7 Knowledge Representation

Note: we will skip 7.3 and 7.4

Additional references for the slides:
Robert Wilensky's CS188 slides:
www.cs.berkeley.edu/~wilensky/cs188/lectures/index.html
John F. Sowa's examples: www.jfsowa.com/cg/cgexampw.htm

Chapter Objectives

- **Learn different formalisms for Knowledge Representation (KR)**
- **Learn about representing concepts in a canonical form**
- **Compare KR formalisms to predicate calculus**
- **The agent model: Transforms percepts and results of its own actions to an internal representation**

“Shortcomings” of logic

- **Emphasis on *truth-preserving operations* rather than the nature of human reasoning (or natural language understanding)**

- **if-then relationships do not always reflect how humans would see it:**

$$\forall X\ (\text{cardinal}\ (X) \rightarrow \text{red}(X))$$

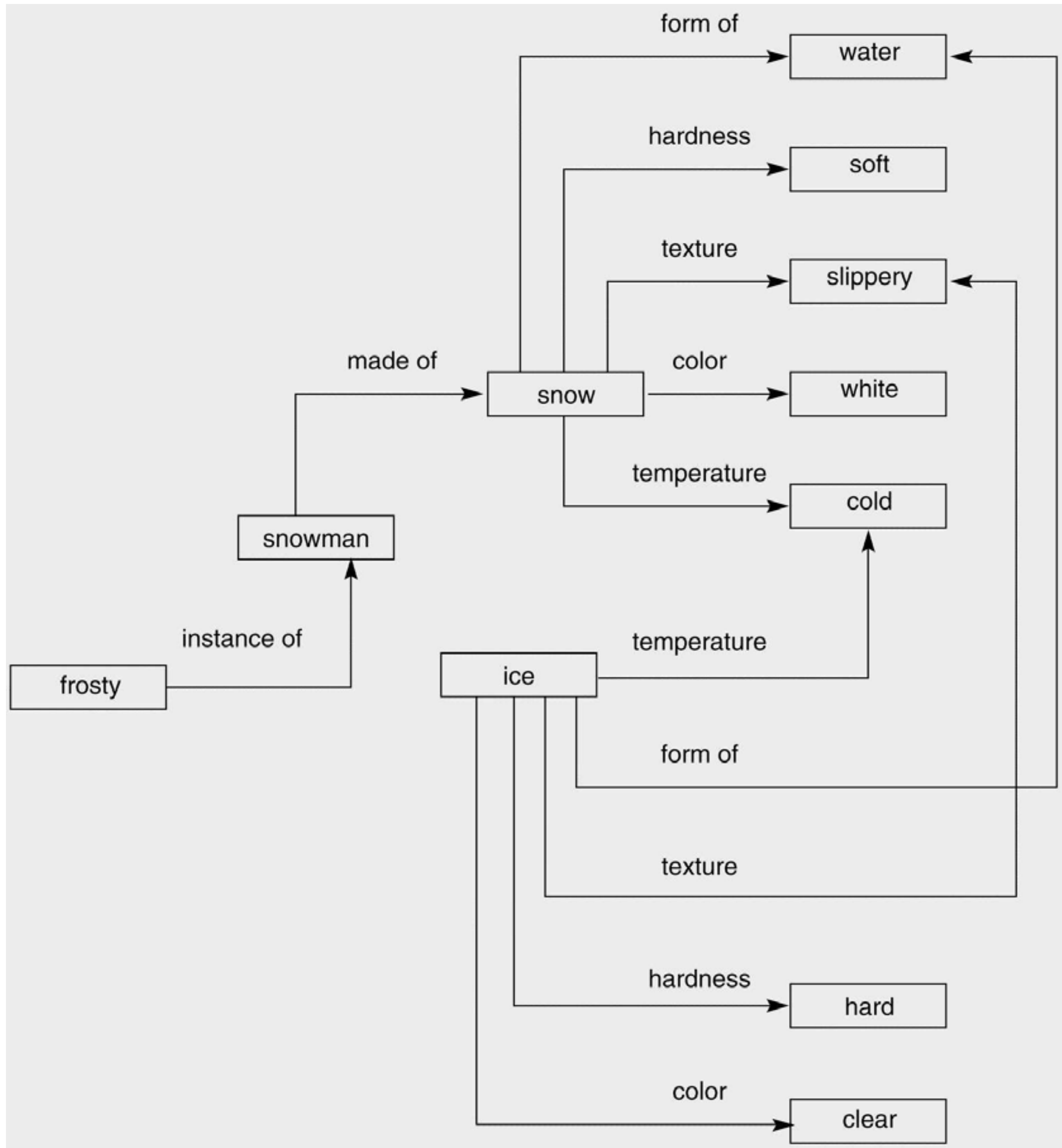
$$\forall X(\neg\ \text{red}\ (X) \rightarrow \neg\ \text{cardinal}(X))$$

- ***Associations* between concepts is not always clear**

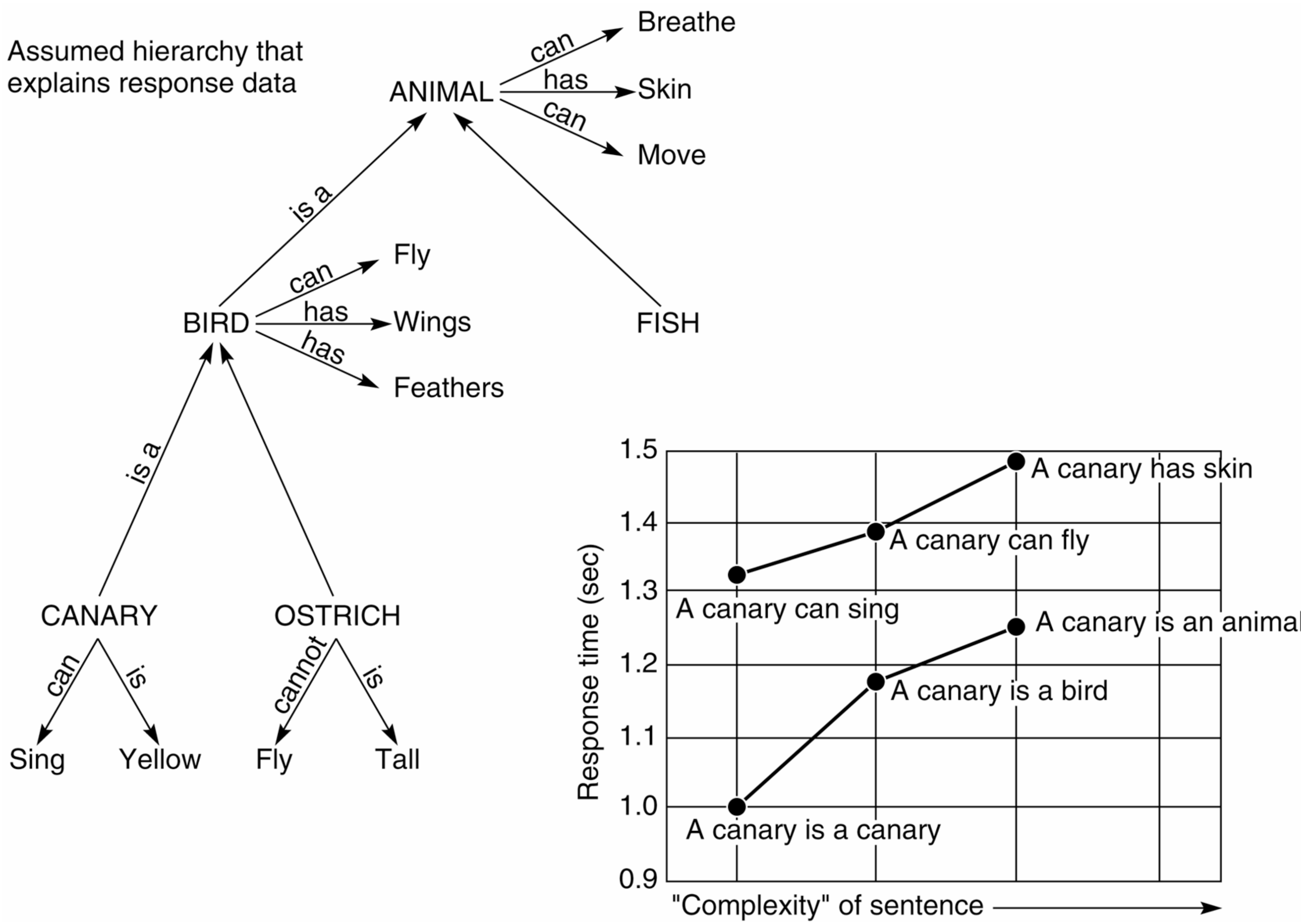
  snow: cold, white, snowman, slippery, ice, drift, blizzard

- **Note however, that the issue here is clarity or ease of understanding rather than expressiveness.**

Network representation of properties of snow and ice



Semantic network developed by Collins and Quillian (Harmon and King 1985)



Meanings of words (concepts)

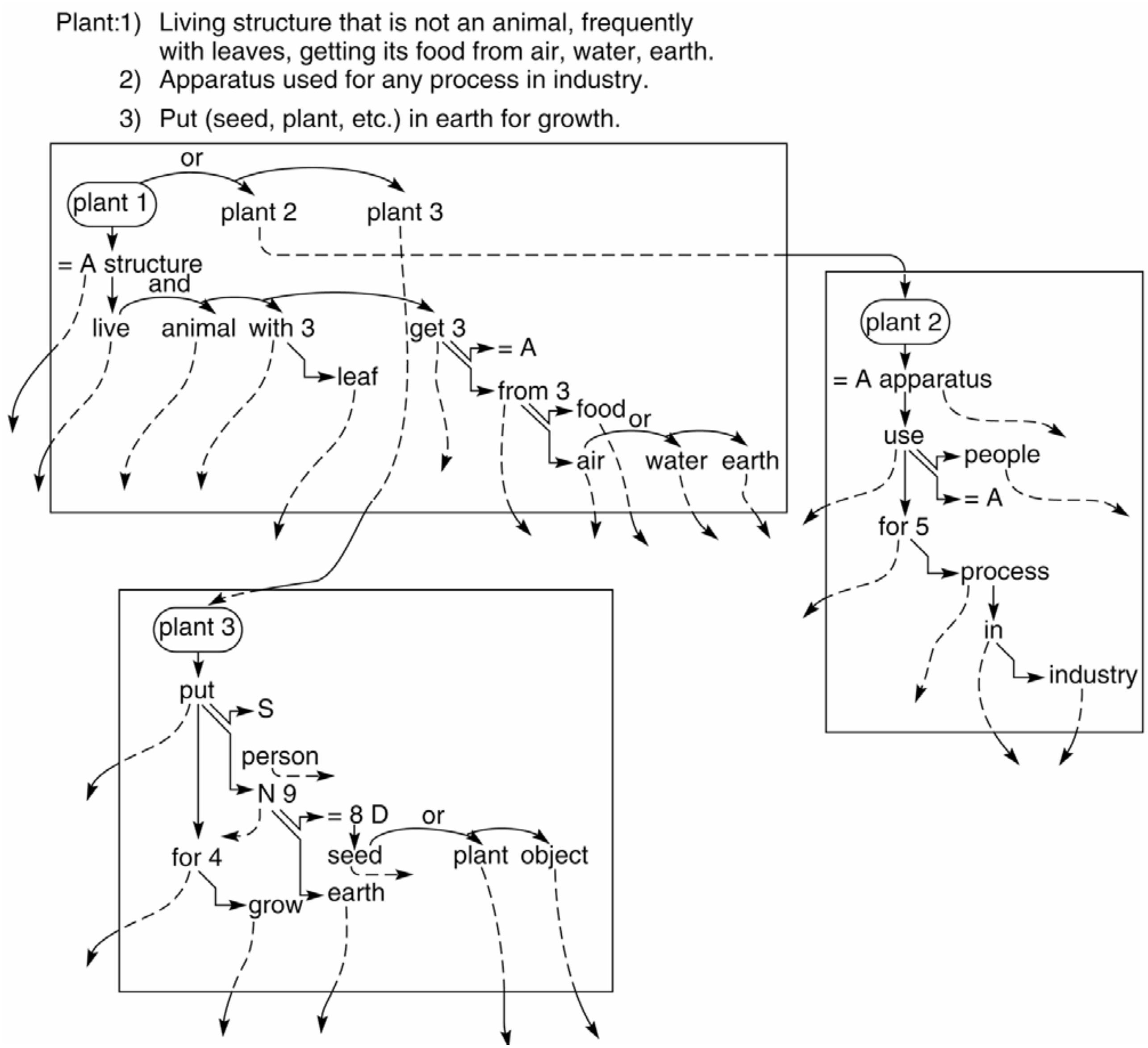
The plant did not seem to be in good shape.

Bill had been away for several days and nobody watered it.

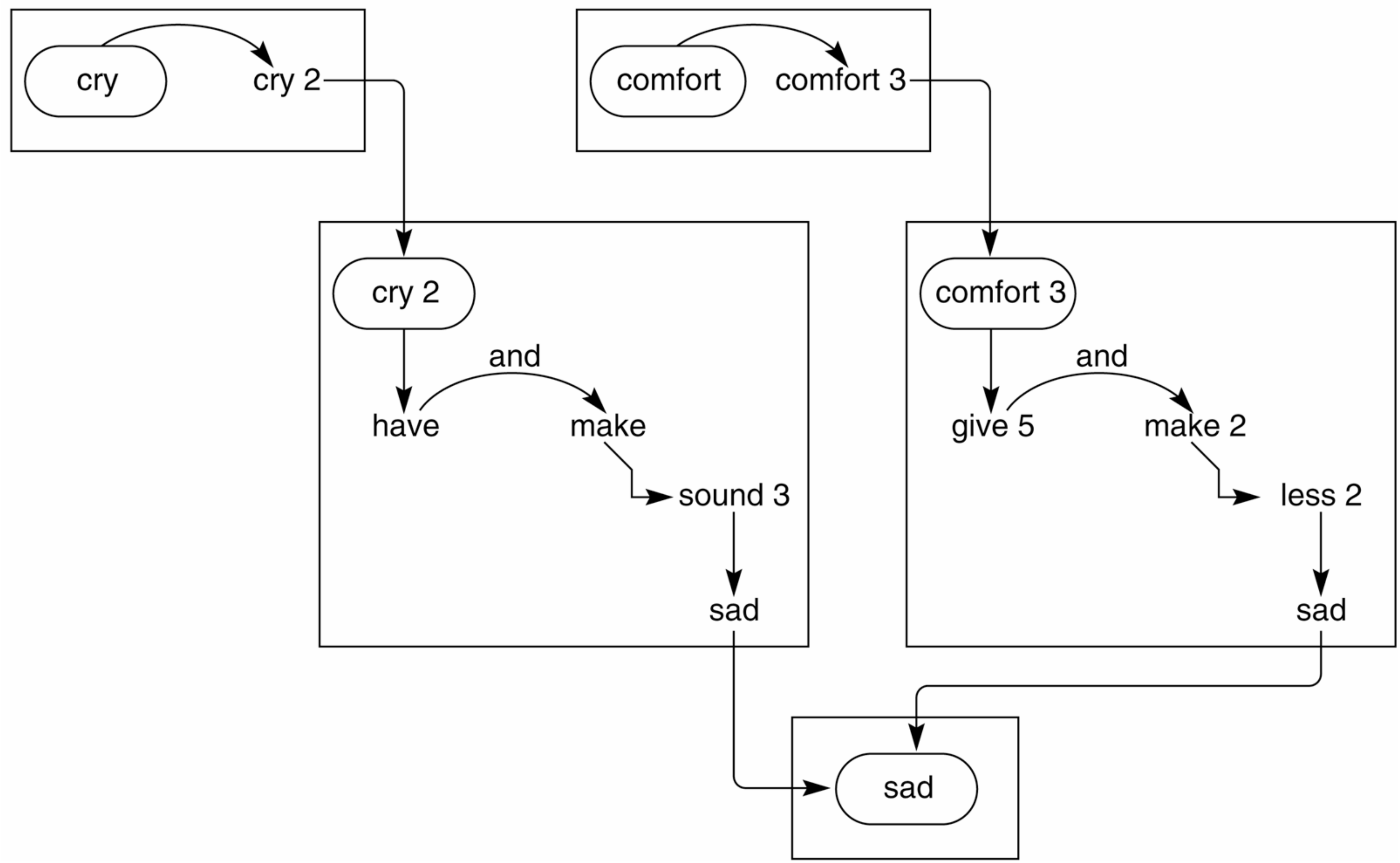
OR

The workers had been on strike for several days and regular maintenance was not carried out.

Three planes representing three definitions of the word “plant” (Quillian 1967)



Intersection path between "cry" and "comfort" (Quillian 1967)



“Case” oriented representation schemes

- Focus on the *case structure* of English verbs
- Case relationships include:

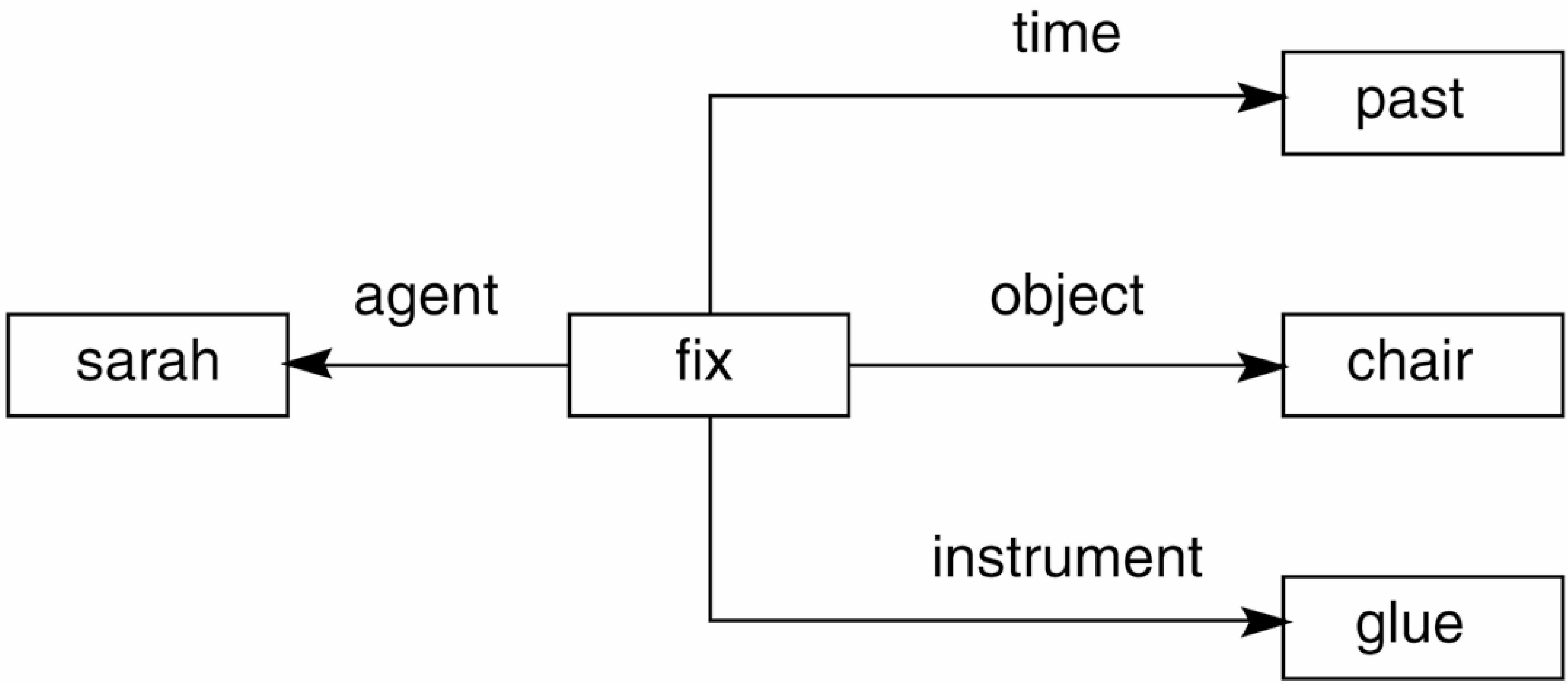
agent	location
object	time
instrument	

- Two approaches

  case frames: A sentence is represented as a verb node, with various case links to nodes representing other participants in the action

  conceptual dependency theory: The situation is classified as one of the standard action types. Actions have conceptual cases (e.g., actor, object).

Case frame representation of “Sarah fixed the chair with glue.”



Conceptual Dependency Theory

- **Developed by Schank, starting in 1968**
- **Tried to get as far away from language as possible, embracing canonical form, proposing an interlingua**
- **Borrowed**
  - **from Colby and Abelson, the terminology that sentences reflected *conceptualizations*, which combine *concepts***
  - **from case theory, the idea of *cases*, but rather assigned these to underlying concepts rather than to linguistic units (e.g., verbs)**
  - **from the dependency grammar of David Hayes, idea of *dependency***

Basic idea

- **Consider the following story:**

Mary went to the playroom when she heard Lily crying.
Lily said, “Mom, John hit me.”
Mary turned to John, “You should be gentle to your little sister.”
“I’m sorry mom, it was an accident, I should not have kicked the ball towards her.” John replied.

- **What are the facts we know after reading this?**

Basic idea (cont'd)

Mary went to the playroom when she heard Lily crying.
Lily said, "Mom, John hit me."
Mary turned to John, "You should be gentle to your little sister."
"I'm sorry mom, it was an accident, I should not have kicked the ball towards her."
John replied.

Mary's location changed.

Lily was sad, she was crying.

John hit Lily (with an unknown object).

John is Lily's brother.

John is taller (bigger) than Lily.

John kicked a ball, the ball hit Lily.

“John hit the cat.”

- **First, classify the situation as of type *Action*.**
- **Actions have cocceptual cases, e.g., all actions require**
  - **Act (the particular type of action)**
  - **Actor (the responsible party)**
  - **Object (the thing acted upon)**

ACT: [apply a force] or PROPEL
ACTOR: john
OBJECT: cat

$$\text{john} \Leftrightarrow \text{PROPEL} \overset{o}{\leftarrow} \text{cat}$$

Conceptual dependency theory

Four primitive conceptualizations:

- **ACTs** **actions**
- **PPs** **objects (picture producers)**
- **AAs** **modifiers of actions (action aiders)**
- **PAs** **modifiers of objects (picture aiders)**

Conceptual dependency theory (cont'd)

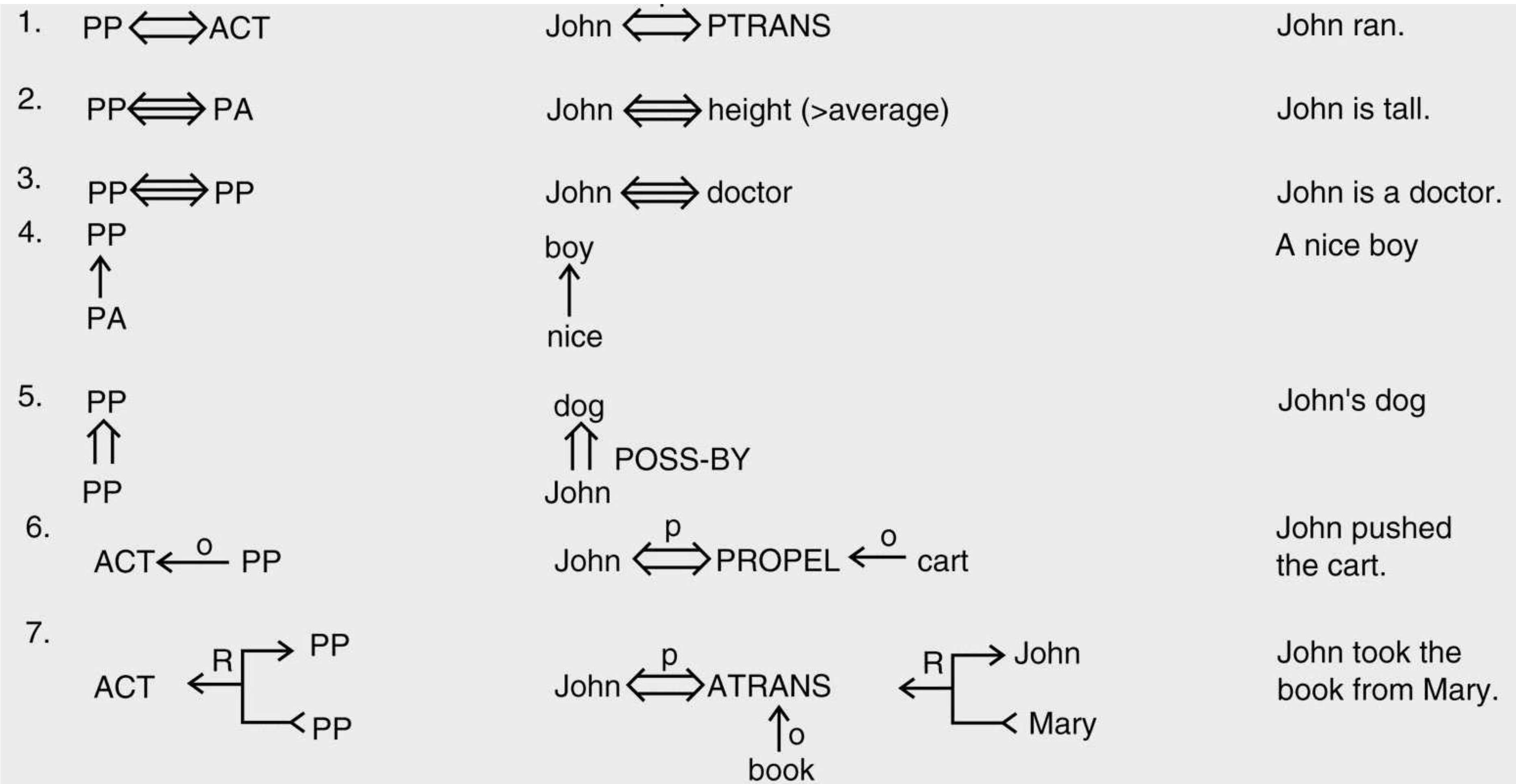
Primitive acts:

• ATRANS	transfer a relationship (give)
• PTRANS	transfer of 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)
• MTRANS	transfer mental information (tell)
• MBUILD	mentally make new information (decide)
• CONC	conceptualize or think about an idea (think)
• SPEAK	produce sound (say)
• ATTEND	focus sense organ (listen)

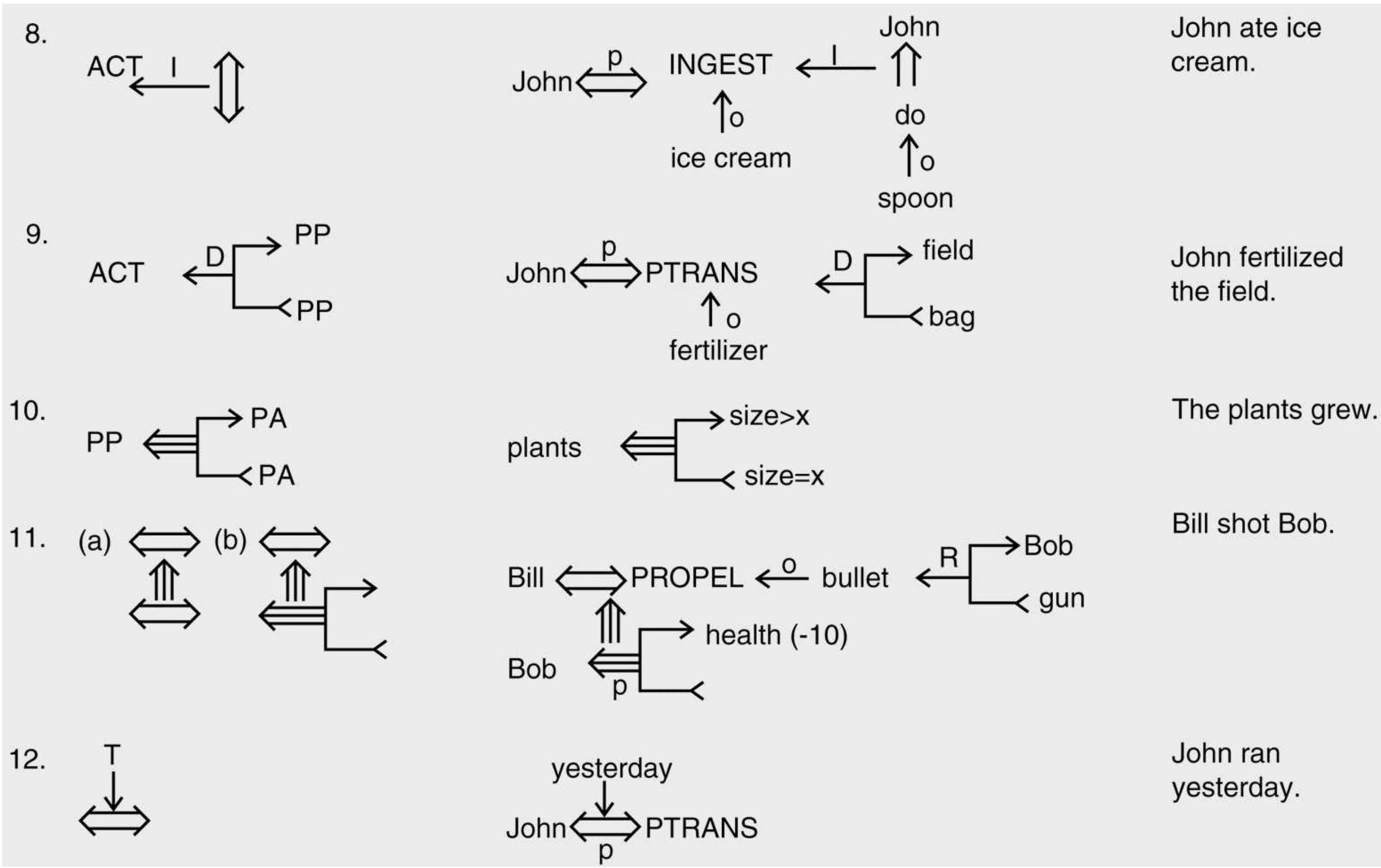
Basic conceptual dependencies

$PP \Leftrightarrow ACT$	indicates that an actor acts.
$PP \Leftrightarrow PA$	indicates that an object has a certain attribute.
$ACT \xleftarrow{O} PP$	indicates the object of an action.
$ACT \xleftarrow{R} \begin{array}{l} \rightarrow PP \\ \leftarrow PP \end{array}$	indicates the recipient and the donor of an object within an action.
$ACT \xleftarrow{D} \begin{array}{l} \rightarrow PP \\ \leftarrow PP \end{array}$	indicates the direction of an object within an action.
$ACT \xleftarrow{1} \Updownarrow$	indicates the instrumental conceptualization for an action.
$\begin{array}{c} X \\ \Uparrow \\ Y \end{array}$	indicates that conceptualization X caused conceptualization Y. When written with a C this form denotes that X COULD cause Y.
$PP \Leftarrow \begin{array}{l} \rightarrow PA2 \\ \leftarrow PA1 \end{array}$	indicates a state change of an object.
$PP1 \leftarrow PP2$	indicates that PP2 is either PART OF or the POSSESSOR OF PP1.

Examples with the basic conceptual dependencies



Examples with the basic conceptual dependencies (cont'd)



CD is a decompositional approach

“John took the book from Pat.”

John <≡> *ATRANS* $\xleftarrow{o}$ book $\leftarrow$ (→ John, ← Pat)

The above form also represents:
“Pat received the book from John.”

The representation analyzes surface forms into an underlying structure, in an attempt to capture common meaning elements.

CD is a decompositional approach

“John gave the book to Pat.”

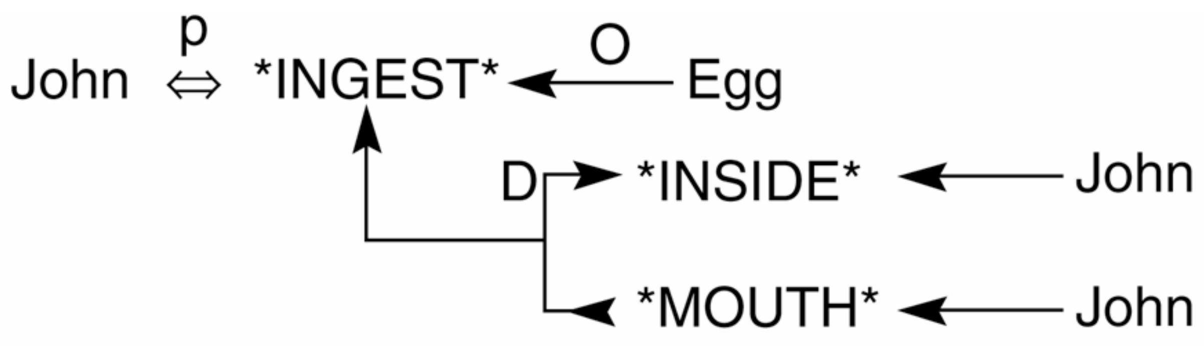
John <≡> *ATRANS* $\overset{o}{\leftarrow}$ book $\leftarrow$ (→ Pat, ← John)

Note that only the donor and recipient have changed.

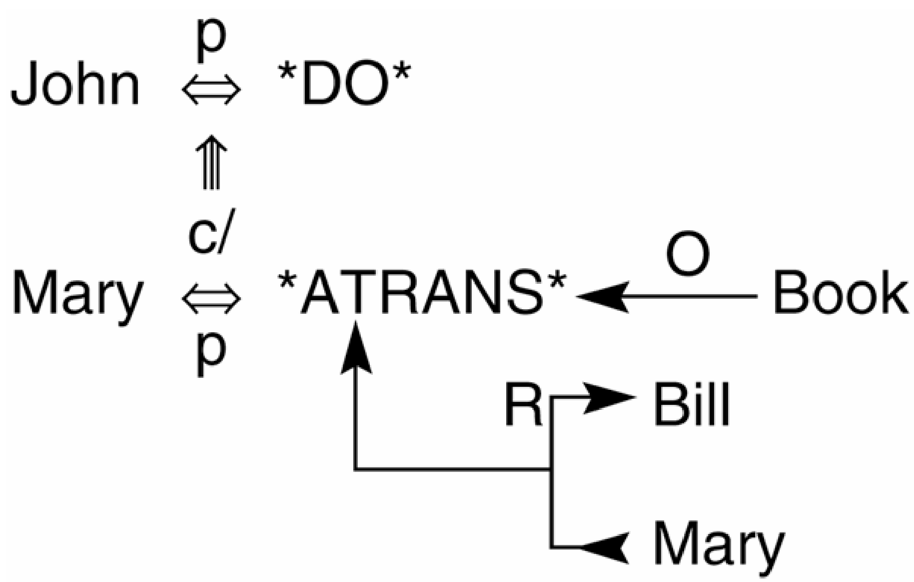
Ontology

- Situations were divided into several types:
  - Actions
  - States
  - State changes
  - Causals
- There wasn't much of an attempt to classify objects

“John ate the egg.”



“John prevented Mary from giving a book to Bill”



Representing Picture Aiders (PAs) or states

thing <≡> state-type (state-value)

- **"The ball is red"** — **ball <≡> color (red)**
- **"John is 6 feet tall"** — **john <≡> height (6 feet)**
- **"John is tall"** — **john <≡> height (>average)**
- **"John is taller than Jane"**

  john <≡> height (X)
  jane <≡> height (Y)
  X > Y

More PA examples

• “John is angry.”	john <≡> anger(5)
• “John is furious.”	john <≡> anger(7)
• “John is irritated.”	john <≡> anger (2)
• “John is ill.”	john <≡> health (-3)
• “John is dead.”	john <≡> health (-10)

Many states are viewed as points on scales.

Scales

- There should be lots of scales
  - The numbers themselves were not meant to be taken seriously
  - But that lots of different terms differ only in how they refer to scales was
- An interesting question is which semantic objects are there to describe locations on a scale?
For instance, modifiers such as “very”, “extremely” might have an interpretation as “toward the end of a scale.”

Scales (cont'd)

- **What is "John grew an inch."**
- **This is supposed to be a state change: somewhat like an action but with no responsible agent posited**

Height (X+1)

John < Ξ

Height (X)

Variations on the story of the poor cat

"John applied a force to the cat by moving some object to come in contact with the cat"

John <≡> *PROPEL* $\xleftarrow{o}$ cat

$\uparrow$ i

John <≡> *PTRANS* $\xleftarrow{o}$ [ ] ←

→ loc(cat)

←

The arrow labeled 'i' denotes instrumental case

Variations on the cat story (cont'd)

"John kicked the cat."

John <≡> *PROPEL* $\xleftarrow{o}$ cat

↑ i

loc(cat)

John <≡> *PTRANS* $\xleftarrow{o}$ foot ←

kick = hit with one's foot

Variations on the cat story (cont'd)

"John hit the cat."

John <≡> *PROPEL* $\overset{o}{\leftarrow}$ **cat**

^

|||

cat <≡ → **Health(-2)**

Hitting was detrimental to the cat's health.

Causals

"John hurt Jane."

$$\text{John} \Leftrightarrow \text{DO} \overset{o}{\leftarrow} \text{Jane}$$

$$\Uparrow$$

$$\text{Jane} \Leftarrow \begin{cases} \rightarrow \text{Pain}(>X) \\ \leftarrow \text{Pain}(X) \end{cases}$$

John did something to cause Jane to become hurt.

Causals (cont'd)

"John hurt Jane by hitting her."

John <≡> PROPEL $\xleftarrow{o}$ **Jane**

$\Uparrow$

Jane <≡

Pain(> X)

Pain (X)

John hit Jane to cause Jane to become hurt.

How about?

"John killed Jane."

"John frightened Jane."

"John likes ice cream."

“John killed Jane.”

John <≡> *DO*

^
|||

Jane <≡

Health(-10)

Health(> -10)

“John frightened Jane.”

John <≡> *DO*

^
|||

Jane <≡

Fear (> X)

Fear (X)

“John likes ice cream.”

John <≡> *INGEST* $\xleftarrow{o}$ IceCream

^
|||

John <≡ → Joy (> X)
← Joy (X)

Comments on CD theory

- **Ambitious attempt to represent information in a language independent way**
  - **formal theory of natural language semantics, reduces problems of ambiguity**
  - **canonical form, internally syntactically identical**
  - **decomposition addresses problems in case theory by revealing underlying conceptual structure. Relations are between concepts, not between linguistic elements**
  - **prospects for machine translation are improved**

Comments on CD theory (cont'd)

The major problem is incompleteness

- no quantification
- no hierarchy for objects (and actions), everything is a primitive
- are those the right primitives?
- Is there such a thing as a conceptual primitive?
  (e.g., MOVE to a physiologist is complex)
- how much should the inferences be carried? CD didn't explicitly include logical entailments such as "hit" entails "being touched", "bought" entails being at a store
- fuzzy logic? Lots of linguistic details are very lexically-dependent, e.g., likely, probably
- still not well studied/understood, a more convincing methodology never arrived

Understanding stories about restaurants

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

Did John eat dinner?
Did John pay by cash or credit card?
What did John buy?
Did he stop at the bank on the way home?

Restaurant stories (cont'd)

Sue went out to lunch. She sat at a table and called a waitress, who brought her a menu. She ordered a sandwich.

Was Sue at a restaurant?
Why did the waitress bring Sue a menu?
Who does "she" refer to in the last sentence?

Restaurant stories (cont'd)

Kate went to a restaurant. She was shown to a table and ordered steak from a waitress. She sat there and waited for a long time. Finally, she got mad and she left.

Who does "she" refer to in the third sentence?
Why did Kate wait?
Why did she get mad? (might not be in the "script")

Restaurant stories (cont'd)

John visited his favorite restaurant on the way to the concert. He was pleased by the bill because he liked Mozart.

Which bill? (which “script” to choose: restaurant or concert?)

Scripts

- ***Entry conditions*: conditions that must be true for the script to be called.**
- ***Results*: conditions that become true once the script terminates.**
- ***Props*: "things" that support the content of the script.**
- ***Roles*: the actions that the participants perform.**
- ***Scenes*: a presentation of a temporal aspect of a script.**

A RESTAURANT script

Script:	**RESTAURANT**
Track:	**coffee shop**
Props:	**Tables, Menu, F = food,** **Check, Money**
Roles:	**S= Customer** **W = Waiter** **C = Cook** **M = Cashier** **O = Owner**

A RESTAURANT script (cont'd)

Entry conditions:	**S is hungry** **S has money**
Results:	**S has less money** **O has more money** **S is not hungry** **S is pleased (optional)**

A RESTAURANT script (cont'd)

Scene 1: Entering

S PTRANS S into restaurant

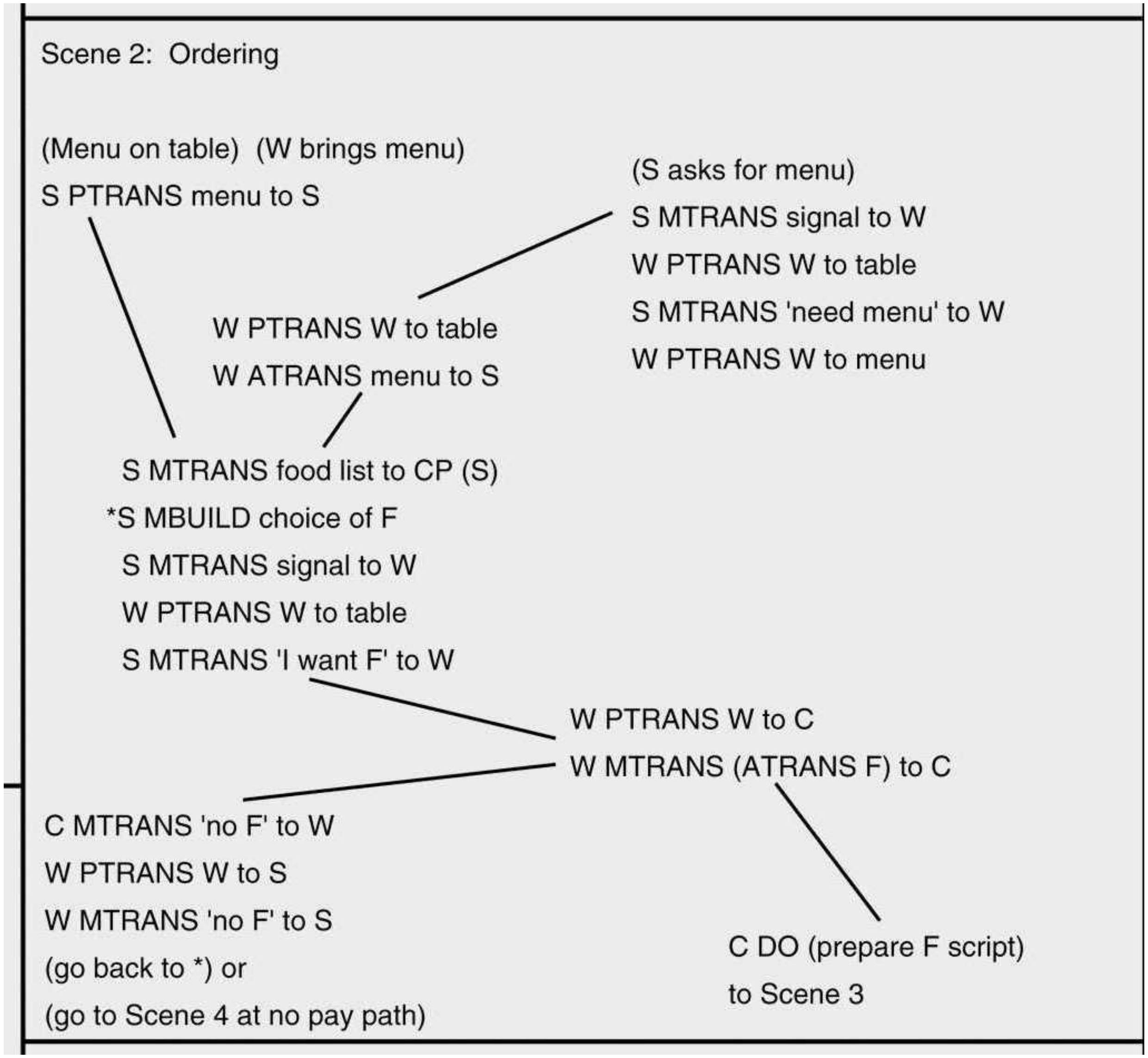
S ATTEND eyes to tables

S MBUILD where to sit

S PTRANS S to table

S MOVE S to sitting position

A RESTAURANT script (cont’d)



A RESTAURANT script (cont'd)

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 S to out of restaurant

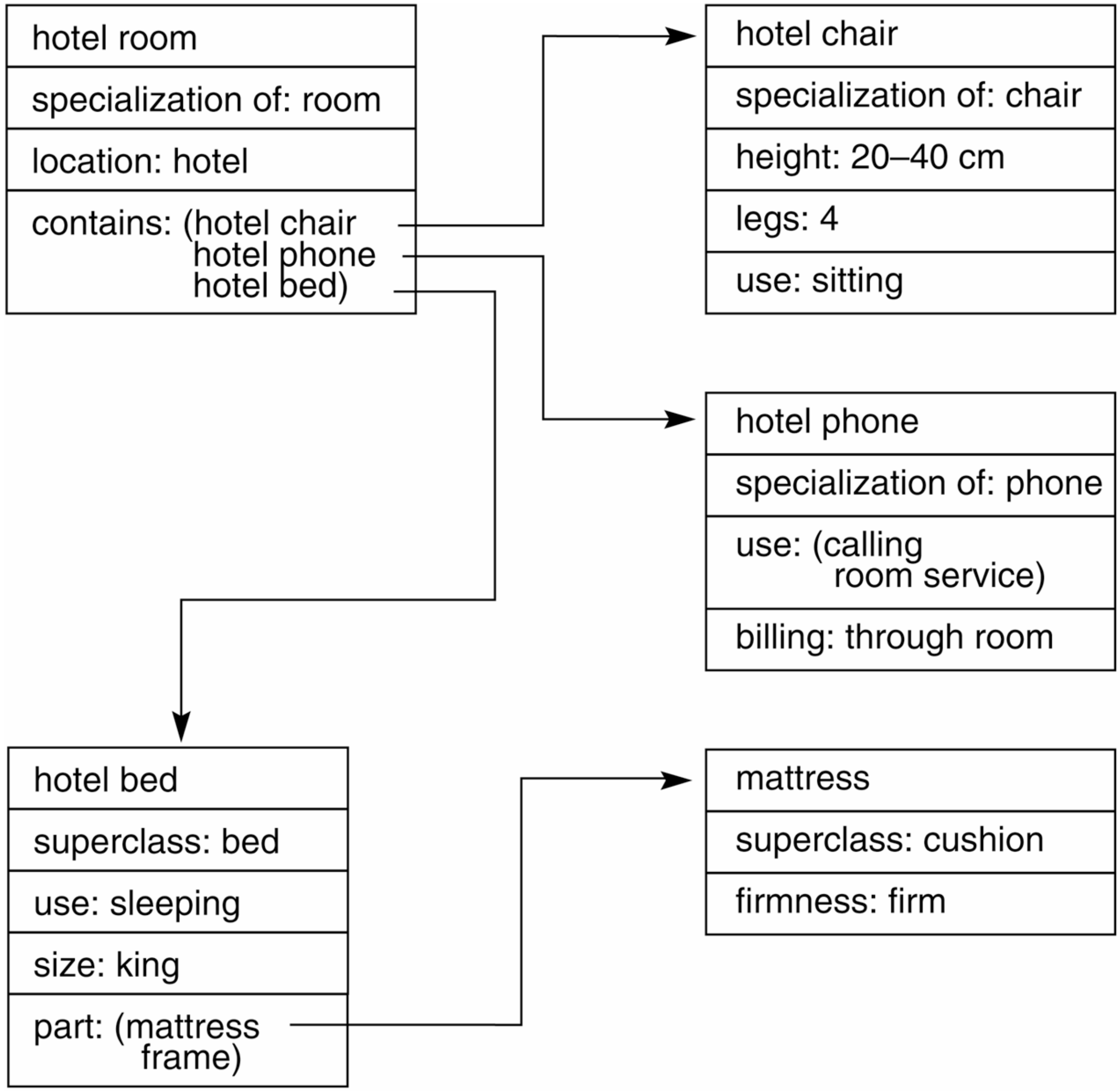
Frames

Frames are similar to scripts, they organize stereotypic situations.

Information in a frame:

- **Frame identification**
- **Relationship to other frames**
- **Descriptors of the requirements**
- **Procedural information**
- **Default information**
- **New instance information**

Part of a frame description of a hotel room



Conceptual graphs

A finite, connected, bipartite graph

Nodes: either concepts or conceptual relations

Arcs: no labels, they represent relations between concepts

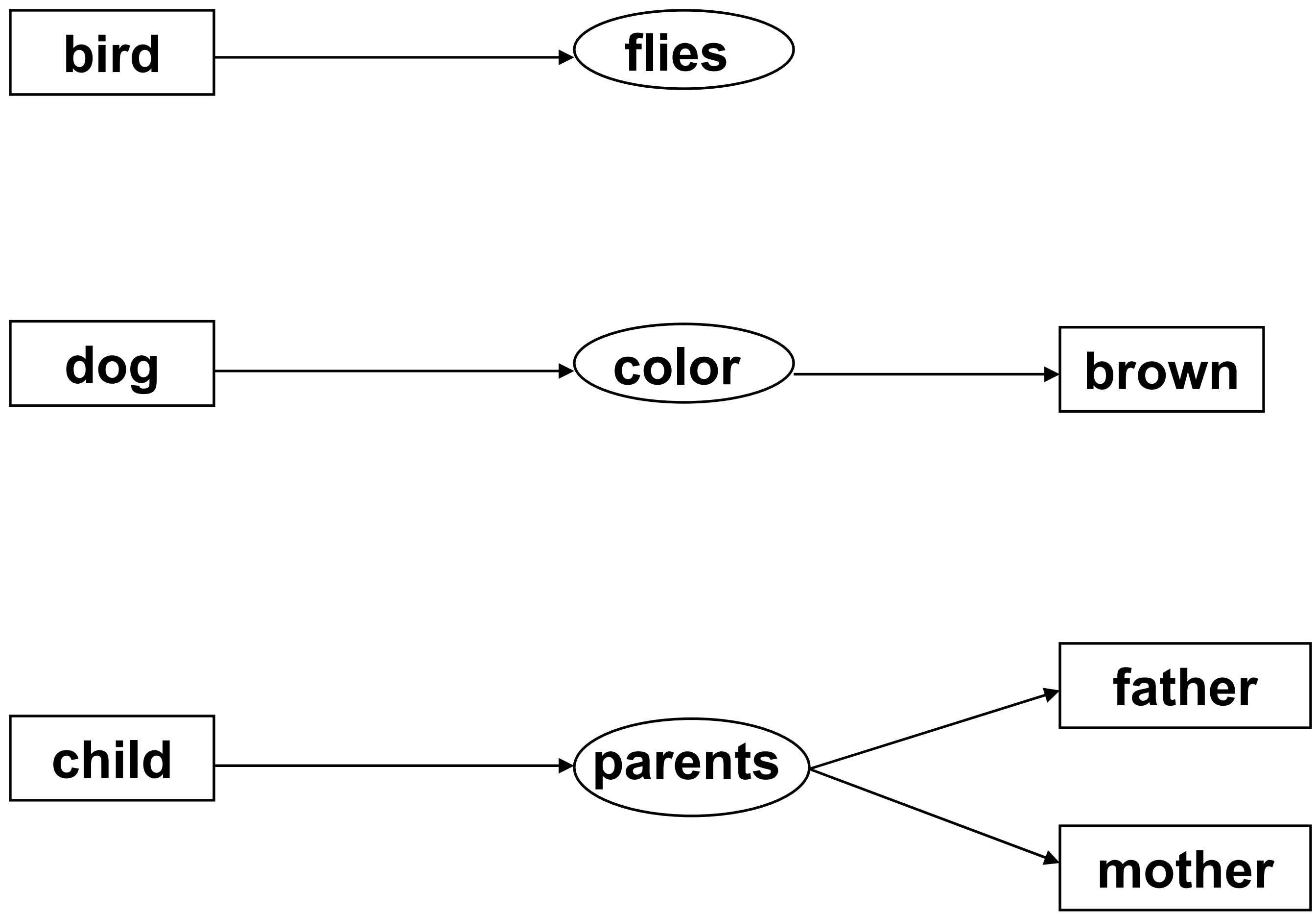
Concepts: concrete (e.g., book, dog) or abstract (e.g., like)

Conceptual relations of different arities

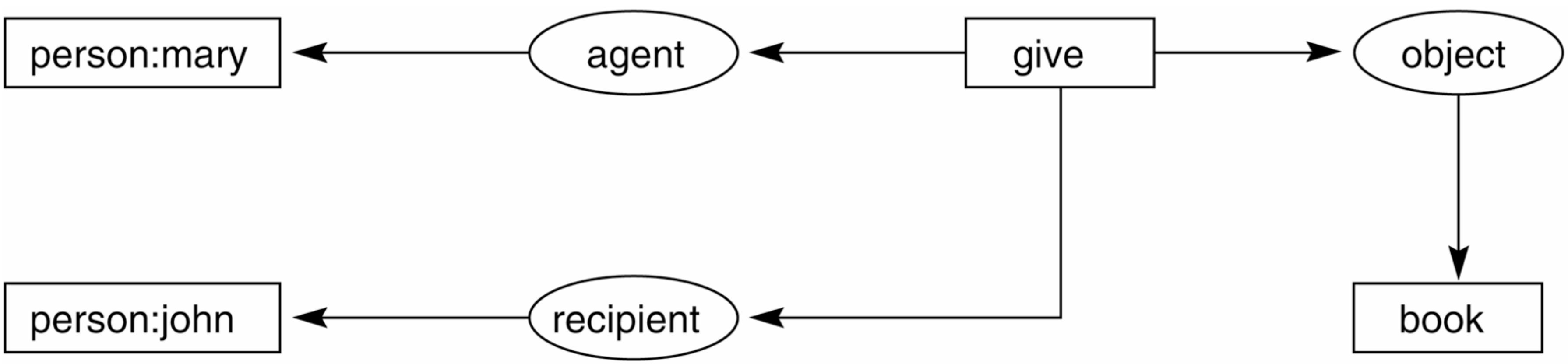
Flies is a unary relation

Color is a binary relation

Parents is a ternary relation

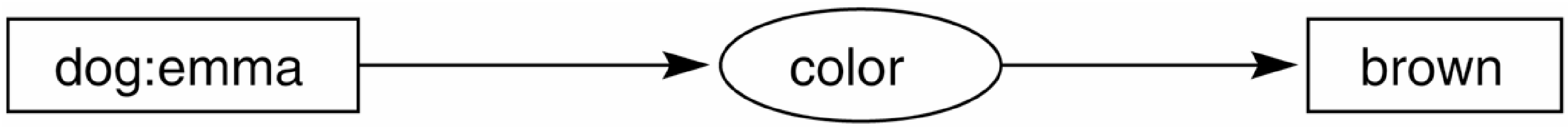


“Mary gave John the book.”



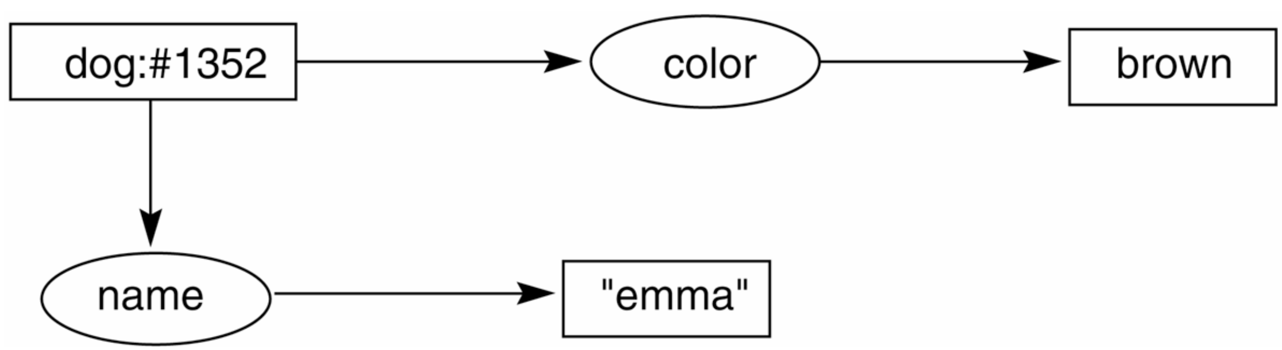
Conceptual graphs involving a brown dog

Conceptual graph indicating that the dog named emma dog is brown:

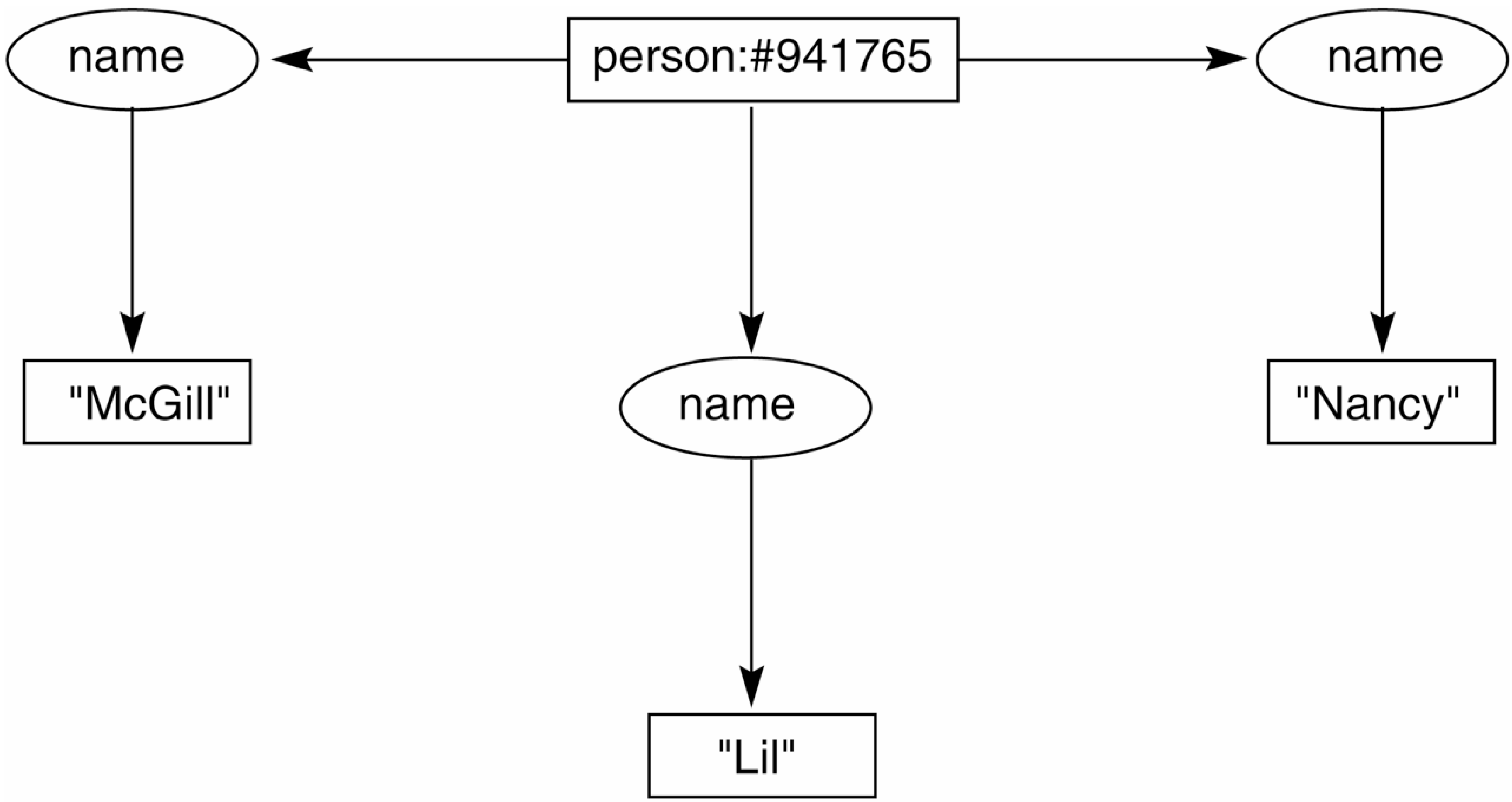


Conceptual graph indicating that a particular (but unnamed) dog is brown:

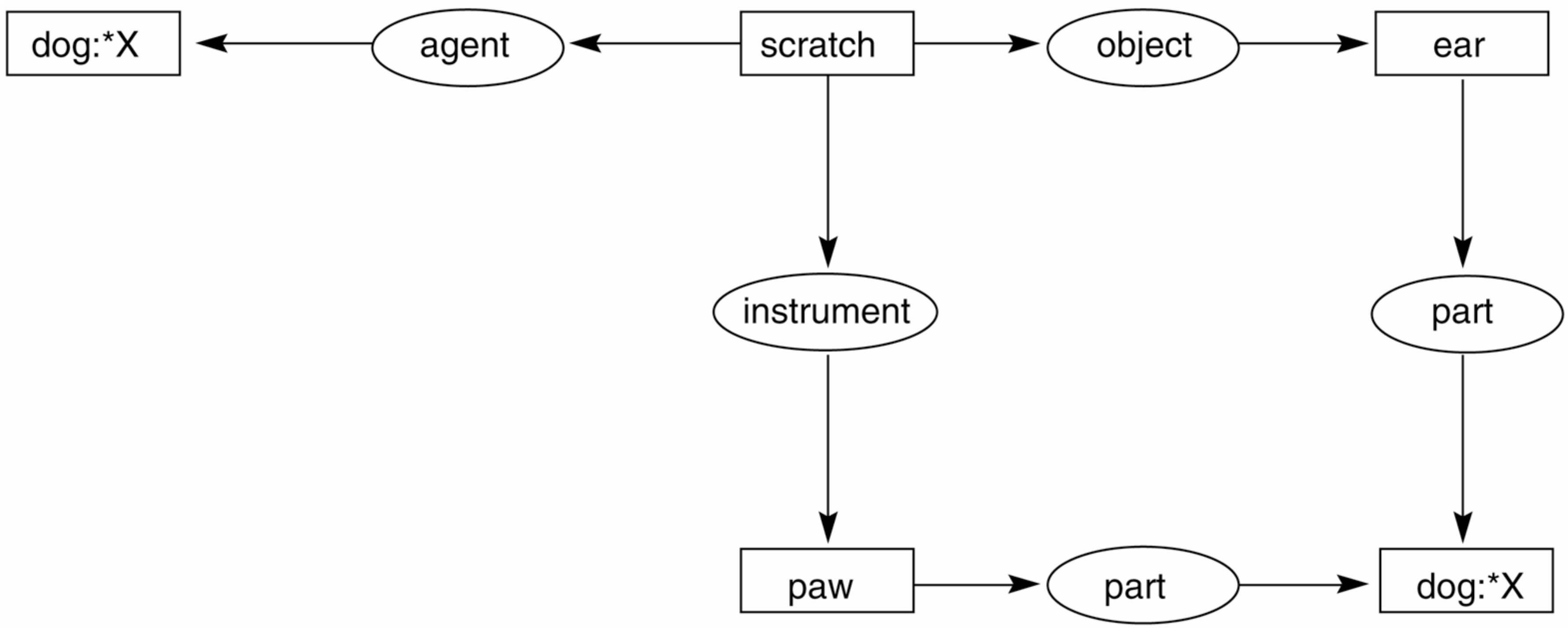
Conceptual graph indicating that a dog named emma is brown:



Conceptual graph of a person with three names



“The dog scratches its ear with its paw.”



The type hierarchy

A *partial ordering* on the set of types:

$$t \leq s$$

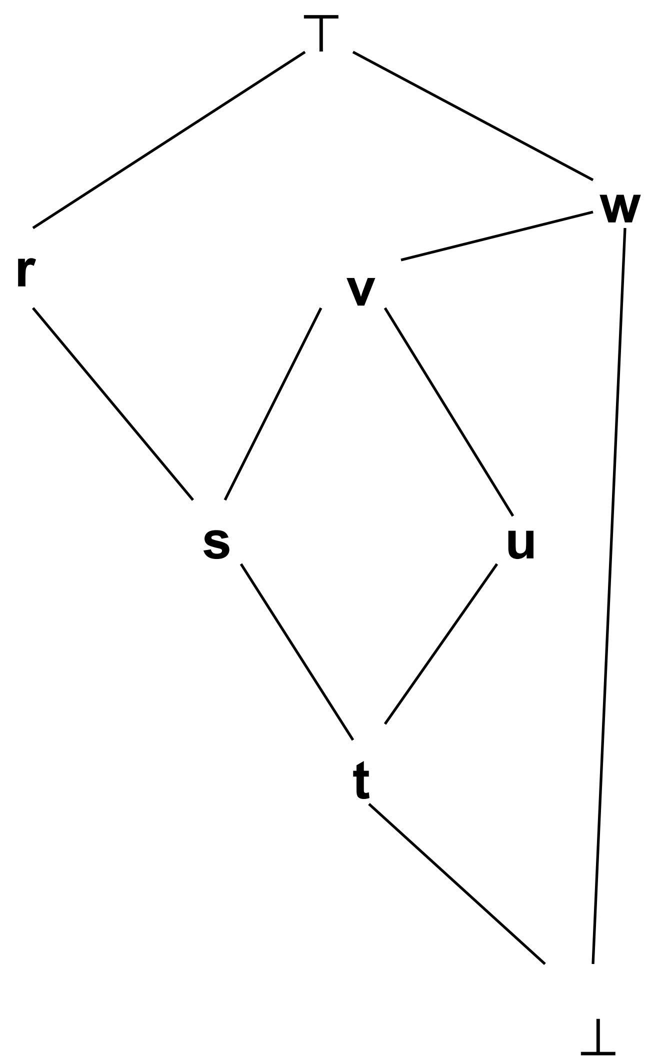
where, t is a *subtype* of s, s is a *supertype* of t.

If $t \leq s$ and $t \leq u$, then t is a *common subtype* of s and u.

If $s \leq v$ and $u \leq v$, then v is a *common supertype* of s and u.

Notions of: ***minimal common supertype***
maximal common subtype

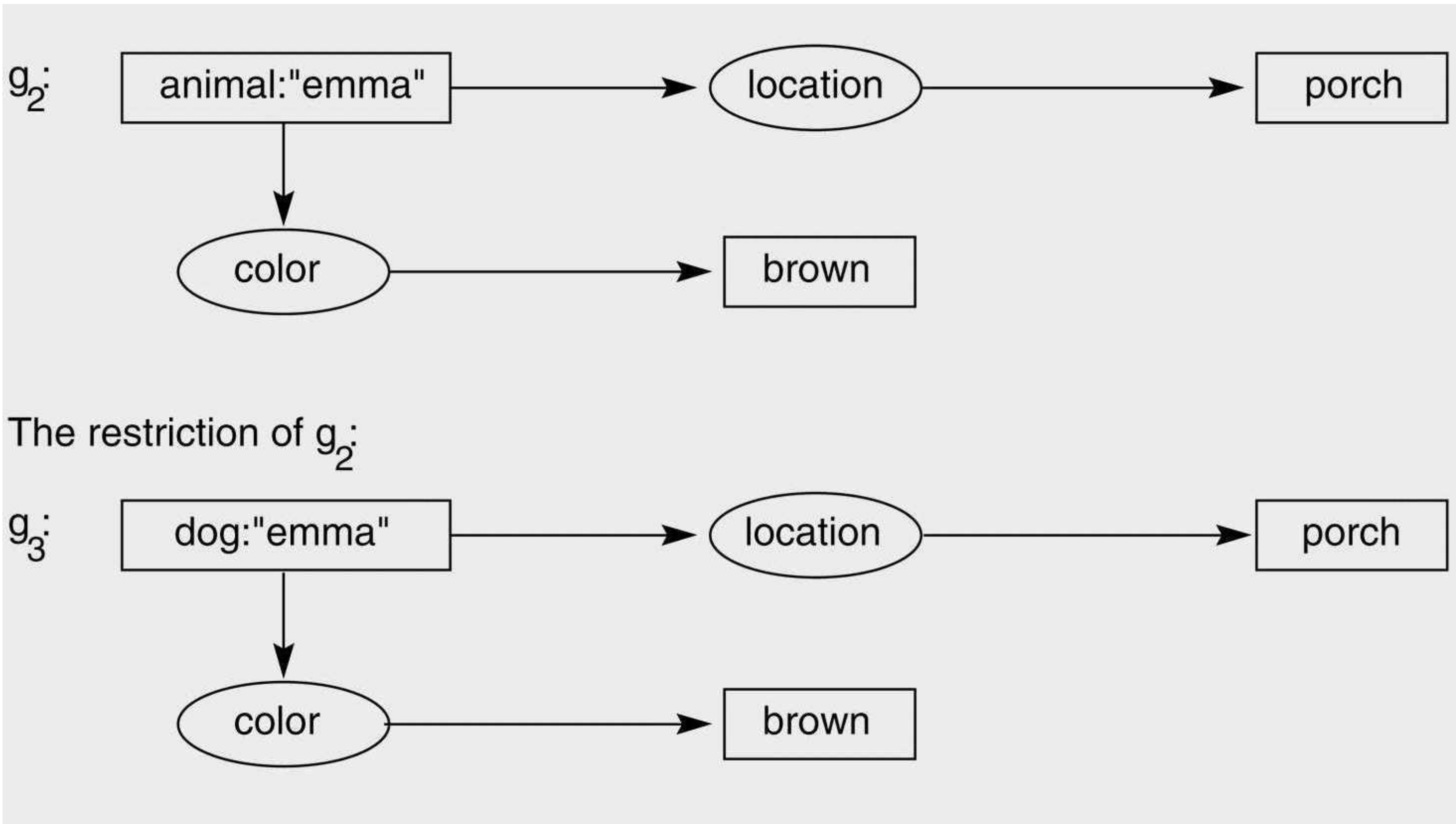
A lattice of subtypes, supertypes, the universal type, and the absurd type



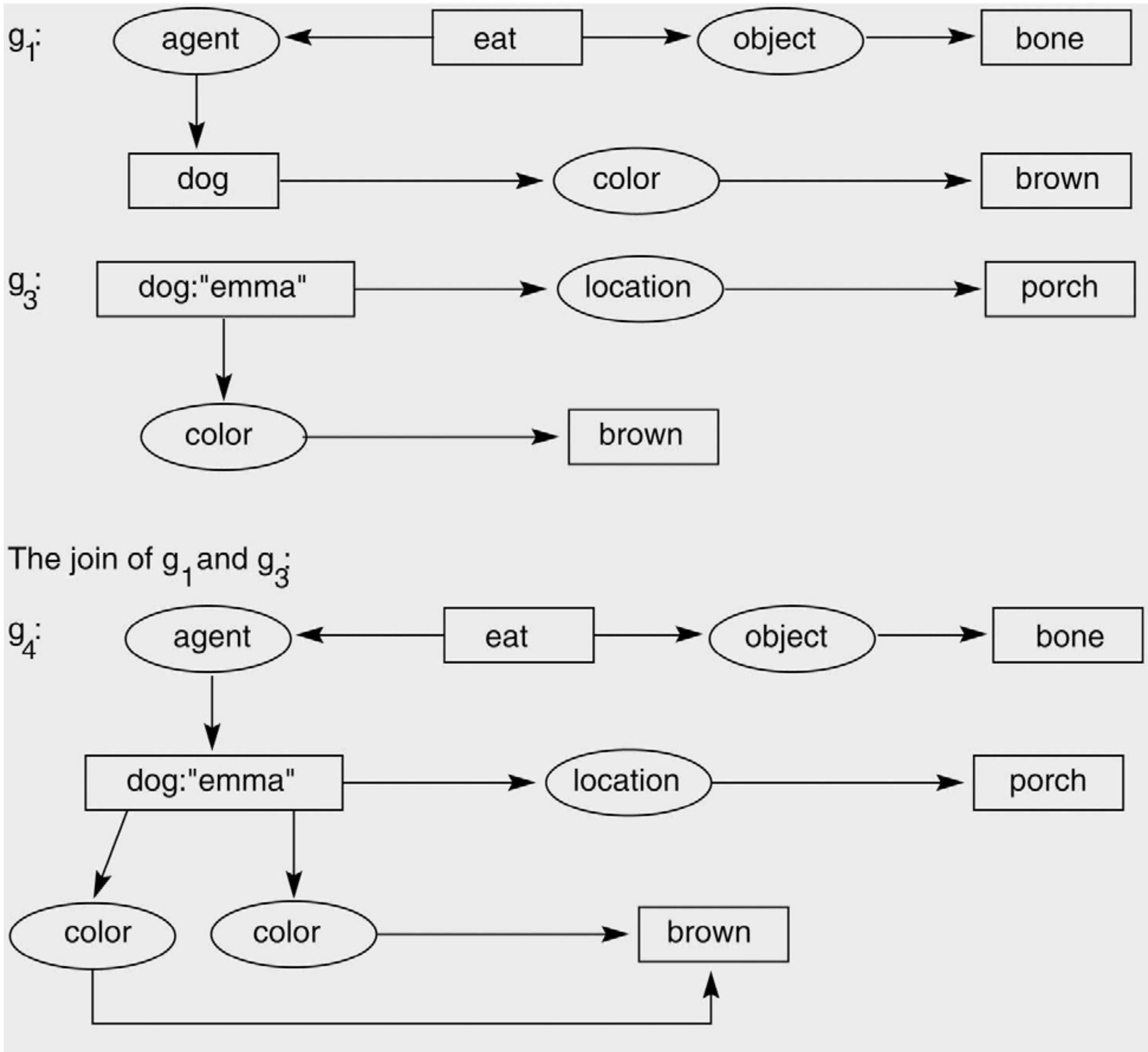
Four graph operations

- ***copy*: exact copy of a graph**
- ***restrict*: replace a concept node with a node representing its specialization**
- ***join*: combines graph based on identical nodes**
- ***simplify*: delete duplicate relations**

Restriction

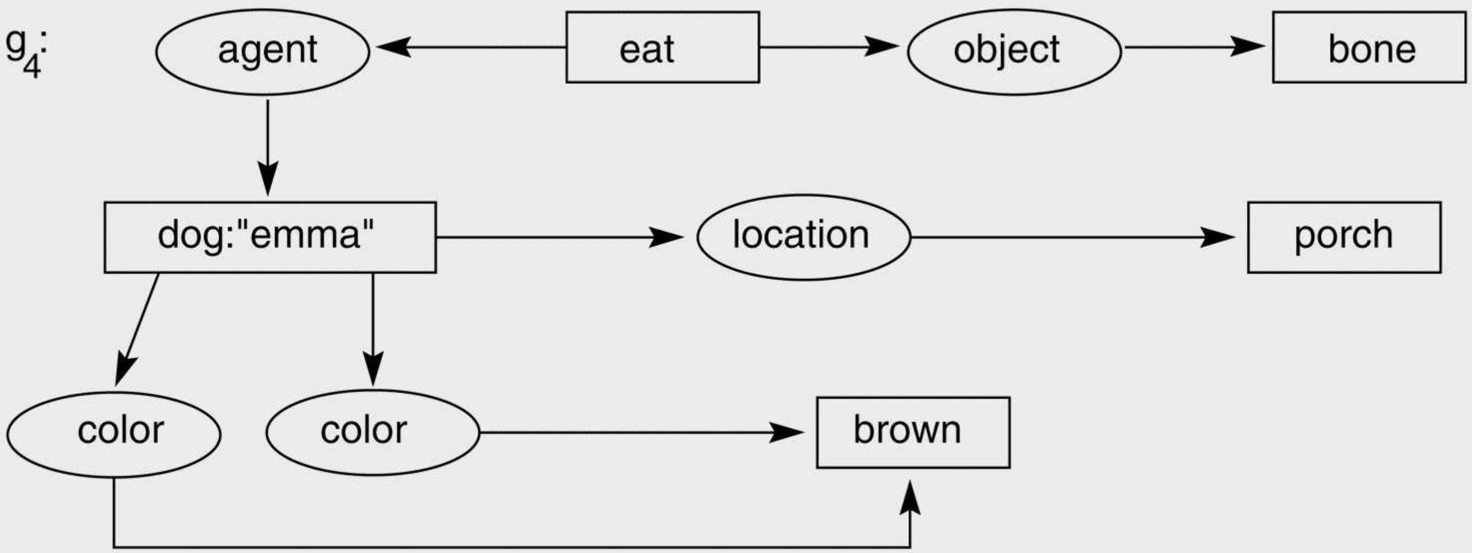


Join

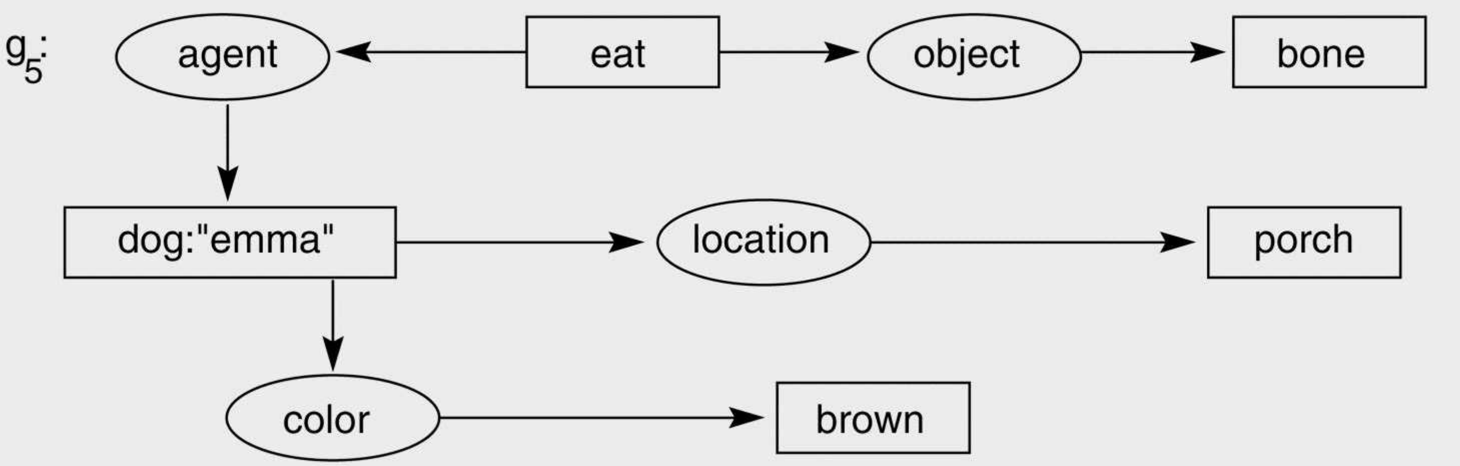


g1:
agent
eat
object
bone
dog
color
brown
g3:
dog:"emma"
location
porch
color
brown
The join of g1 and g3:
g4:
agent
eat
object
bone
dog:"emma"
location
porch
color
color
brown

Simplify



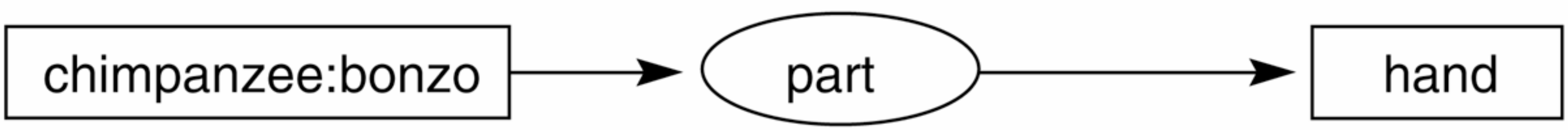
The simplify of g_4:



Inheritance in conceptual graphs

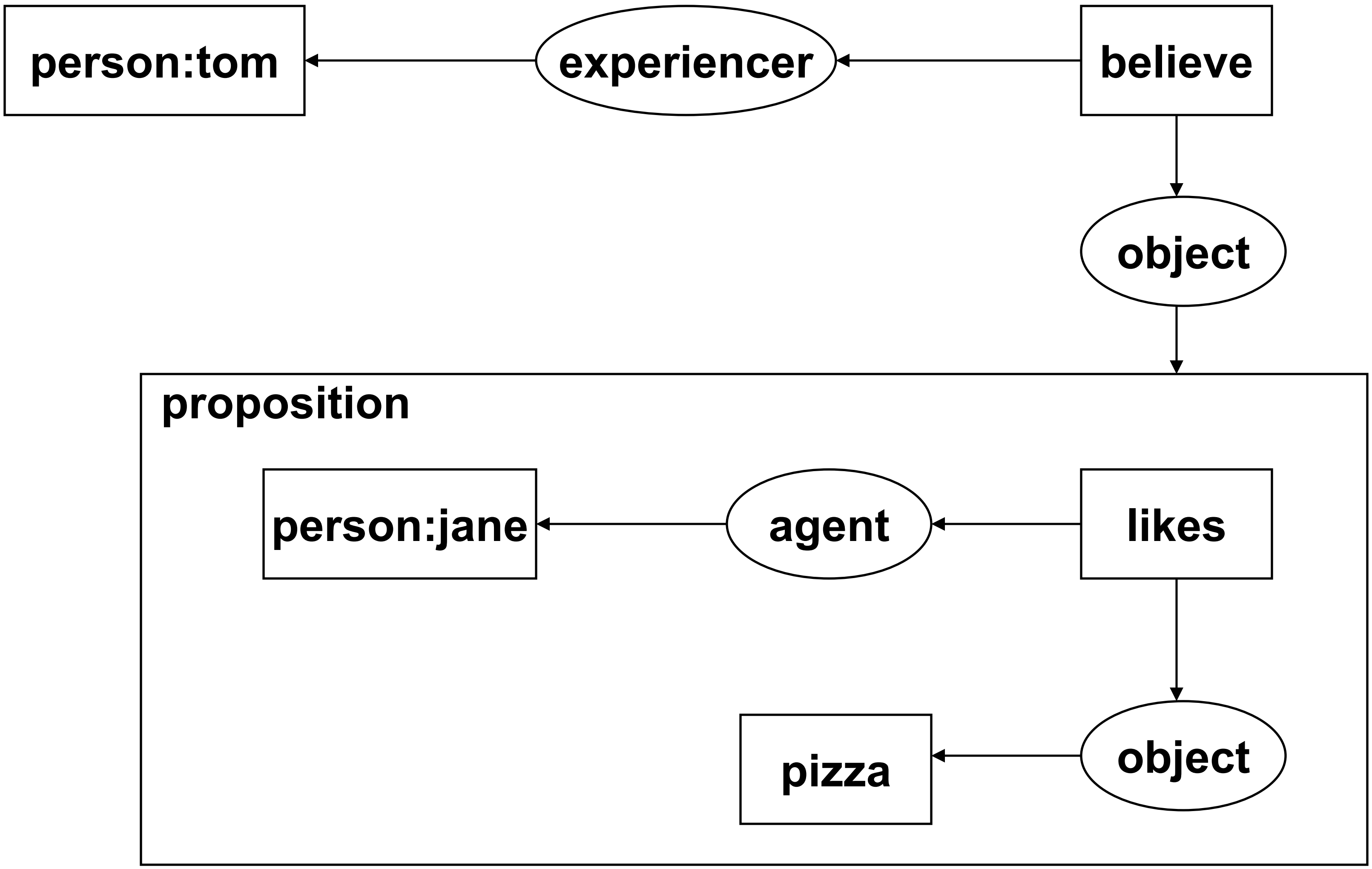
A conceptual graph

Inheritance of a property by a subclass

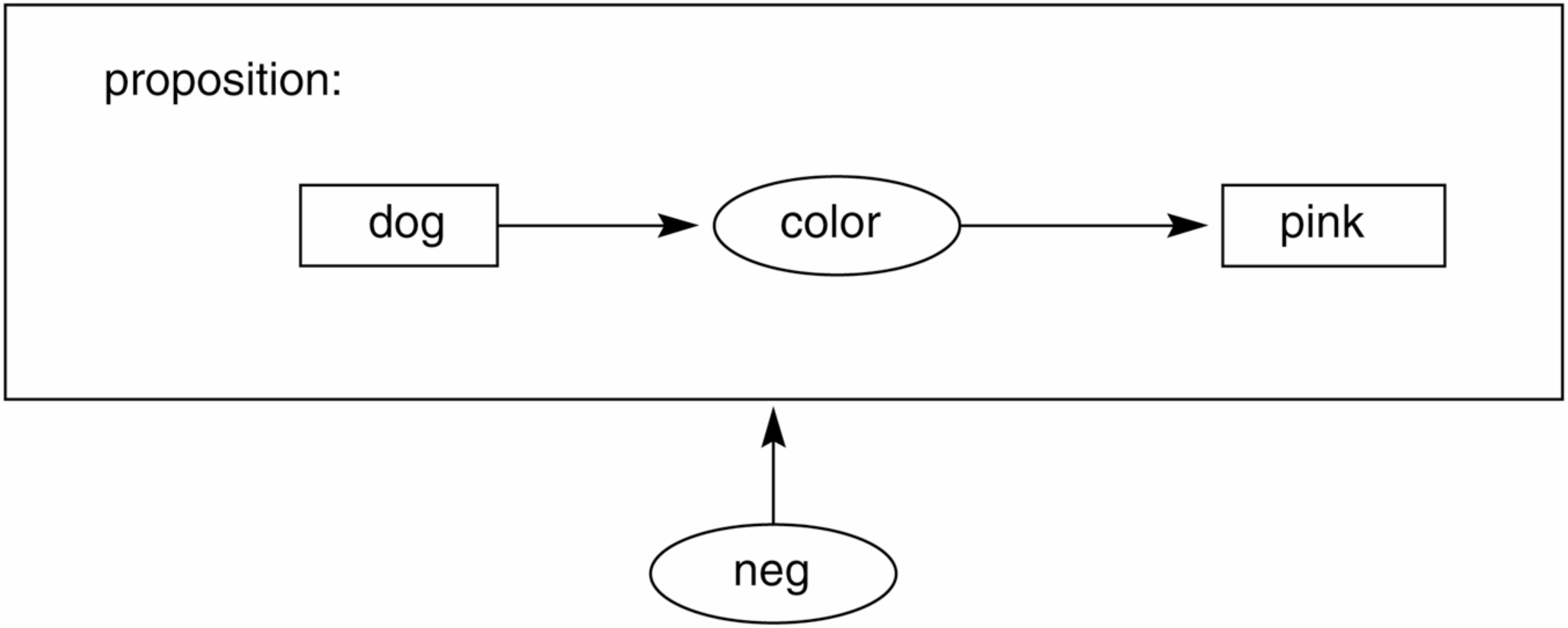


Inheritance of a property by an individual

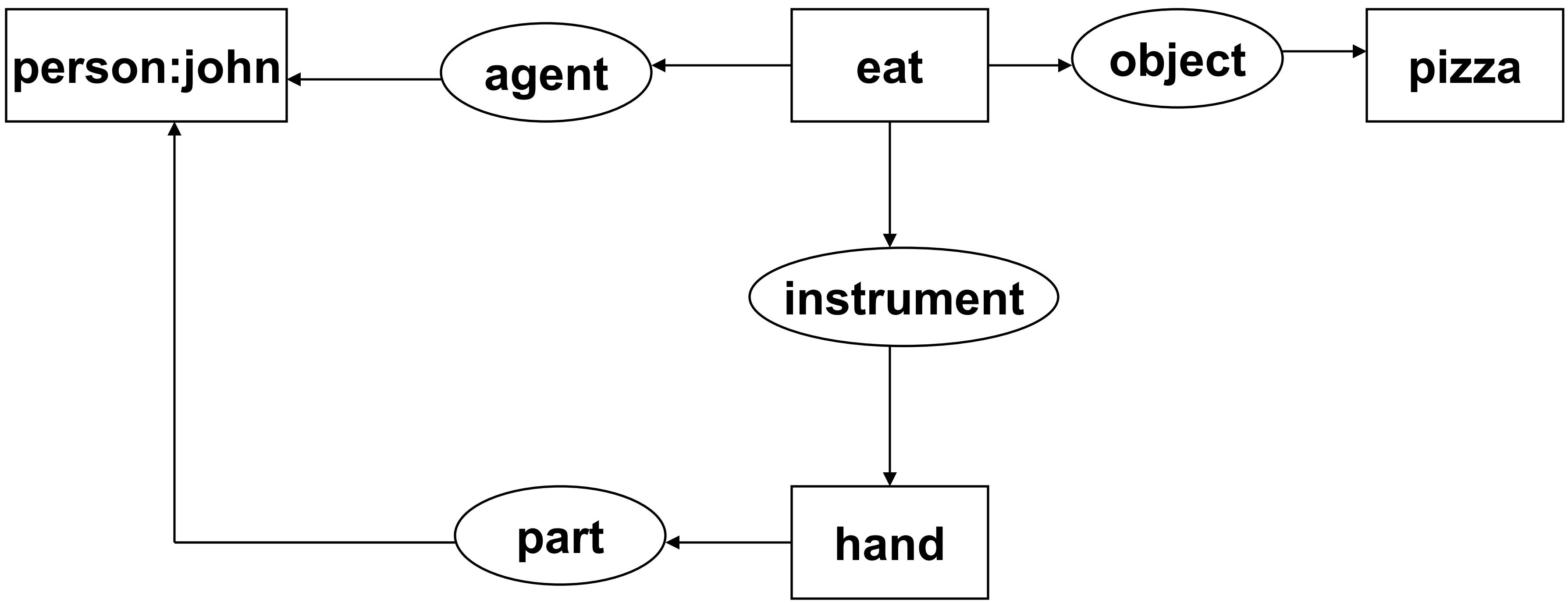
“Tom believes that Jane likes pizza.”



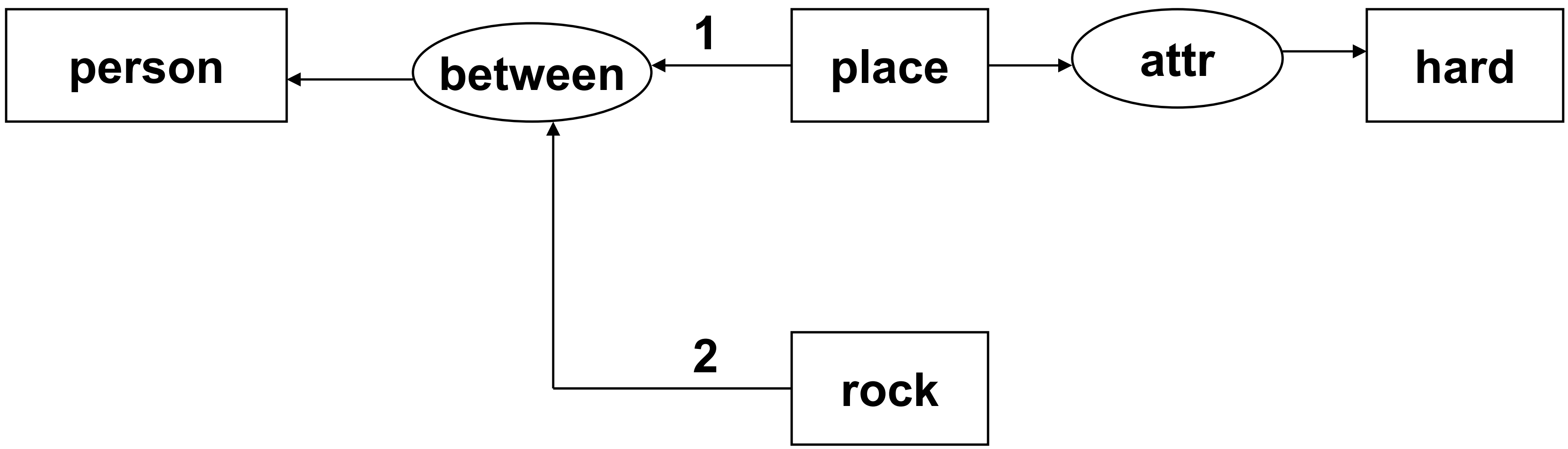
“There are no pink dogs.”



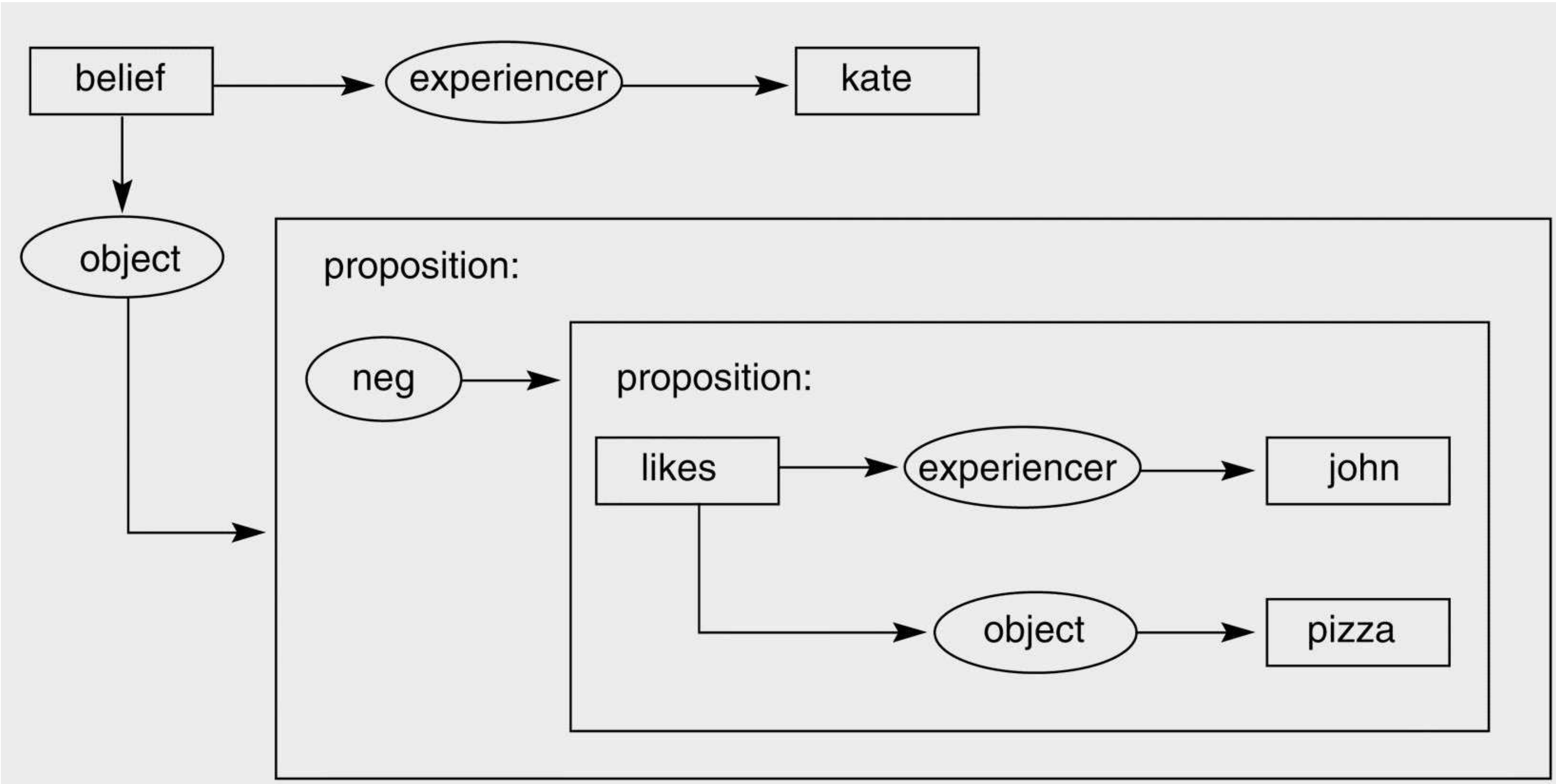
Translate into English



Translate into English



Translate into English



Algorithm to convert a conceptual graph, g, to a predicate calculus expression

1. Assign a unique variable, x_1, x_2, …, x_n, to each one of the n generic concepts in g.

2. Assign a unique constant to each individual constant in g. This constant may simply be the name or marker used to indicate the referent of the concept.

3. Represent each concept by a unary predicate with the same name as the type of that node and whose argument is the variable or constant given that node.

4. Represent each n-ary conceptual relation in g as an n-ary predicate whose name is the same as the relation. Let each argument of the predicate be the variable or constant assigned to the corresponding concept node linked to that relation.

5. Take the conjunction of all the atomic sentences formed under 3 and 4. This is the body of the predicate calculus expression. All the variables in the expression are existentially quantified.

Example conversion

1. Assign variables to generic concepts — X_1

2. Assign constants to individual concepts — emma

3. Represent each concept node — dog(emma) — brown(X_1)

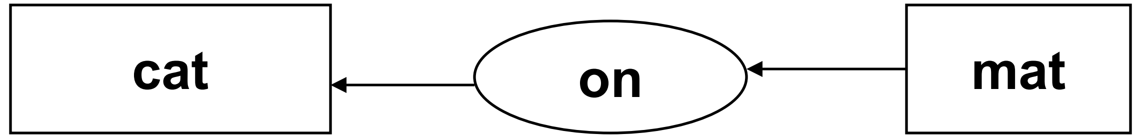
4. Represent each n-ary relation — color(emma, X_1)

5. Take the conjunction all the predicates from 3 and 4 — $\text{dog(emma)} \wedge \text{color(emma, } X_1) \wedge \text{brown}(X_1)$

All the variables are existentially quantified. — $\exists X_1\ \text{dog(emma)} \wedge \text{color(emma, } X_1) \wedge \text{brown}(X_1)$

Universal quantification

A cat is on a mat.



Every cat is on a mat.