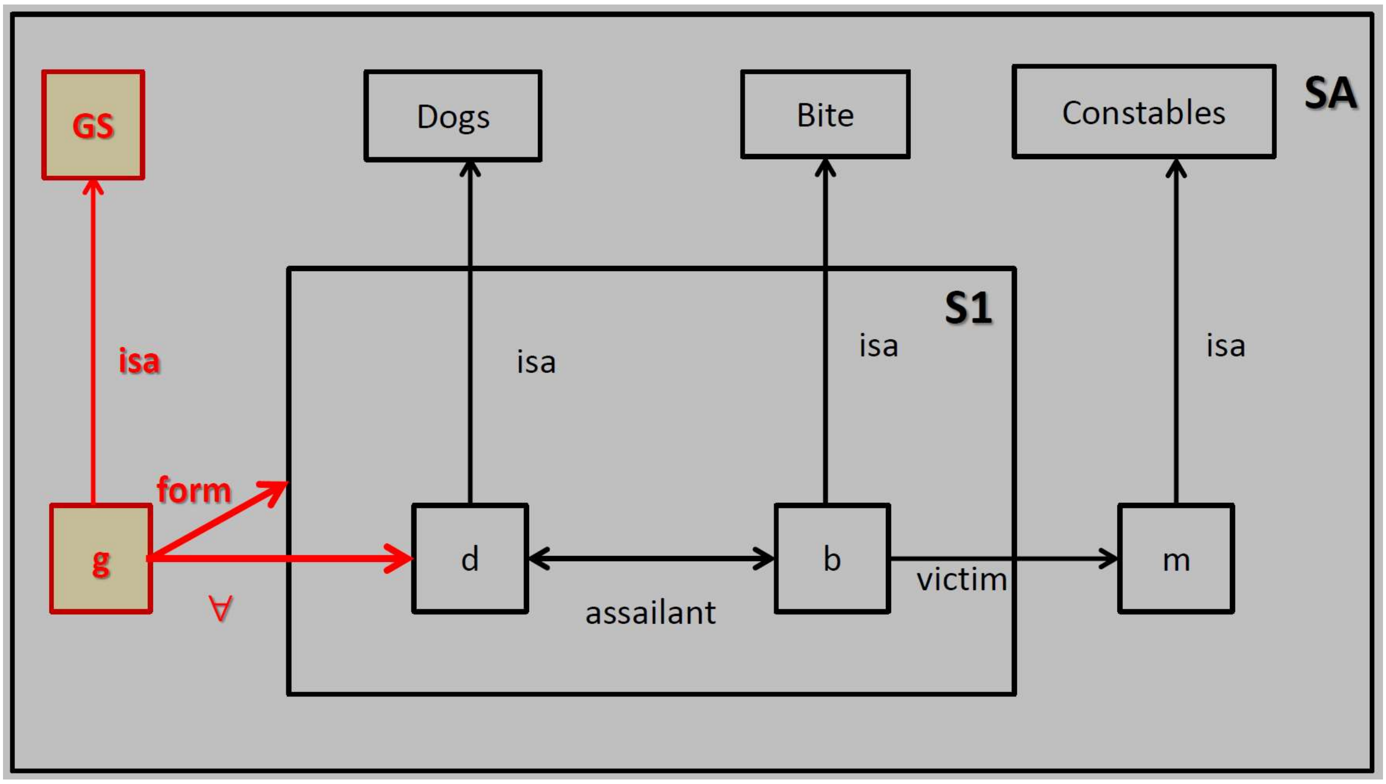
- Form - states relation that is being asserted.
- ∀ connections - one or more, one for each of the universally quantified variables.

- **SA** is the space of partitioned sementic net.

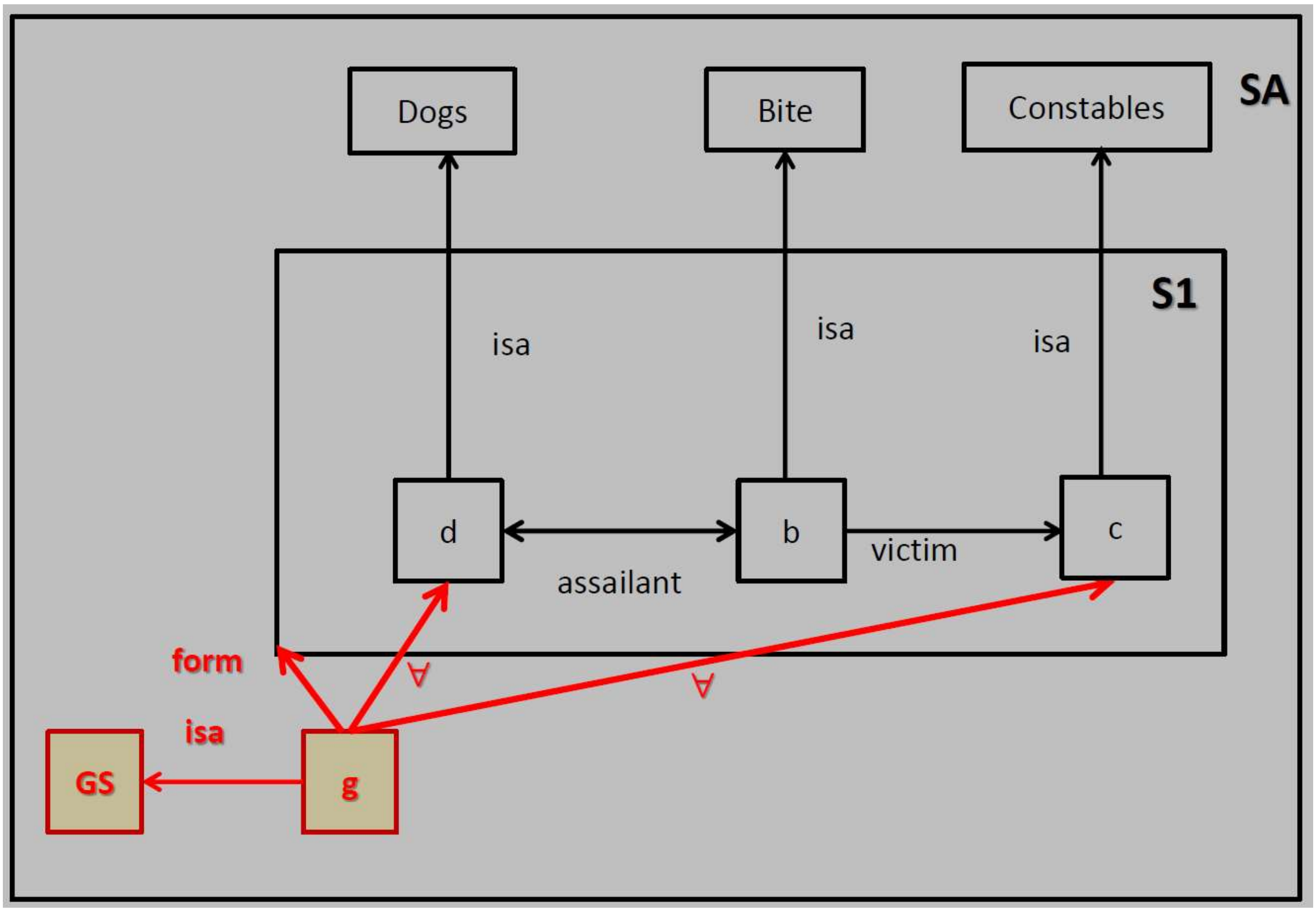
Every dog has bitten a mail carrier



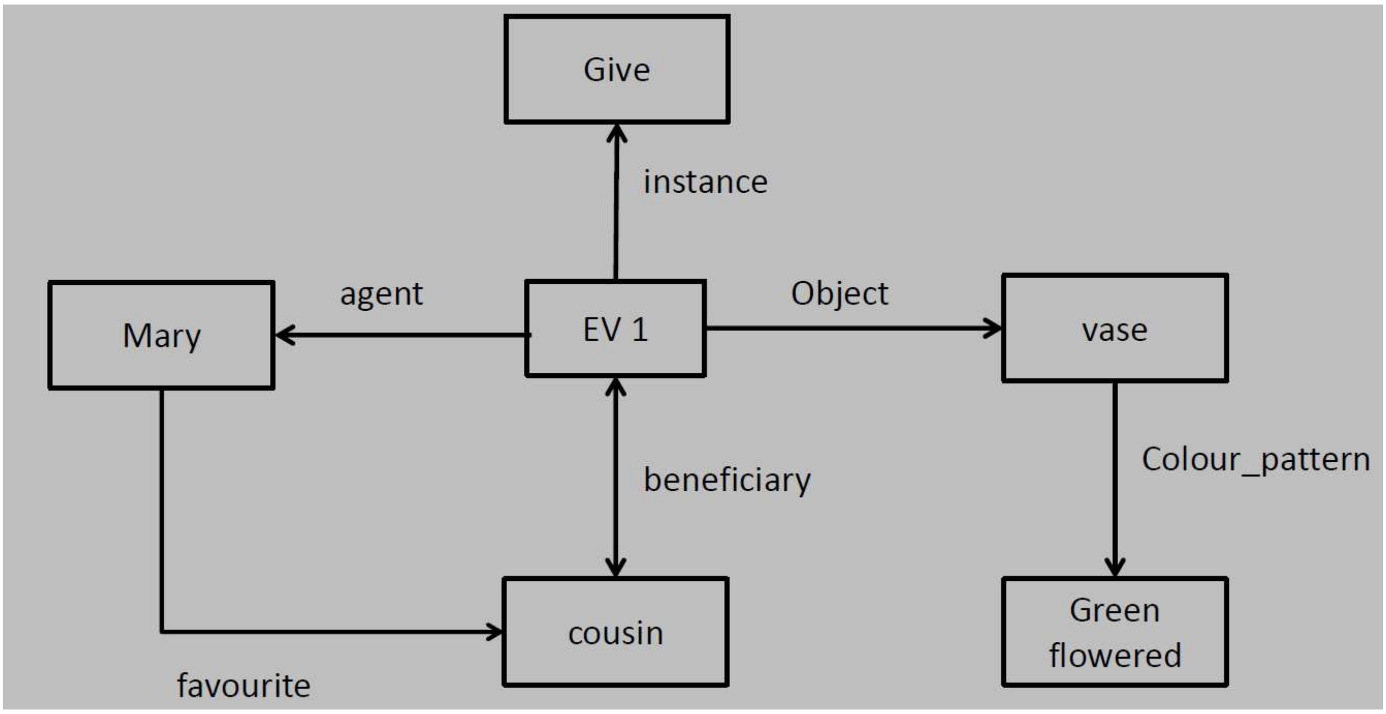
Every dog in the town has bitten the constable



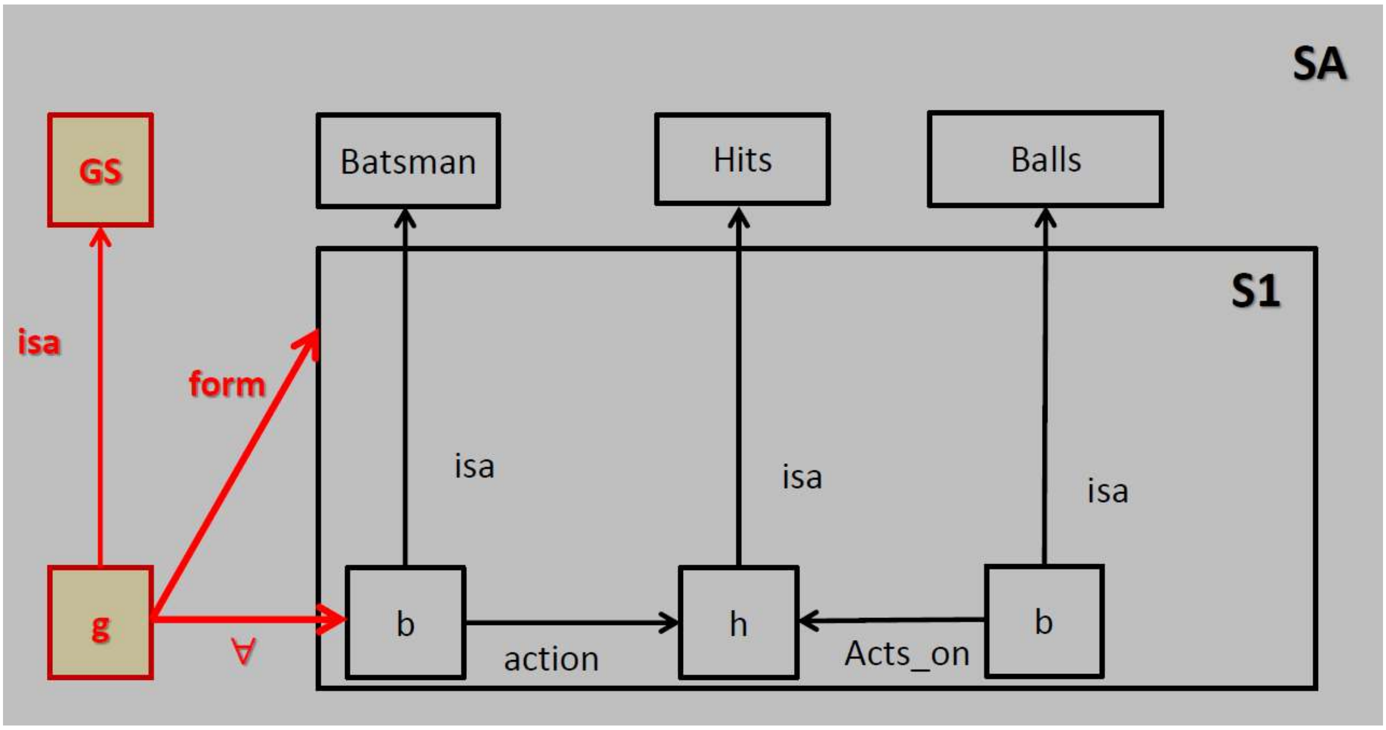
Every dog in the town has bitten every constable



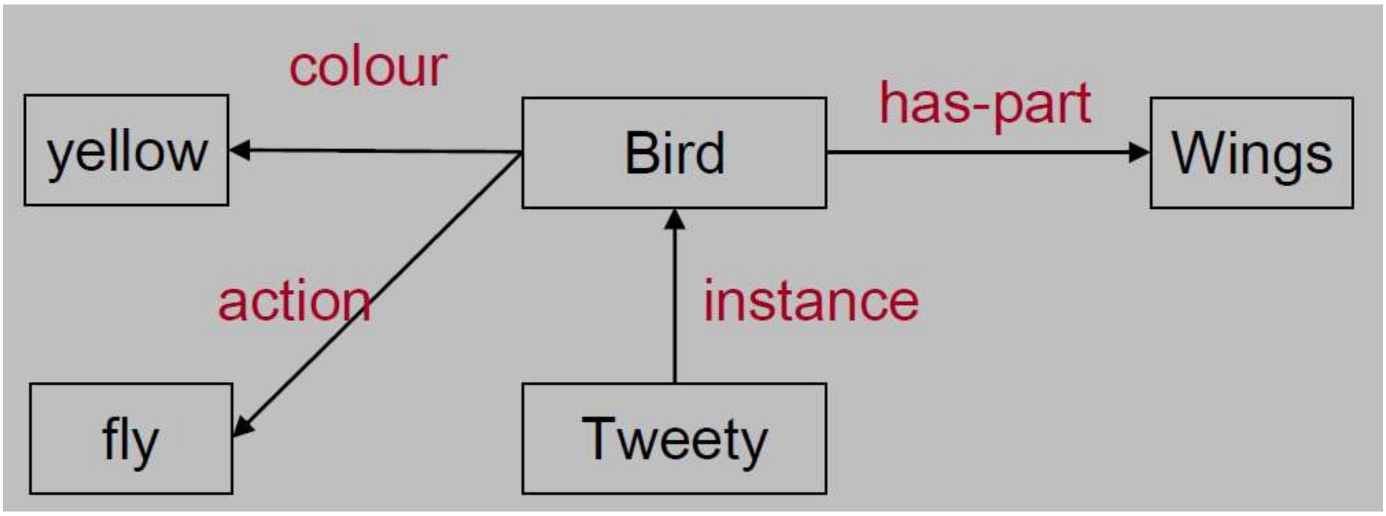
Mary gave the green flowered vase to her favourite cousin



Every batsman hits a ball



Tweety is a kind of bird who can fly. It is Yellow in colour and has wings



Frame

A frame is a collection of attributes (usually called slots) and associated values (and possibly constraints on values) that describe some entity in the world.

A single frame taken alone is rarely useful. Instead, we build frame systems out of collections of frames that are connected to each other by virtue of the fact that tbe value of an attribute of one frame may be another frame.

Each frame represents either a class (a set) or an instance (an element of a class).

The `isa` relation is used for subset relation. The set of adult males is a subset of the set of people. The set of major league baseball players is a subset of the set of adult males, and so forth.

The instance relation corresponds to the relation element-of. Pee Wee Reese is an element of the set of fielders.

Because a class represents a set, there are two kinds of attributes that can be associated with it. There are **attributes about the set itself, and there are attributes that are to be inherited by each element of the set. We indicate the difference between these two by prefixing the later with an asterisk (*)**.

> For example, consider the class ML-Baseball-Player, We have shown only two properties of it as a set: (it is a subset of the set of adult males. And it has cardinality 624 (i.e., there are 624 major league baseball players). We have listed five properties that all major league baseball players have (height, bats, batting-average, team, and uniform-color), and we have specified default values for the first three of them. By providing both kinds of slots, we allow a class both to define a set of objects and to describe a prototypical object of the set.

Person
- isa: Mammal
- cardinality: 6,000,000,000
- * Handed: right

Adult__Male
- isa: Person
- Cardinality: 2,000,000,000
- * Height: 5- 10

ML_Baseball_Player
- isa: Adult_Male
- cardinality: 624
- * height: 6-1
- * bats: equal to handed
- * batting-avg: 0.252
- * team:
- *uniform_color:

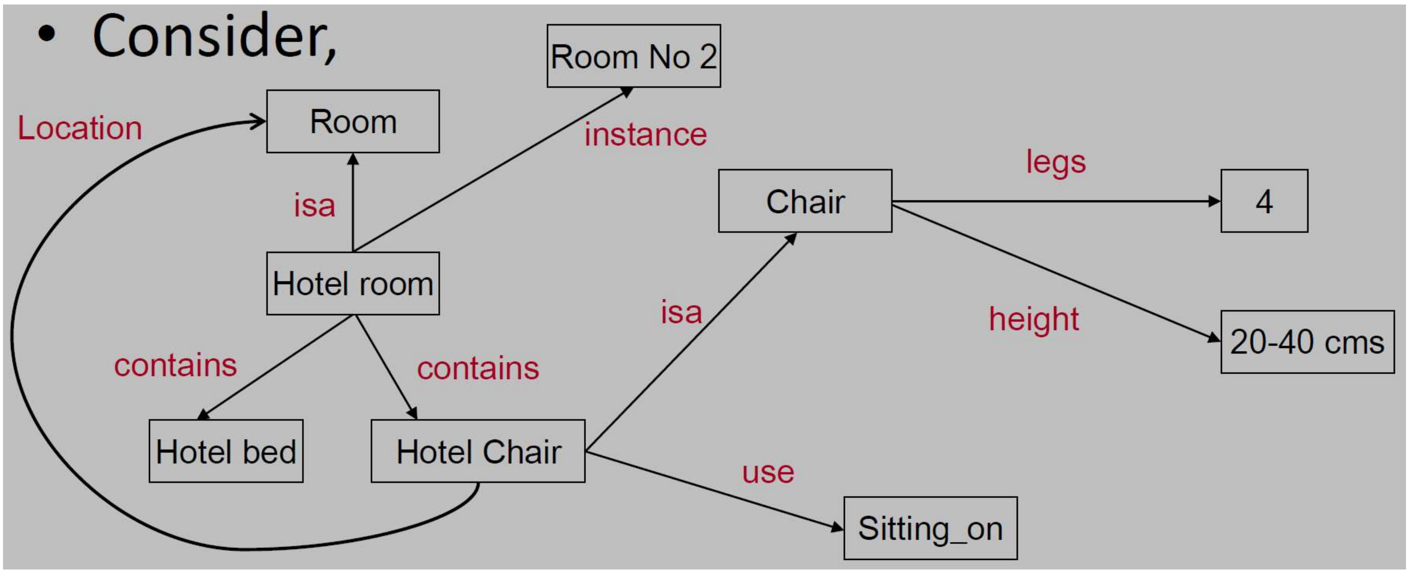
Jack_Roberts
- instance: Fielder
- height: 5-10
- balls: right
- batting_avg: 0.309
- team: Chicago cubs
- uniform_color: blue

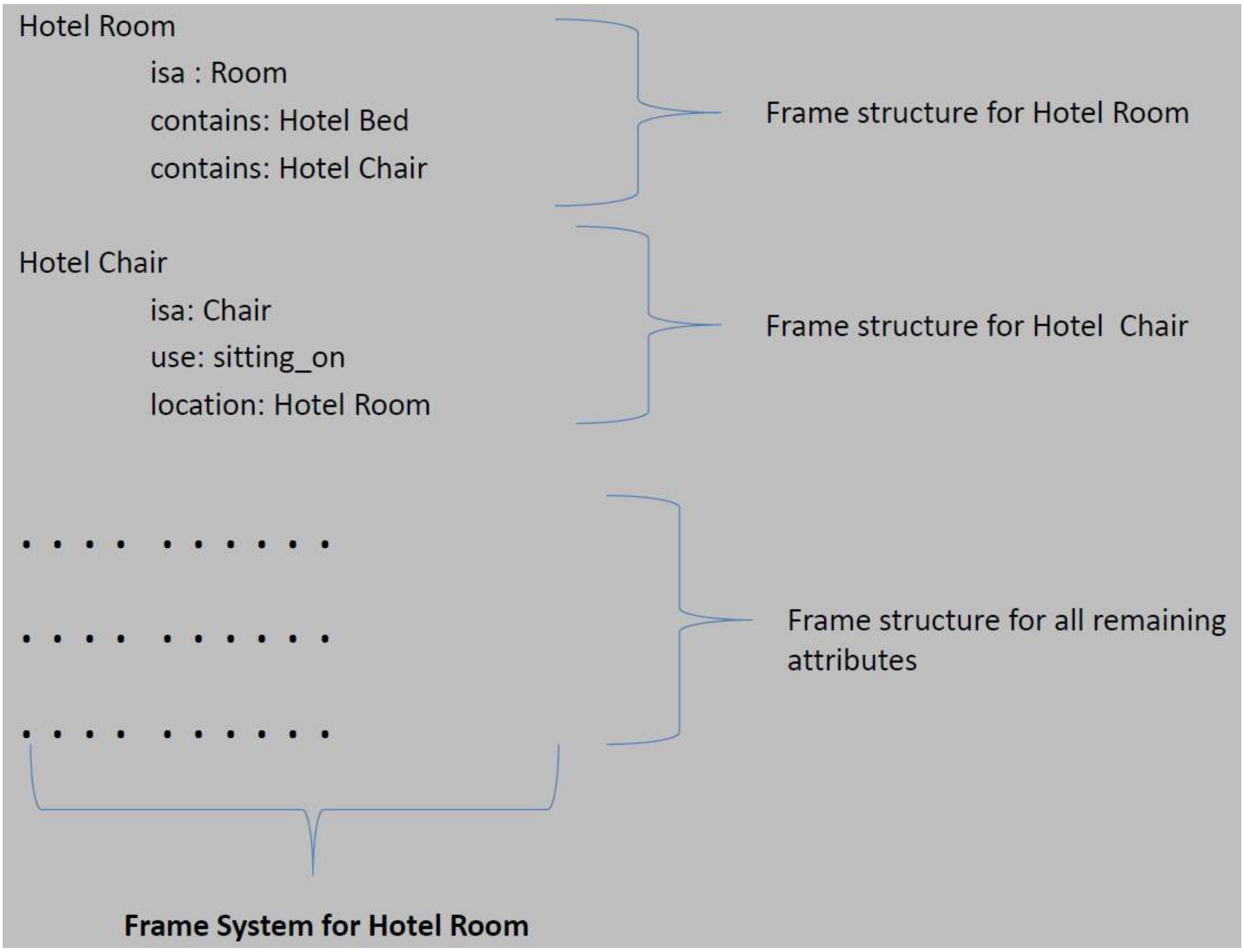
Individual frame

Fielder
- isa: ML_Baseball_Player
- cardinality: 376
- batting_avg: 0.262

ML_Baseball_Team
- isa: Team
- cardinality: 26
- team_size: 24
- manager:

Example 2





Class Vs Meta Class

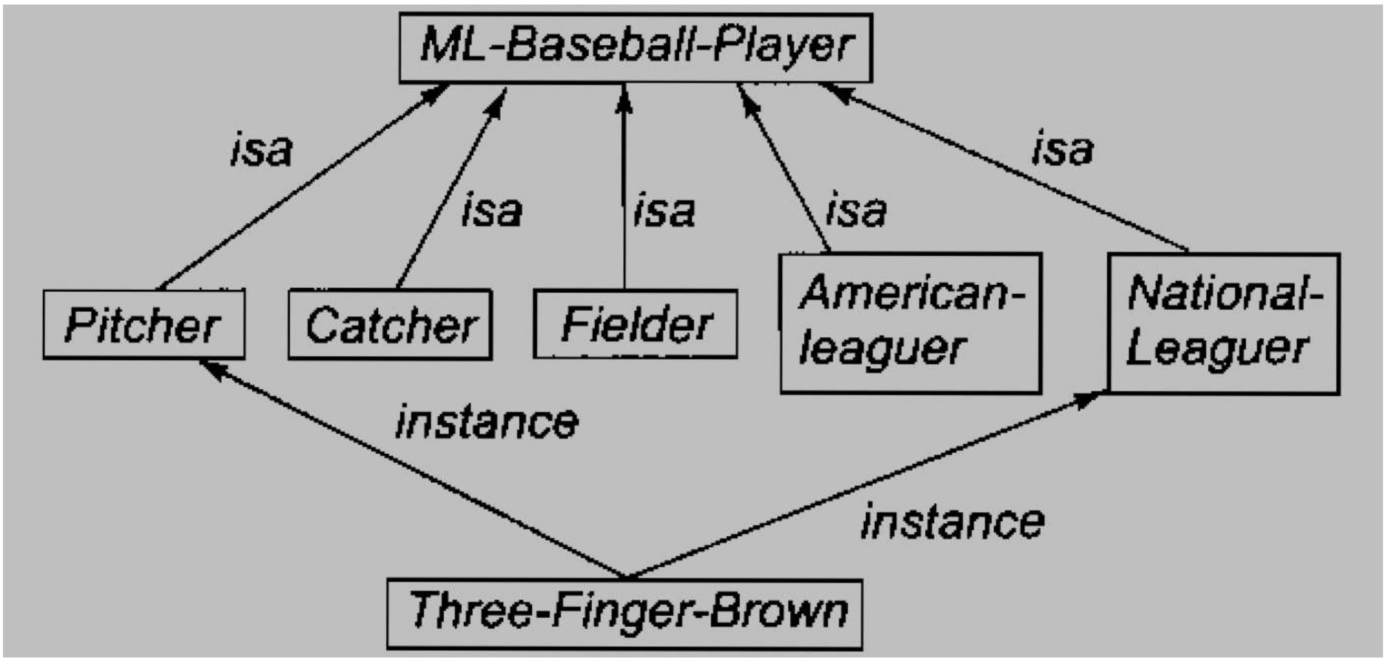
A class can be viewed as a subset (isa) of a larger class that also contains its elements and an instance (instance) of a class of sets, from which it inherits its set-level properties.

To make this distinction clear, it is useful to distinguish between regular classes, whose elements are individual entities, and meta classes, which are special classes whose elements are themselves classes.

A class is now an element of (instance) some class (or classes) as well as a subclass (isa) of one or more classes.

A class inherits properties from the class of which it is an instance, just as any instance does. In addition, a class passes inheritable properties down from its superclasses to its instances.

Is covered by: relationship is called as 'covered-by' when we have a class and it has set of subclasses, the union of which is equal to the superclass



ML_Baseball_Player

is covered-by: { Pitcher, Catcher, Fielder,American leaguer, National leaguer}

Pitcher

isa: ML_Baseball_Player

mutually_disjoint-with: {Catcher, Fielder }

Catcher

isa: ML_Baseball_Player

mutually_disjoint-with: {Pitcher, Fielder }

Fielder

isa: ML_Baseball_Player

mutually_disjoint-with: {Pitcher, Catcher}

Conceptual Dependancy

- Focuses on concepts instead of syntax.
- Focuses on understanding instead of structure.
- Assumes inference is fundamental to understanding.
- Introduced idea of a canonical meaning representation.
    - different words and structures represent the same concept.
    - language-independent meaning representation.

Canonical Meaning Representations

John gave Mary a book.

John gave a book to Mary.

Mary was given a book by John.

Mary took a book from John.

Mary received a book from John.

Conceptual Primitives

- basic meaning elements that underlie the words that we use.
- can be combined to represent complex meanings.
- an interlingual representation.
  Goal: to represent meaning so that general rules can be applied, without duplicating information.
  Theory: a small number of primitive actions can represent any sentence.

Conceptual Primitives for Actions

Eleven primitives can account for most actions in the physical world:

- ATRANS : to change an abstract relationship of a physical object.
- ATTEND : to direct a sense organ or focus an organ towards a stimulus.
- INGEST : to take something inside an animate object.
- EXPEL : to take something from inside an animate object and force it out.
- GRASP : to physically grasp an object
- MBUILD : to create or combine thoughts.
- MTRANS : to transfer information mentally.
- MOVE : to move a body part
- PROPEL : to apply a force to
- PTRANS : to change the location of a physical object.
- SPEAK : to produce a sound.

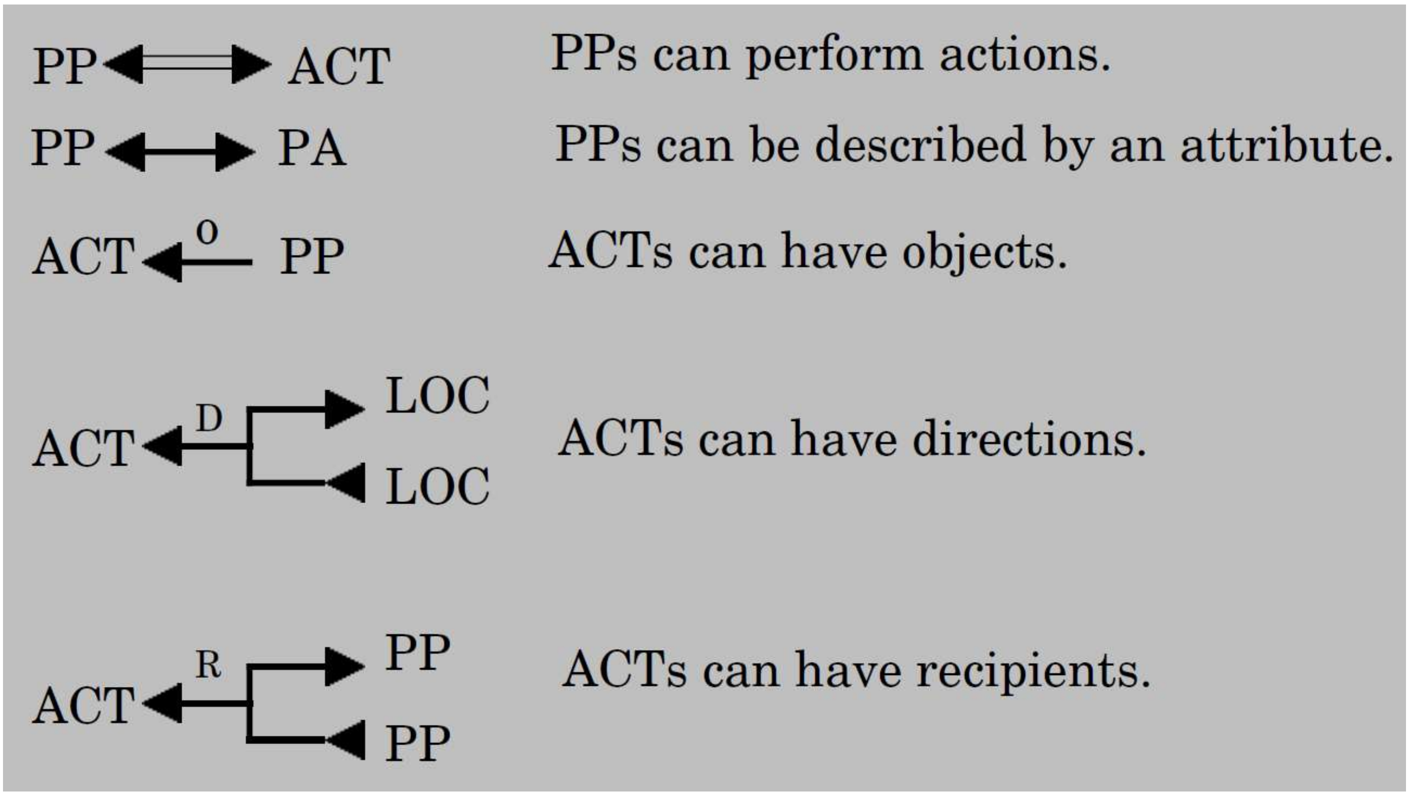
Conceptual Categories

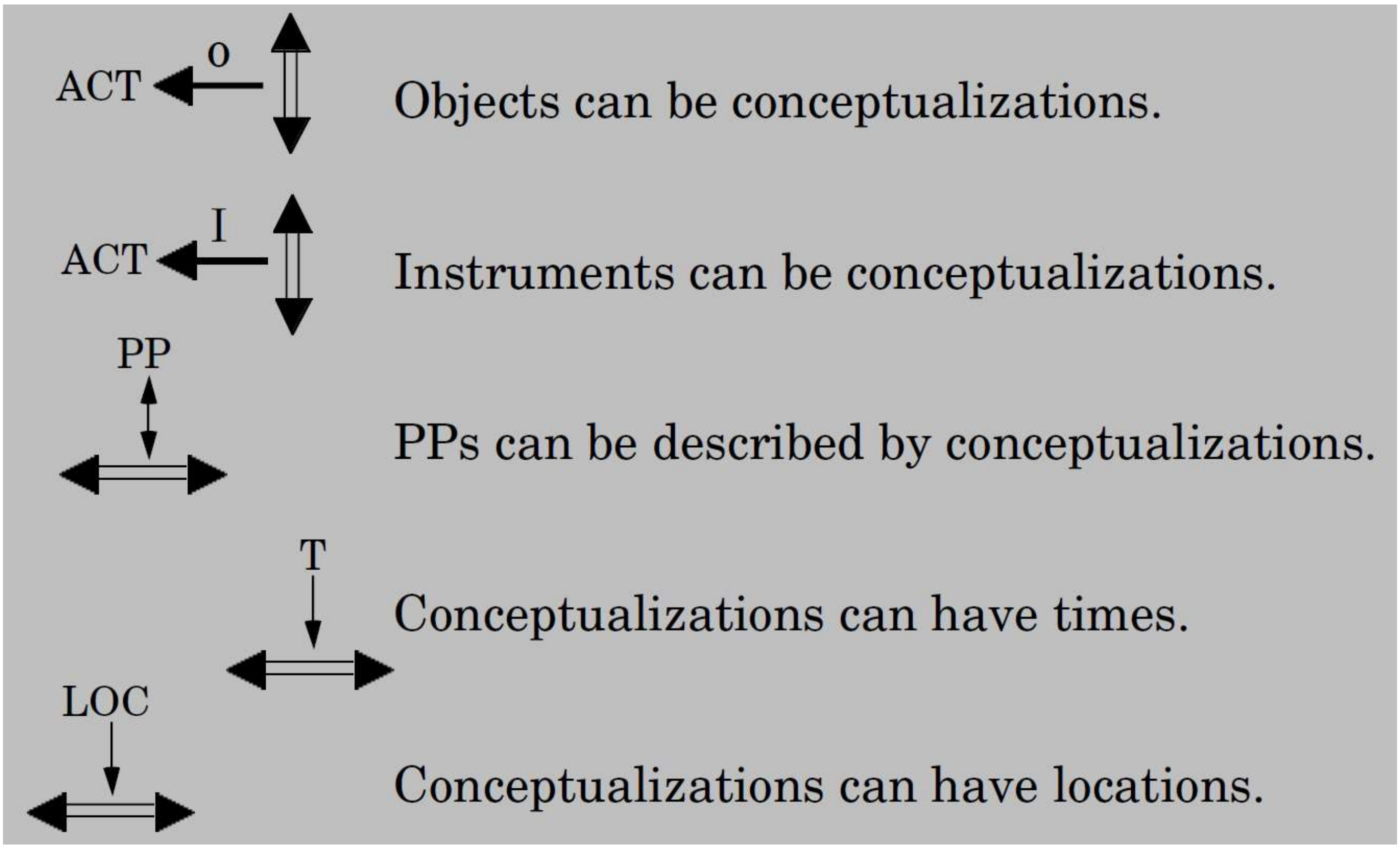
- PP (picture producer) : physical object.
  Actors must be an animate PP, or a natural force.
- ACT : One of eleven primitive actions.
- LOC : Location.
- T : Time.
- AA (action aider) : modifications of features of an ACT.
  e.g., speed factor in PROPEL.
- PA : attributes of an object, of the form STATE(VALUE).
  e.g., COLOR(red).

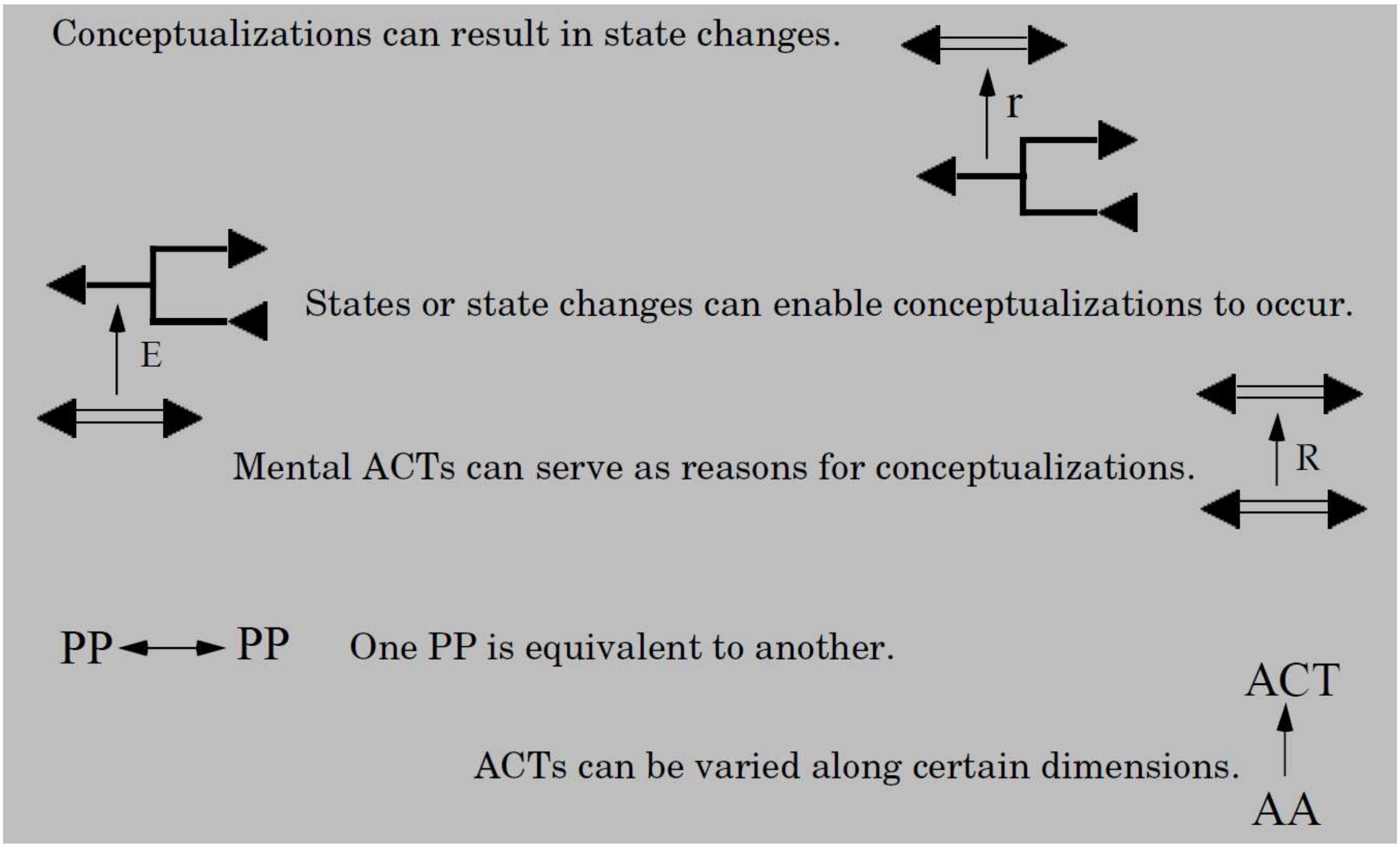
Conceptual Roles

- Conceptualization: The basic unit of the conceptual level of understanding.
- Actor: The performer of an ACT.
- ACT: An action done to an object.
- Object: A thing that is acted upon.
- Recipient: The reciever of an object as the result of an ACT.
- Direction: The location that an ACT is directed toward.
- State: The state that an object is in.

Conceptual Syntax Rules







Conceptual Tenses

past	p
future	f
negation	/
start of a transition	ts
end of a transition	tf
conditional	c
continuous	k
interrogative	?
timeless	∞
present	nil

States

states of objects are described by scales with numerical values.

HEALTH	(range -10 to 10)
dead	-10
gravely ill	-9
sick	-9 to -1
under the weather	-2
all right	0
tip top	+7
perfect health	+10