Structured Knowledge

Chapter 7

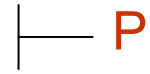
Logic Notations

Does logic represent well knowledge in structures?

Logic Notations

Frege's *Begriffsschrift* (concept writing) - 1879:

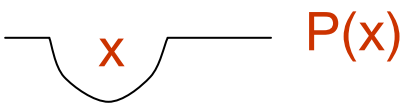
assert P



not P

if P then Q

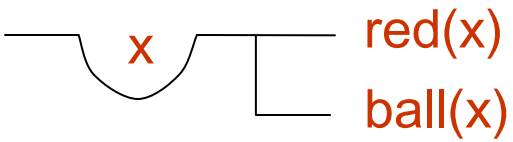
for every x, P(x)



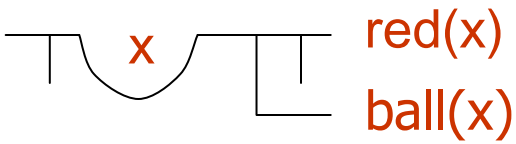
Logic Notations

Frege's *Begriffsschrift* (concept writing) - 1879:

"Every ball is red"



"Some ball is red"



Logic Notations

Algebraic notation - Peirce, 1883:

Universal quantifier: $\Pi_x P_x$

Existential quantifier: $\sum_x P_x$

Logic Notations

Algebraic notation - Peirce, 1883:

"Every ball is red": $\prod_x(ball_x \prec red_x)$

"Some ball is red": $\sum_x(ball_x \bullet red_x)$

Logic Notations

Peano's and later notation:

"Every ball is red": $(\forall x)(\text{ball}(x) \supset \text{red}(x))$

"Some ball is red": $(\exists x)(\text{ball}(x) \wedge \text{red}(x))$

Logic Notations

Existential graphs - Peirce, 1897:

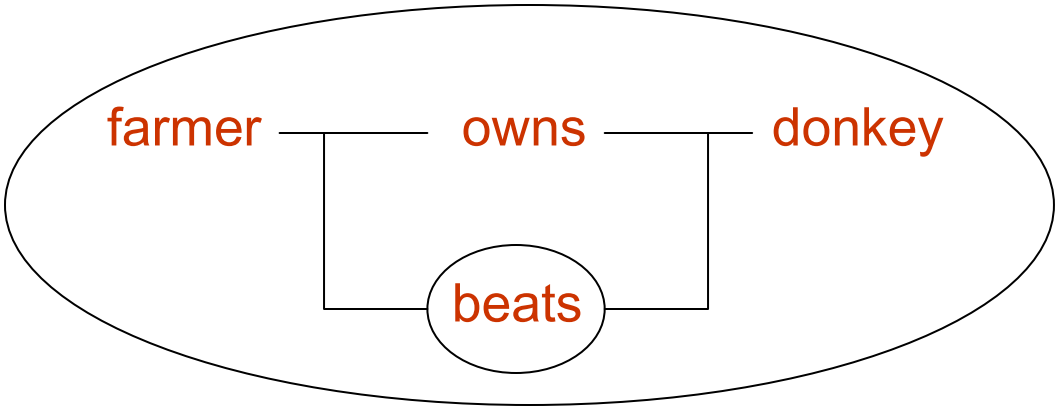
Existential quantifier: a link structure of bars, called line of identity, represents ∃

Conjunction: the juxtaposition of two graphs represents ∧

Negation: an oval enclosure represents ¬

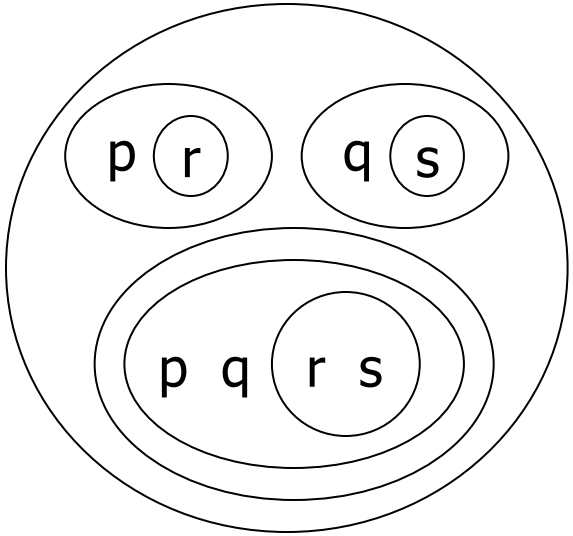
Logic Notations

"If a farmer owns a donkey, then he beats it":



Logic Notations

$$((p \rightarrow r) \wedge (q \rightarrow s)) \rightarrow ((p\ q) \rightarrow (r\ s))$$



Existential Graphs

EG's rules of inferences:

Erasure: in a positive context, any graph may be erased.

Insertion: in a negative context, any graph may be inserted.

Iteration: a copy of a graph may be written in the same context or any nested context.

Deiteration: any graph may be erased if a copy of its occurs in the same context or a containing context.

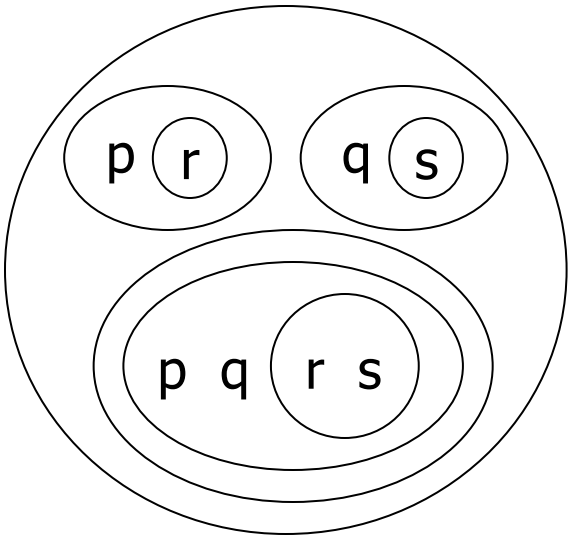
Double negation: two negations with nothing between them may be erased or inserted.

Existential Graphs

Prove: $((p \rightarrow r) \wedge (q \rightarrow s)) \rightarrow ((p\ q) \rightarrow (r\ s))$ is valid

Existential Graphs

Prove: $((p \rightarrow r) \wedge (q \rightarrow s)) \rightarrow ((p\ q) \rightarrow (r\ s))$ is valid



Existential Graphs

Prove: $((p \rightarrow r) \wedge (q \rightarrow s)) \rightarrow ((p\ q) \rightarrow (r\ s))$

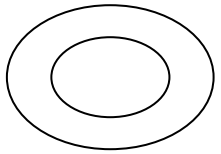
Erasure: in a positive context, any graph may be erased.

Insertion: in a negative context, any graph may be inserted.

Iteration: a copy of a graph may be written in the same context or any nested context.

Deiteration: any graph may be erased if a copy of its occurs in the same context or a containing context.

Double negation: two negations with nothing between them may be erased or inserted.



Existential Graphs

Prove: $((p \rightarrow r) \wedge (q \rightarrow s)) \rightarrow ((p\ q) \rightarrow (r\ s))$

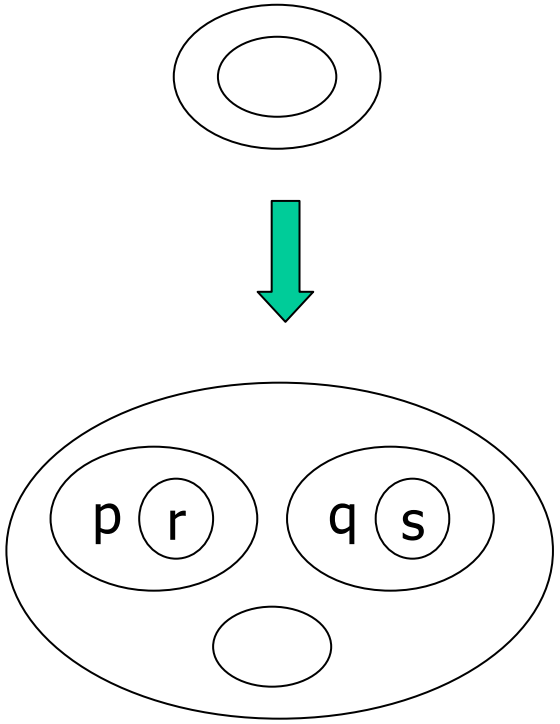
Erasure: in a positive context, any graph may be erased.

Insertion: in a negative context, any graph may be inserted.

Iteration: a copy of a graph may be written in the same context or any nested context.

Deiteration: any graph may be erased if a copy of its occurs in the same context or a containing context.

Double negation: two negations with nothing between them may be erased or inserted.



Existential Graphs

Prove: $((p \rightarrow r) \wedge (q \rightarrow s)) \rightarrow ((p\ q) \rightarrow (r\ s))$

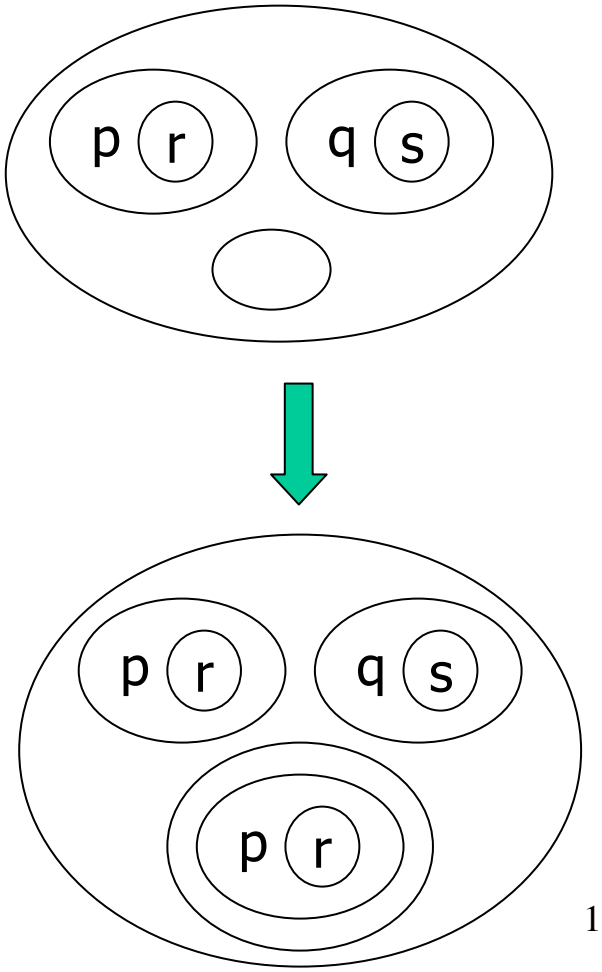
Erasure: in a positive context, any graph may be erased.

Insertion: in a negative context, any graph may be inserted.

Iteration: a copy of a graph may be written in the same context or any nested context.

Deiteration: any graph may be erased if a copy of its occurs in the same context or a containing context.

Double negation: two negations with nothing between them may be erased or inserted.



Existential Graphs

Prove: $((p \rightarrow r) \wedge (q \rightarrow s)) \rightarrow ((p\ q) \rightarrow (r\ s))$

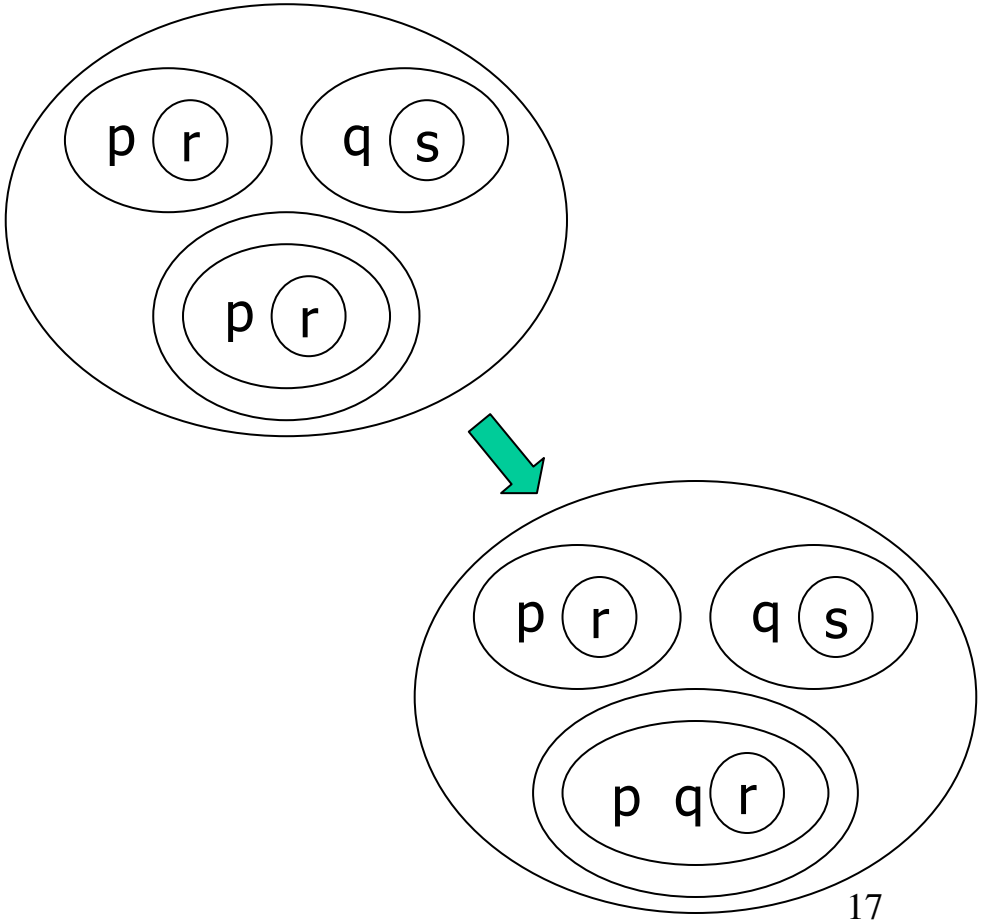
Erasure: in a positive context, any graph may be erased.

Insertion: in a negative context, any graph may be inserted.

Iteration: a copy of a graph may be written in the same context or any nested context.

Deiteration: any graph may be erased if a copy of its occurs in the same context or a containing context.

Double negation: two negations with nothing between them may be erased or inserted.



Existential Graphs

Prove: $((p \rightarrow r) \wedge (q \rightarrow s)) \rightarrow ((p\ q) \rightarrow (r\ s))$

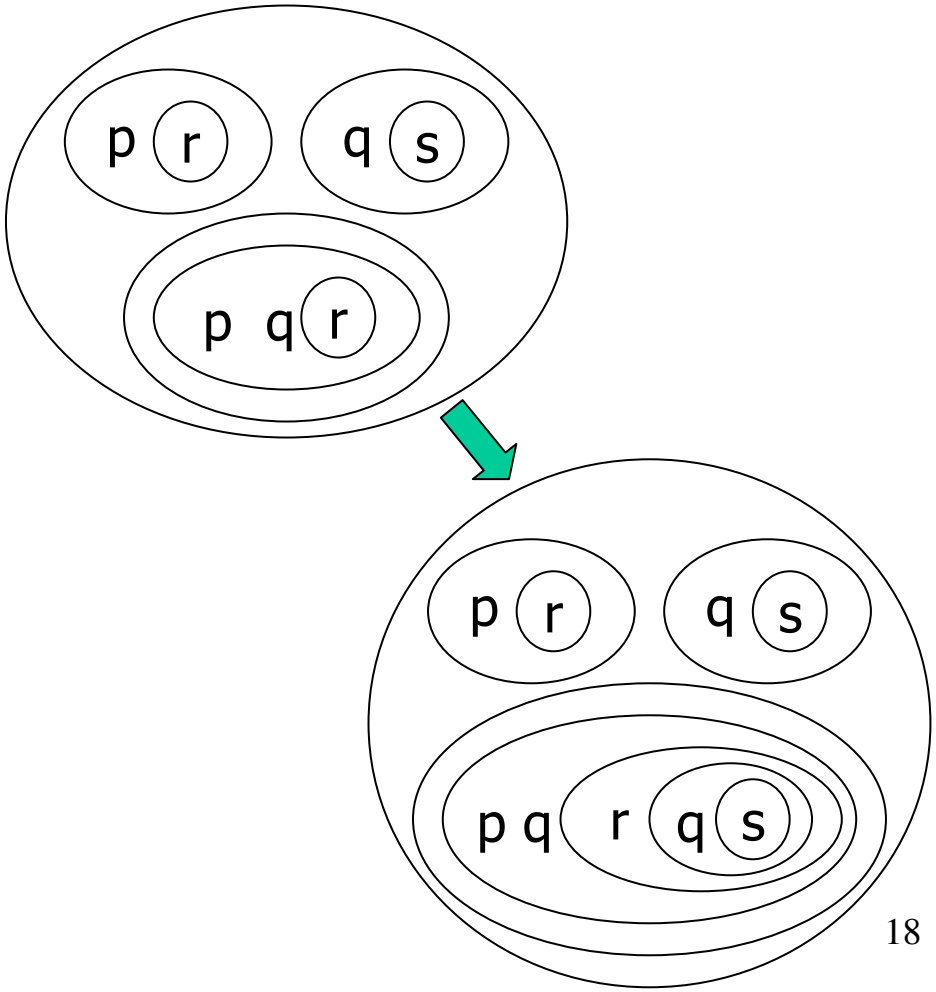
Erasure: in a positive context, any graph may be erased.

Insertion: in a negative context, any graph may be inserted.

Iteration: a copy of a graph may be written in the same context or any nested context.

Deiteration: any graph may be erased if a copy of its occurs in the same context or a containing context.

Double negation: two negations with nothing between them may be erased or inserted.



Existential Graphs

Prove: $((p \rightarrow r) \wedge (q \rightarrow s)) \rightarrow ((p\ q) \rightarrow (r\ s))$

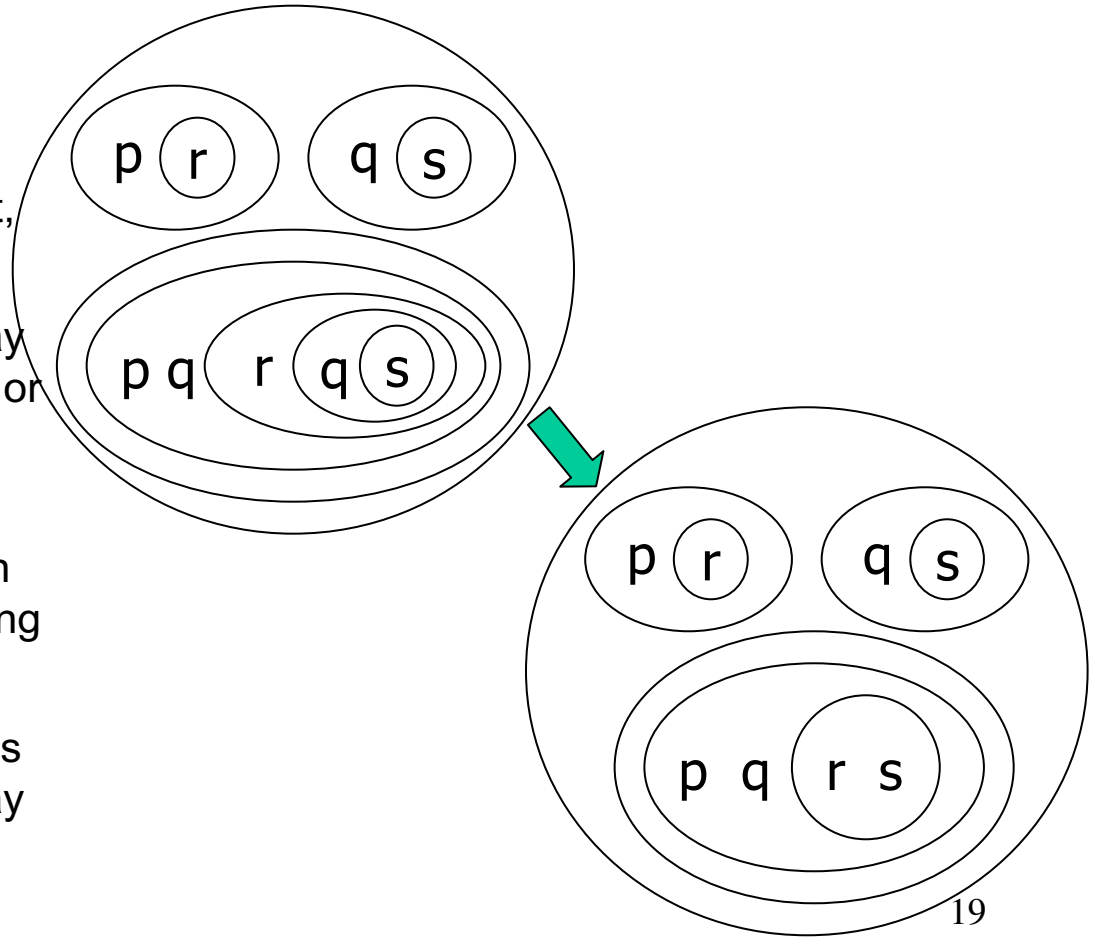
Erasure: in a positive context, any graph may be erased.

Insertion: in a negative context, any graph may be inserted.

Iteration: a copy of a graph may be written in the same context or any nested context.

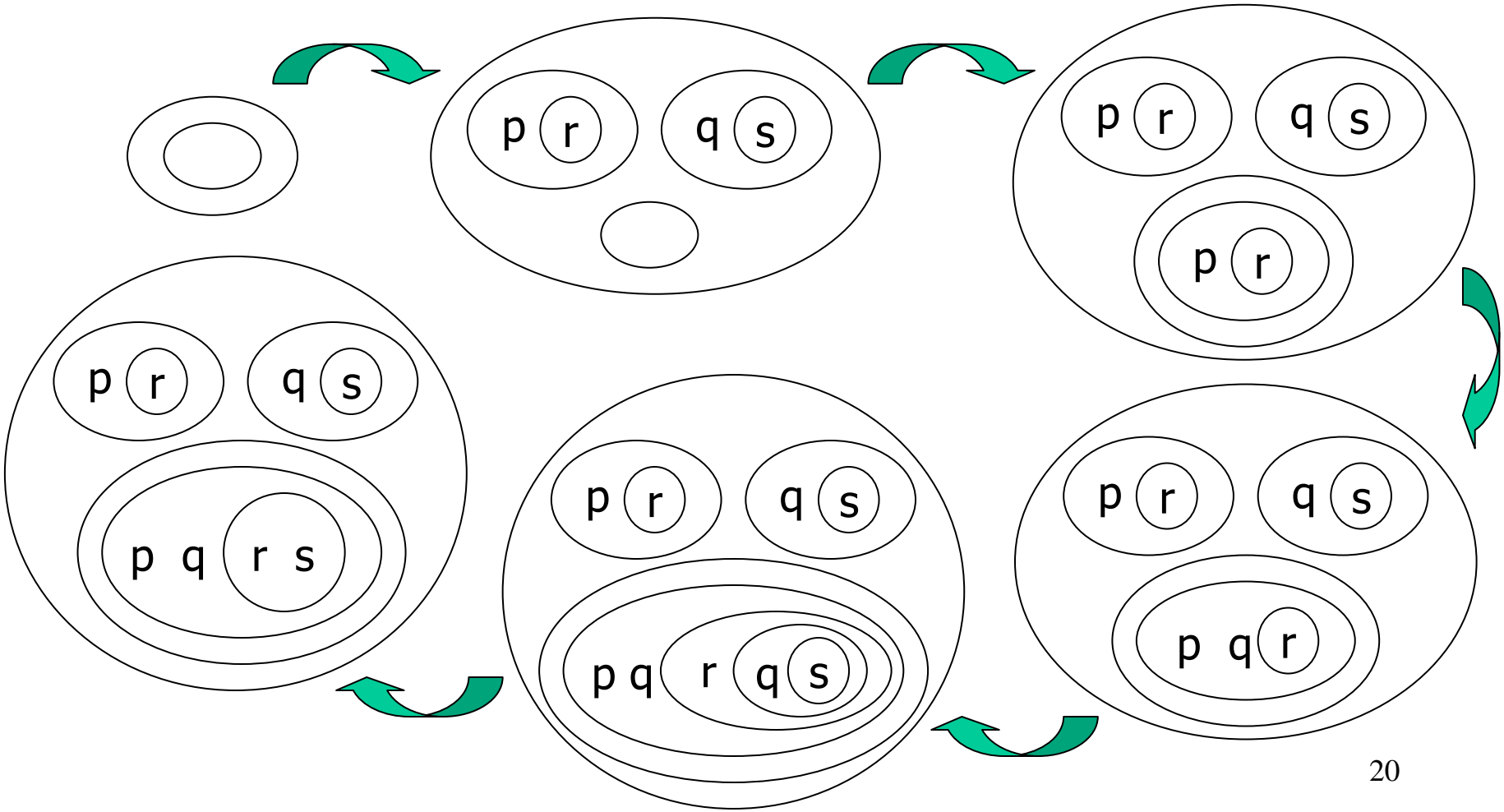
Deiteration: any graph may be erased if a copy of its occurs in the same context or a containing context.

Double negation: two negations with nothing between them may be erased or inserted.



Existential Graphs

Prove: $((p \rightarrow r) \wedge (q \rightarrow s)) \rightarrow ((p\ q) \rightarrow (r\ s))$



Existential Graphs

- α-graphs: propositional logic
- β-graphs: first-order logic
- γ-graphs: high-order and modal logic

Semantic Nets

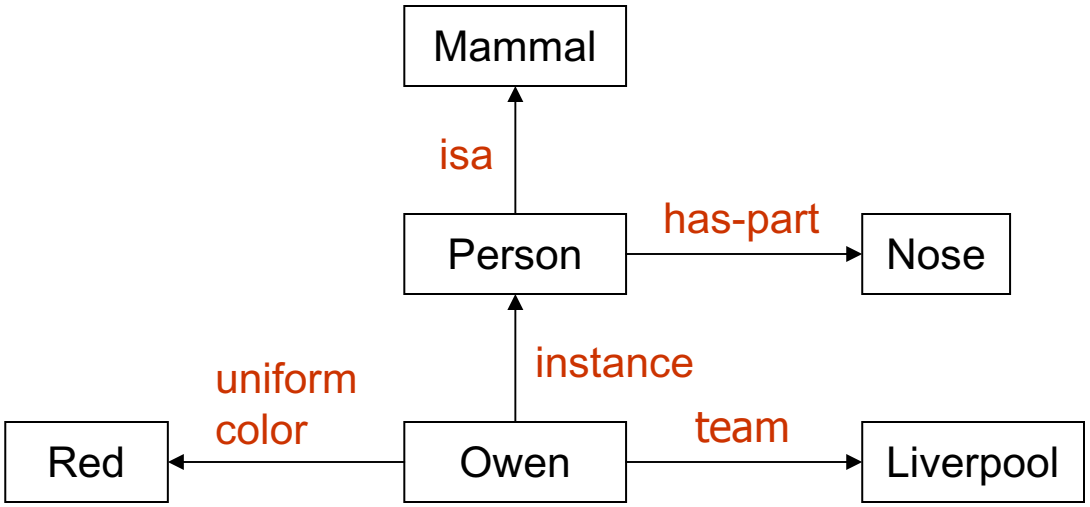
- Since the late 1950s dozens of different versions of semantic networks have been proposed, with various terminologies and notations.
- The main ideas:

  For representing knowledge in structures

  The meaning of a concept comes from the ways it is connected to other concepts

  Labelled nodes representing concepts are connected by labelled arcs representing relations

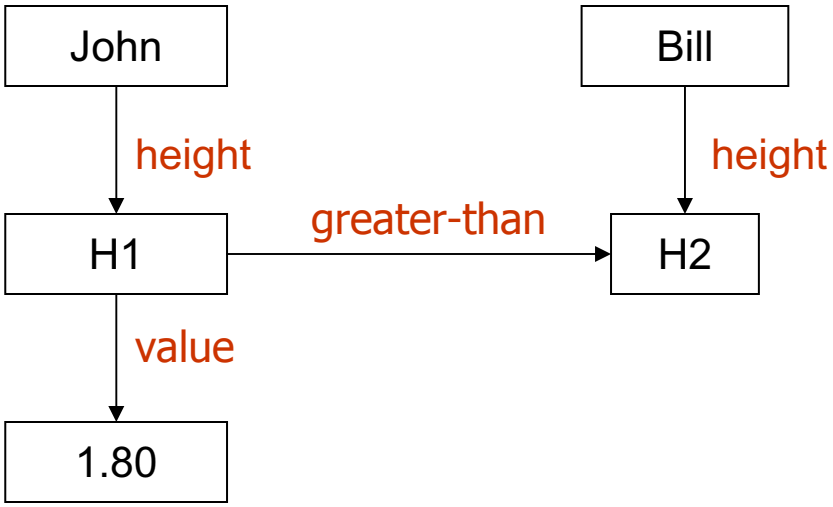
Semantic Nets



person(Owen) ≡ instance(Owen, Person)

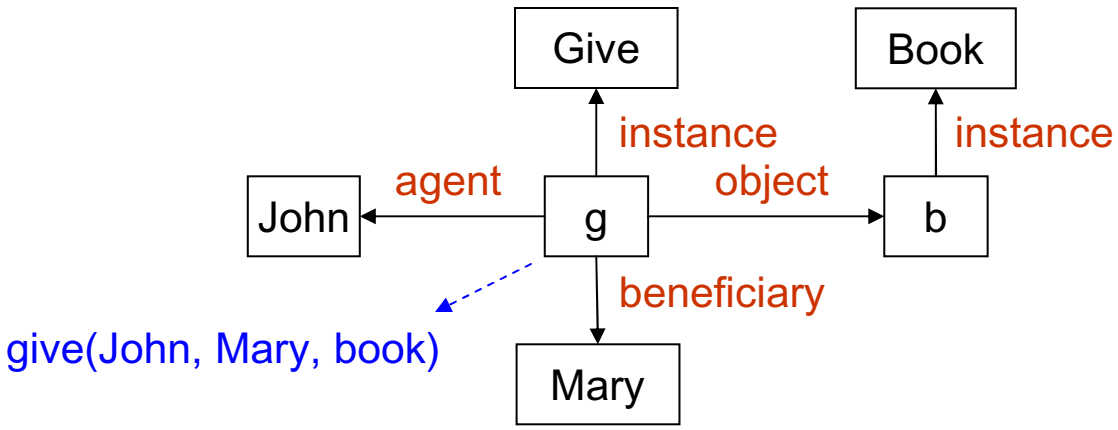
team(Owen, Liverpool)

Semantic Nets



Semantic Nets

"John gives Mary a book"

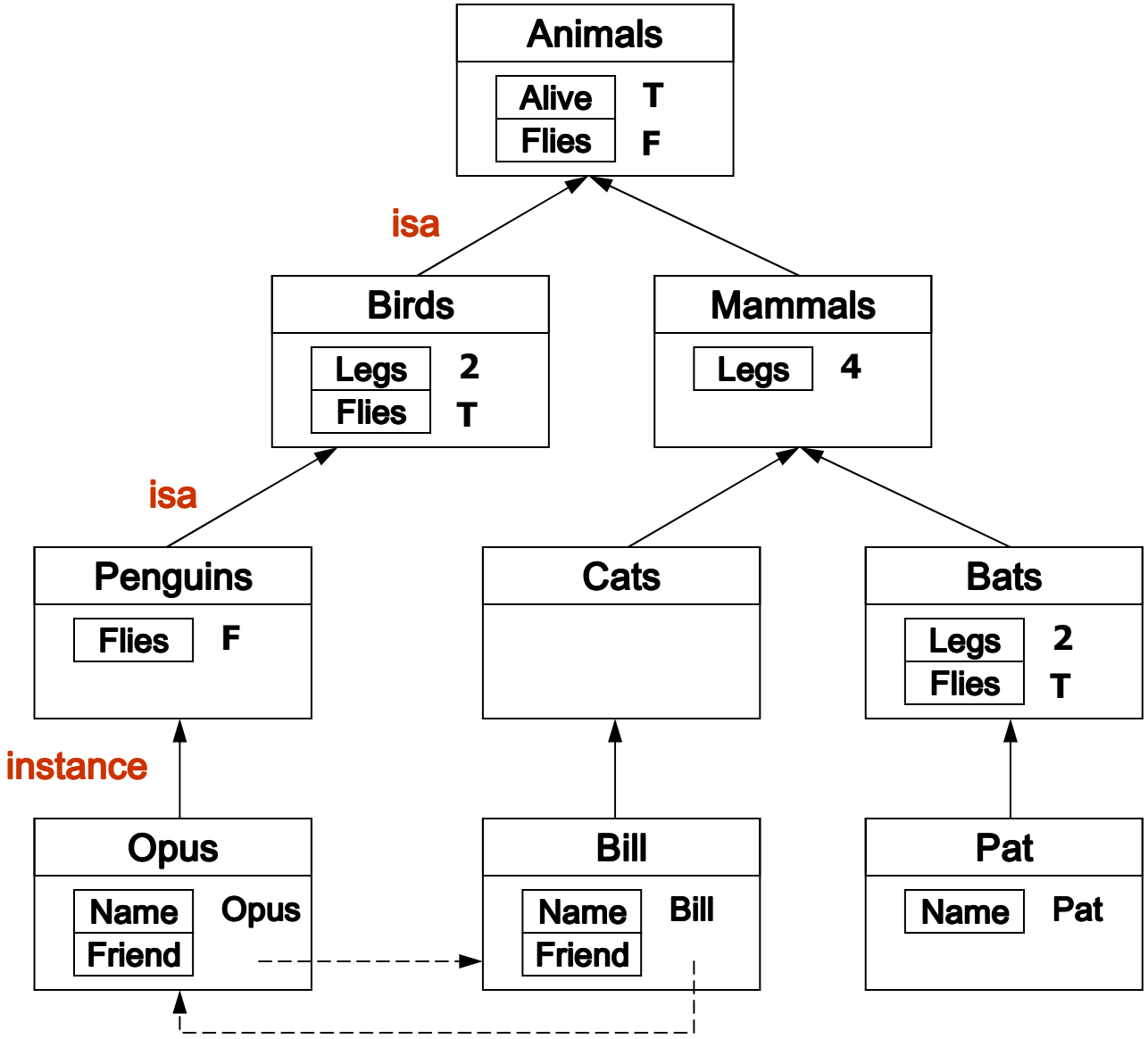


Frames

- A vague paradigm - to organize knowledge in high-level structures
- “A Framework for Representing Knowledge” - Minsky, 1974
- Knowledge is encoded in packets, called frames (single frames in a film)

Frame name + slots

Frames


Animals
Alive T
Flies F
isa
Birds
Legs 2
Flies T
Mammals
Legs 4
isa
Penguins
Flies F
Cats
Bats
Legs 2
Flies T
instance
Opus
Name Opus
Friend
Bill
Name Bill
Friend
Pat
Name Pat

Frames

Hybrid systems:

Frame component: to define terminologies (predicates and terms)

Predicate calculus component: to describe individual objects and rules

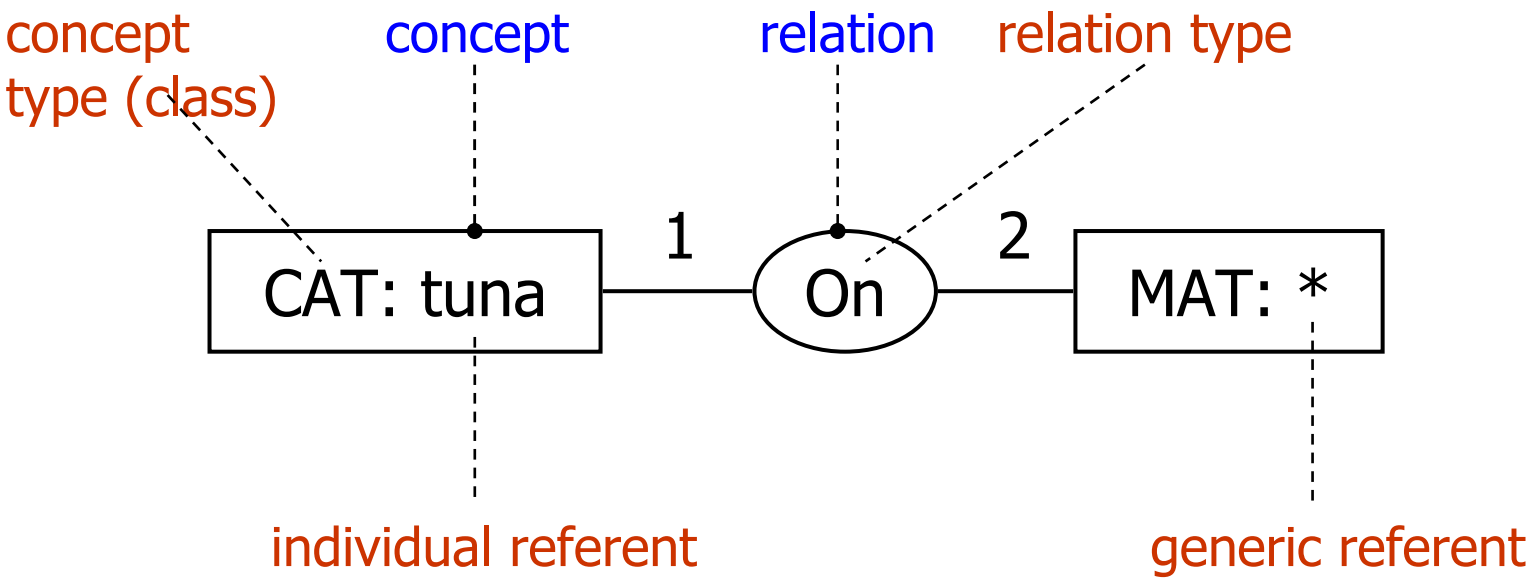
Conceptual Graphs

- Sowa, J.F. 1984. Conceptual Structures: Information Processing in Mind and Machine.
- CG = a combination of Perice's EGs and semantic networks.

Conceptual Graphs

- 1968: term paper to Marvin Minsky at Harvard.
- 1970's: seriously working on CGs
- 1976: first paper on CGs
- 1981-1982: meeting with Norman Foo,
  finding Peirce's EGs
- 1984: the book coming out
- CG homepage: http://conceptualgraphs.org/

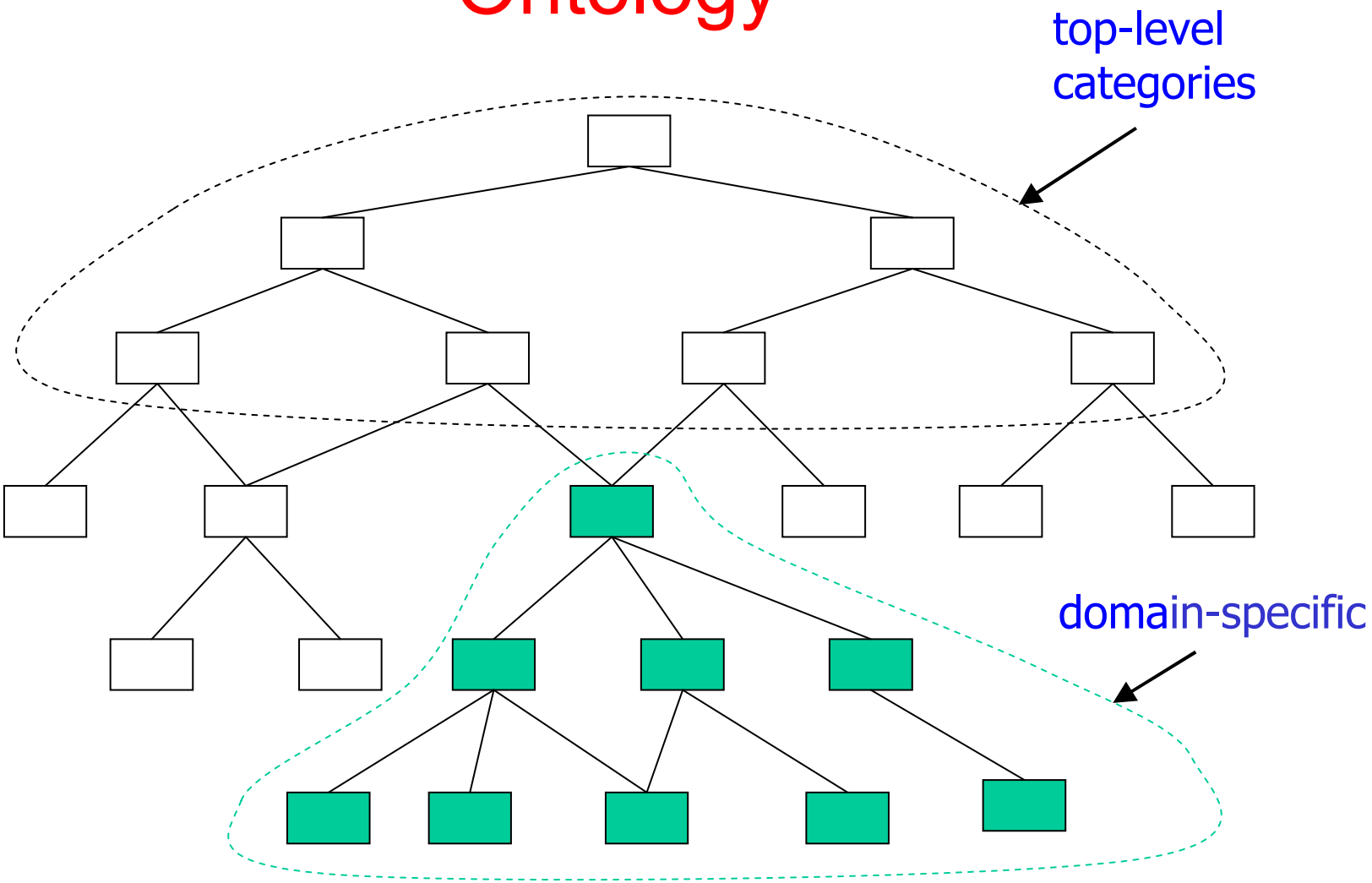
Simple Conceptual Graphs



Ontology

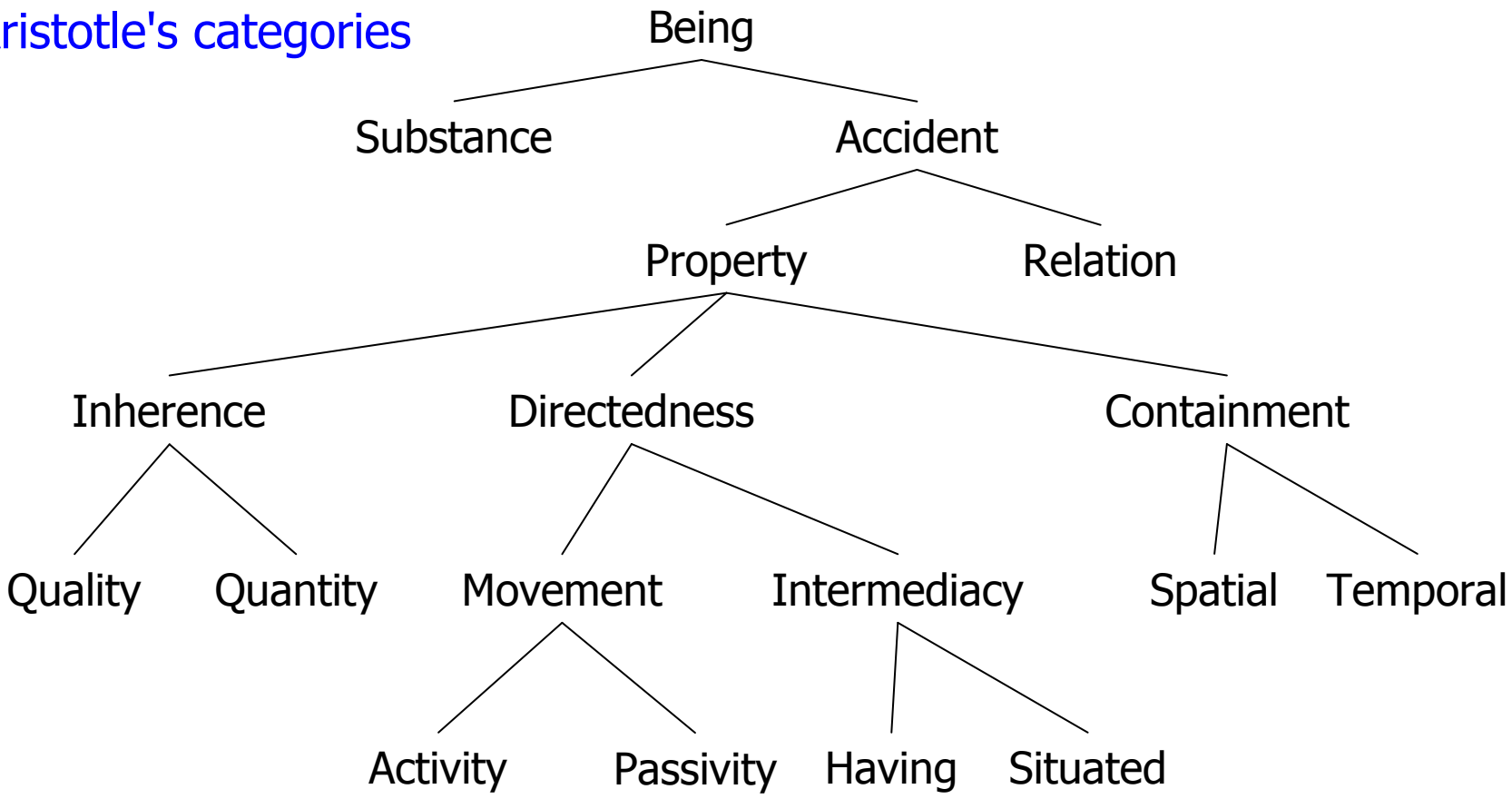
- Ontology: the study of "being" or existence
- An ontology = "A catalog of types of things that are assumed to exist in a domain of interest" (Sowa, 2000)
- An ontology = "The arrangement of kinds of things into types and categories with a well-defined structure" (Passin 2004)

Ontology

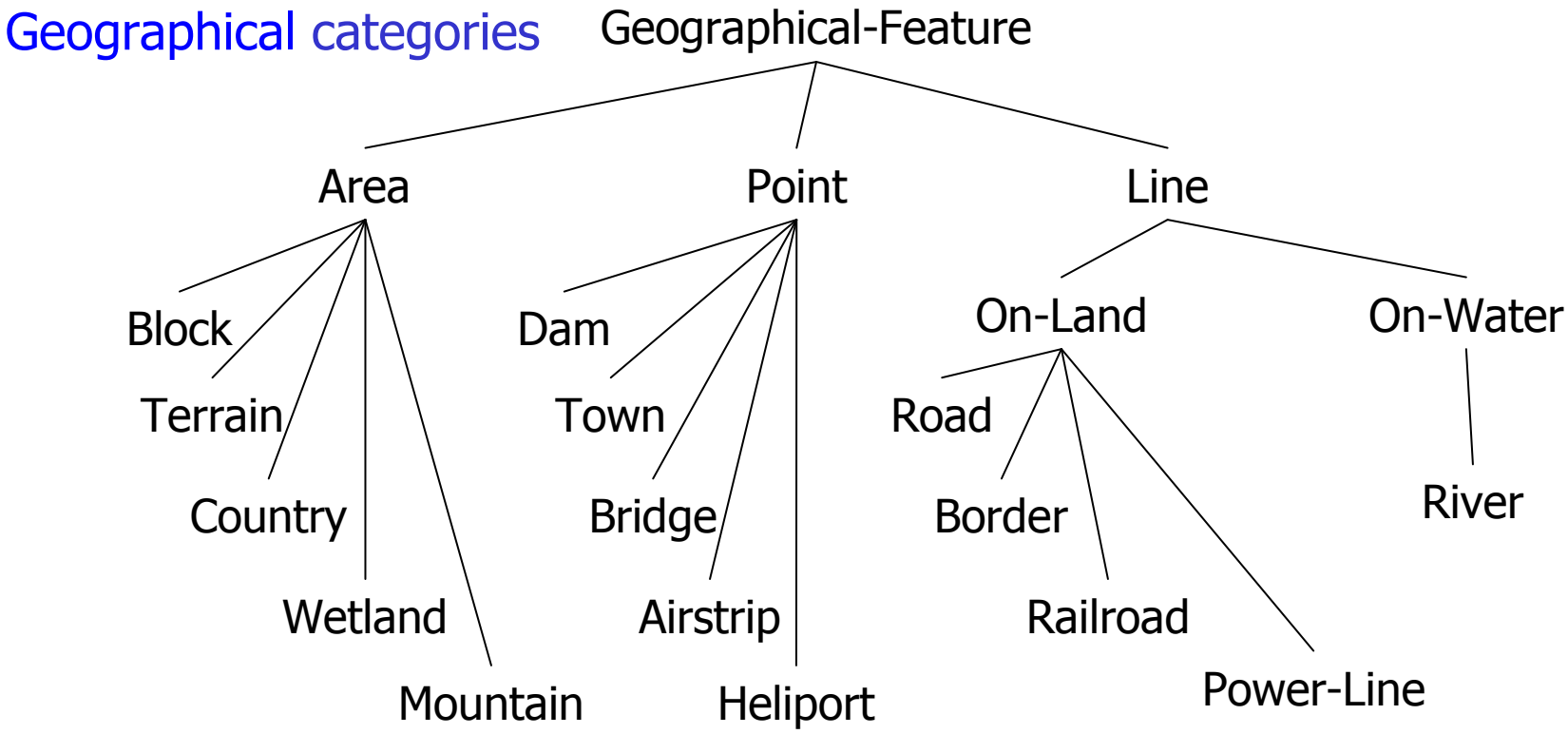


Ontology

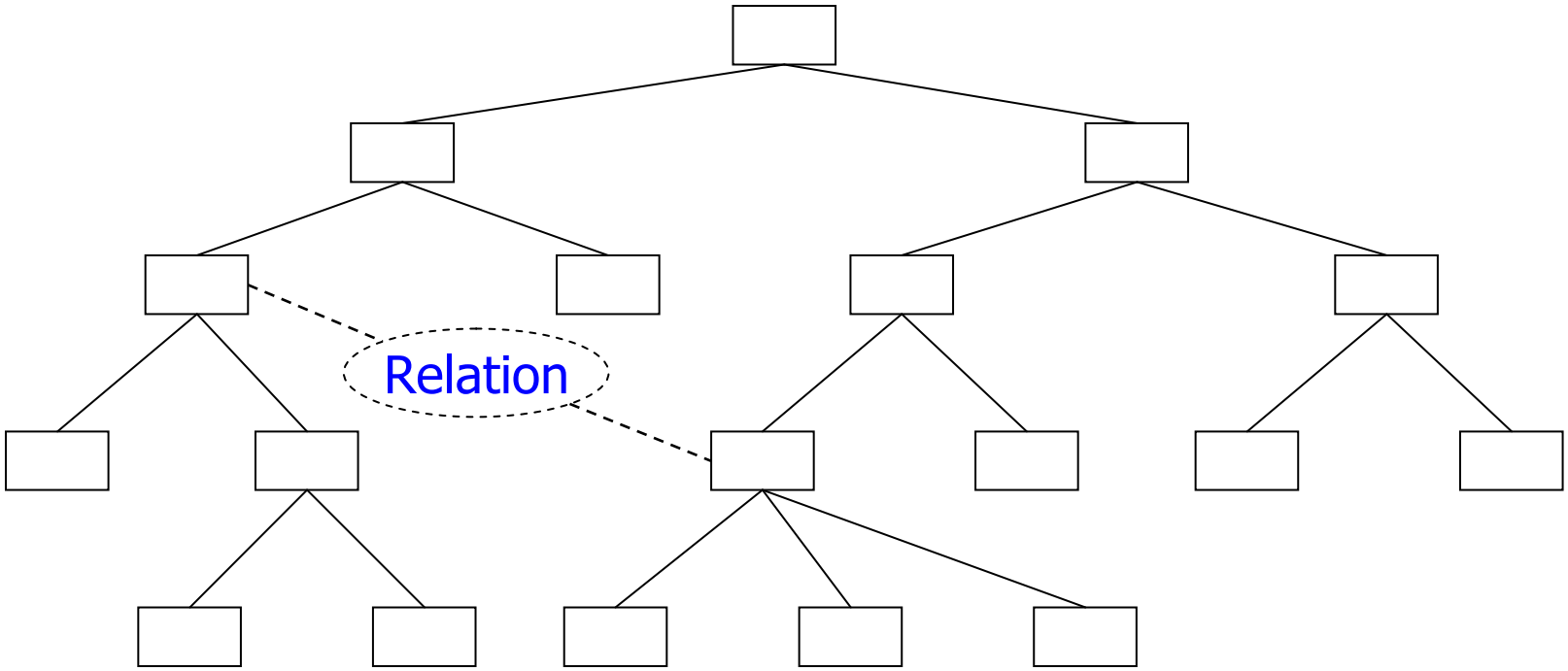
Aristotle's categories



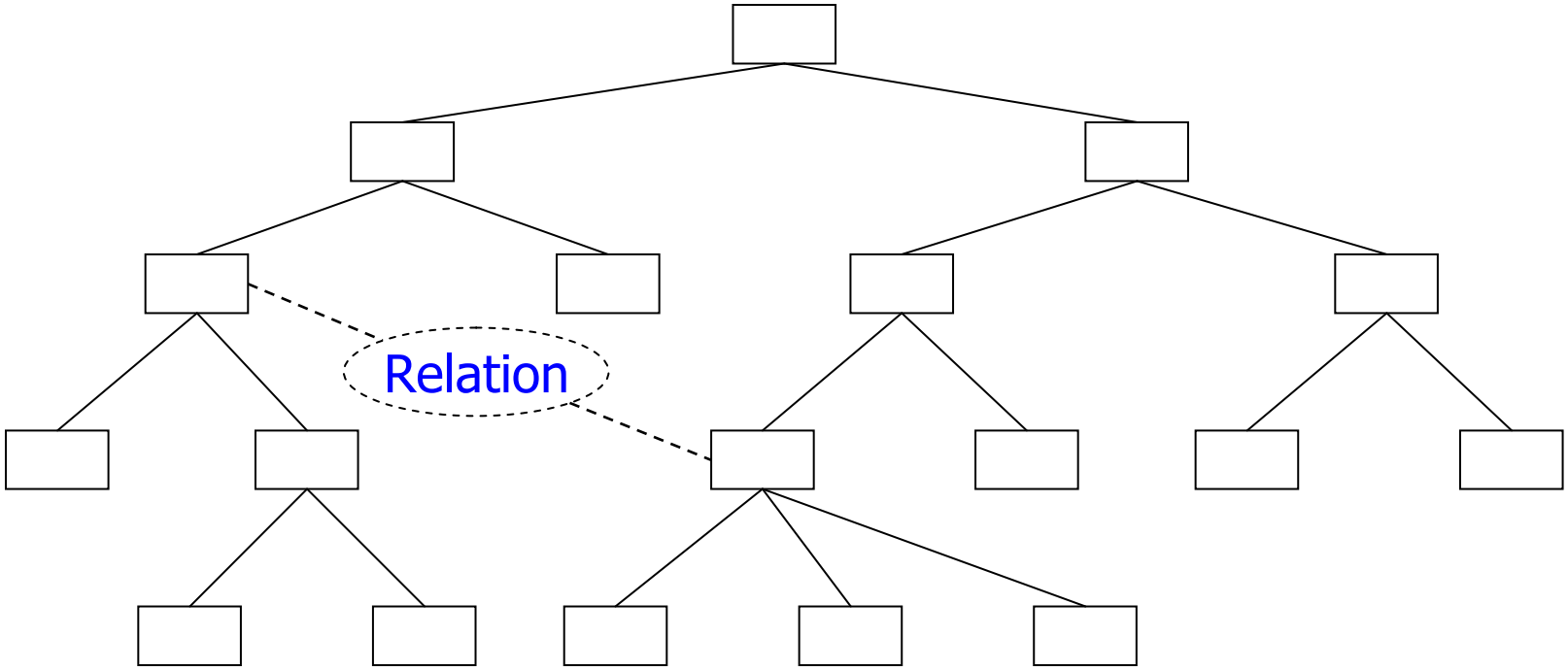
Ontology


Geographical categories
Geographical-Feature
Area
Point
Line
Block
Terrain
Country
Wetland
Mountain
Dam
Town
Bridge
Airstrip
Heliport
On-Land
On-Water
Road
Border
Railroad
Power-Line
River

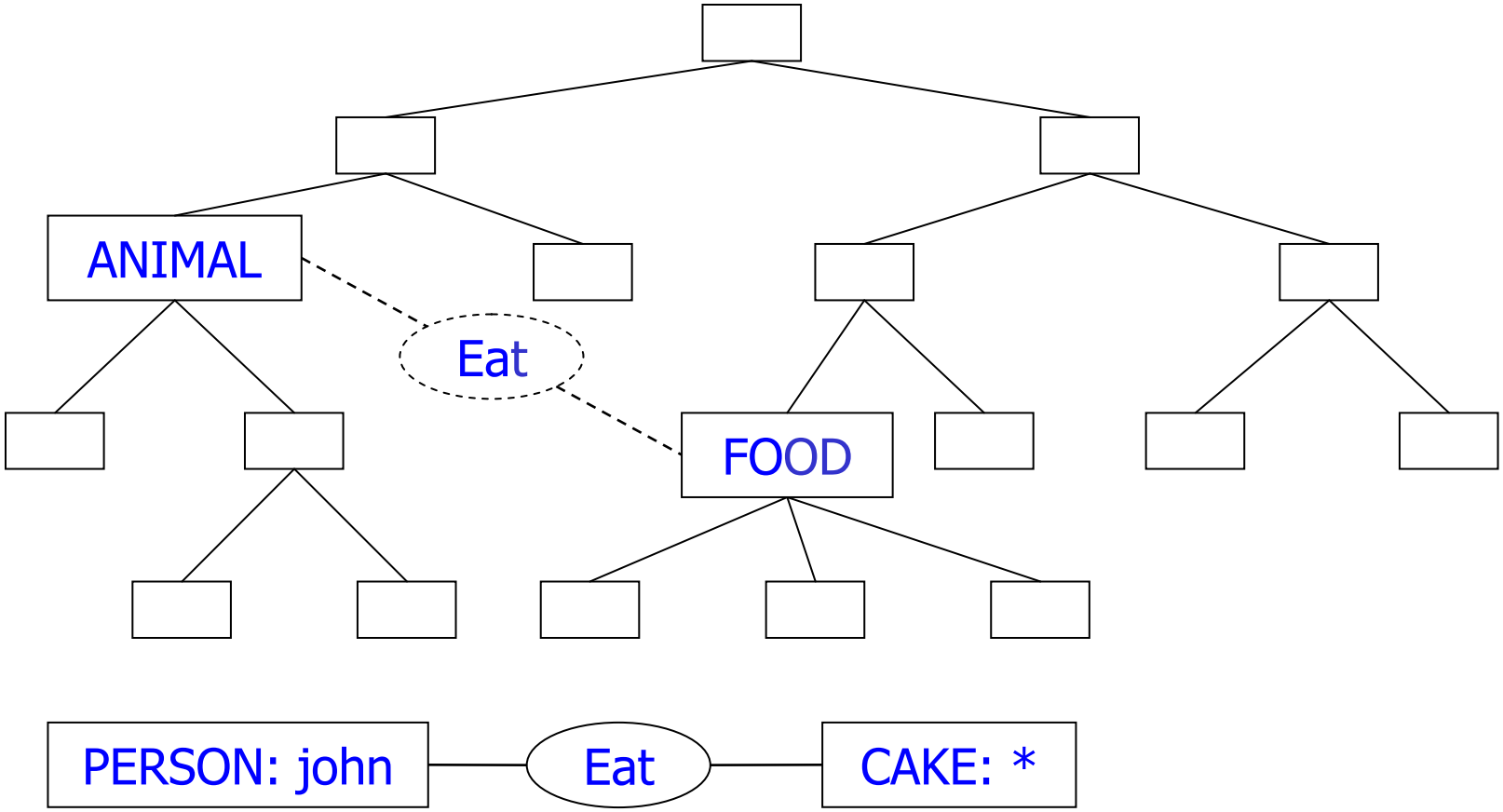
Ontology



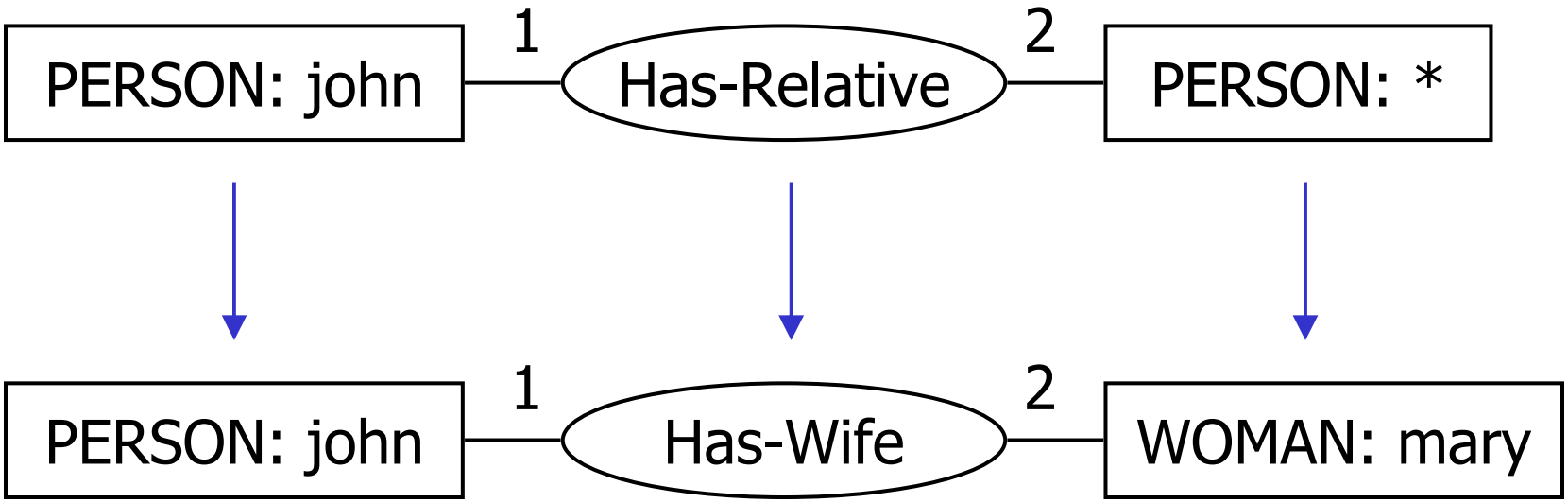
Ontology



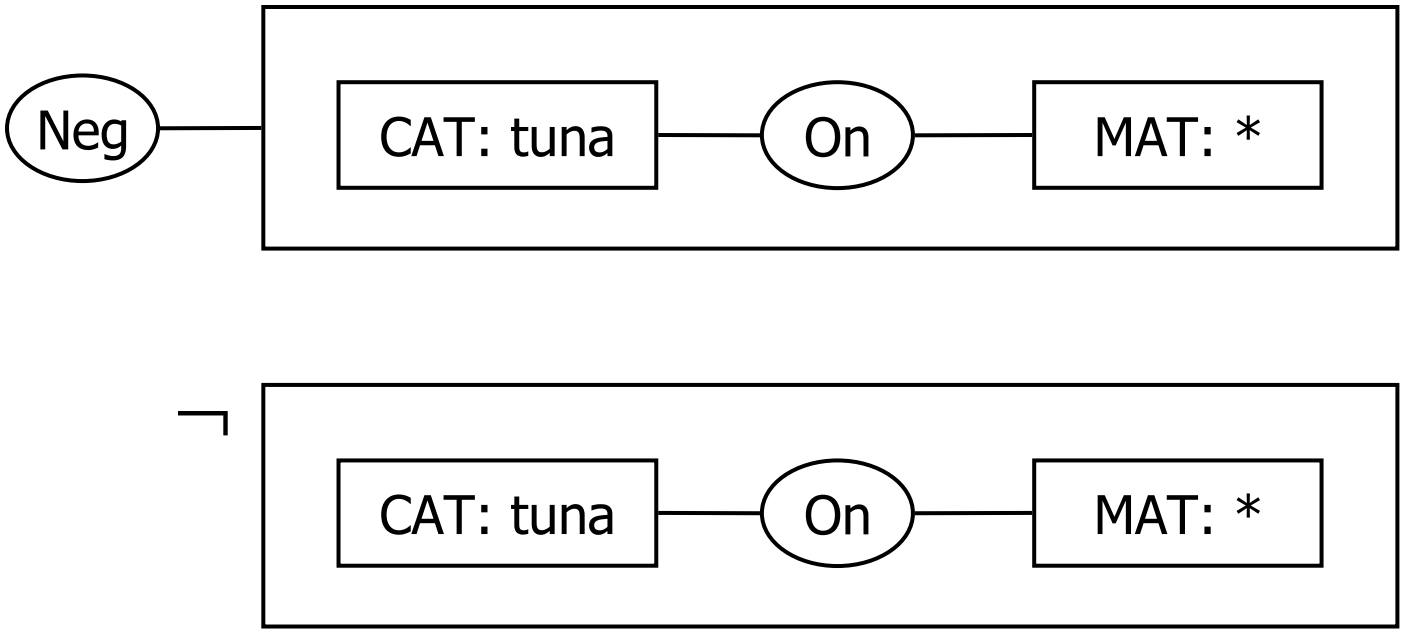
Ontology



CG Projection

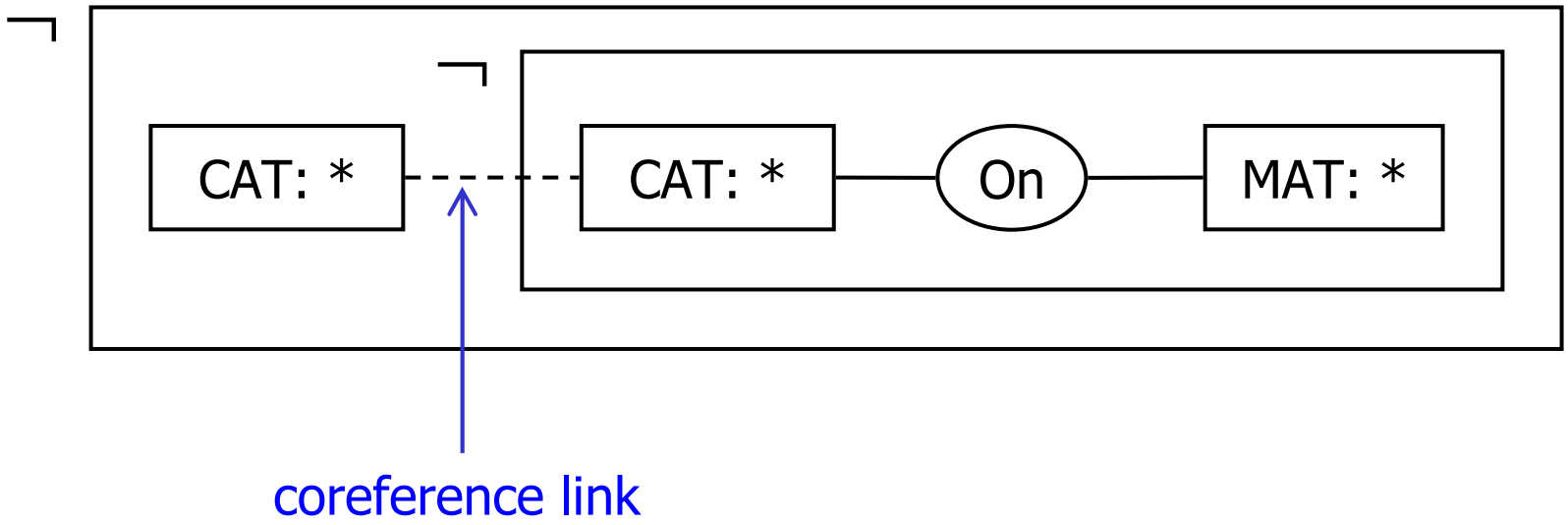


Nested Conceptual Graphs



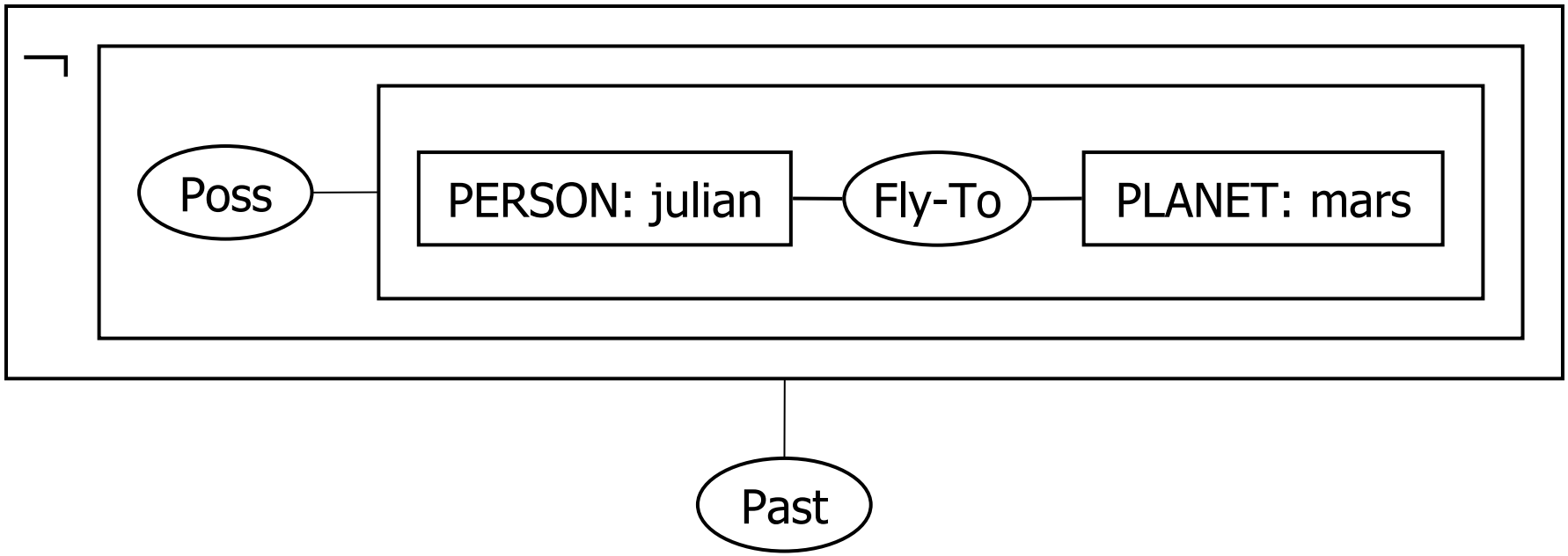
It is not true that cat Tuna is on a mat.

Nested Conceptual Graphs



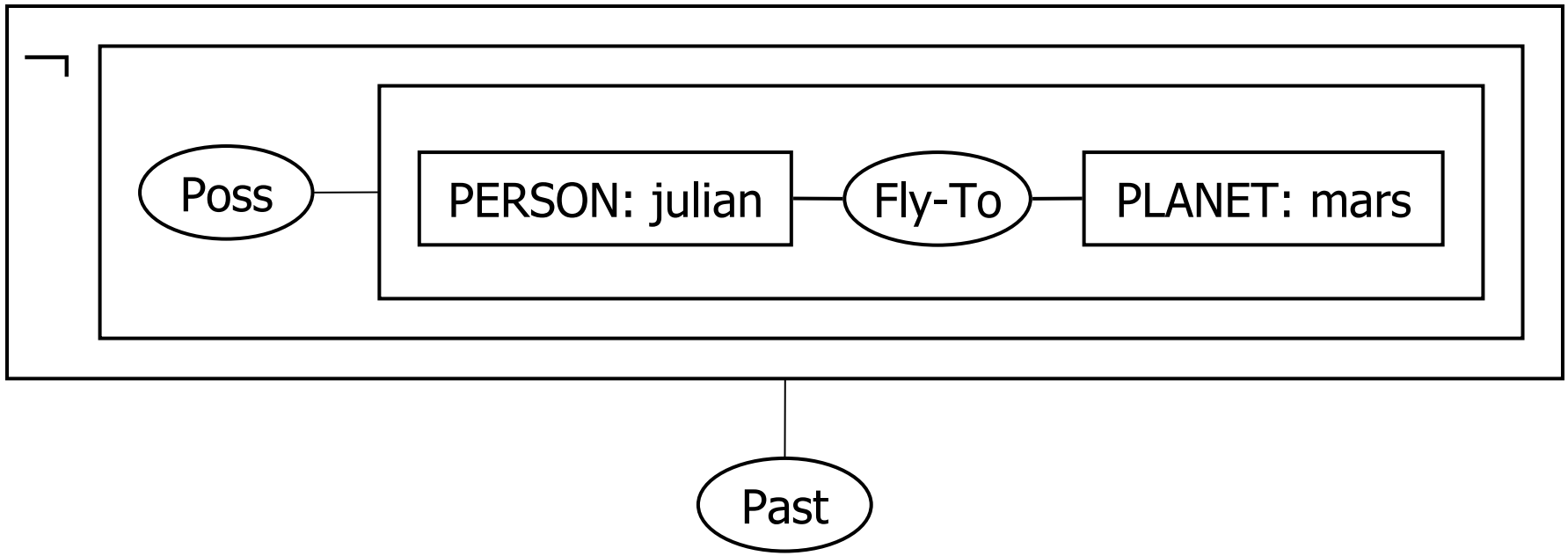
Every cat is on a mat

Nested Conceptual Graphs



Julian could not fly to Mars

Nested Conceptual Graphs



Tom believes that Mary wants to marry a sailor

Exercises

- Reading:

  Sowa, J.F. 2000. Knowledge Representation: Logical, Philosophical, and Computational Foundations (Section 1.1: history of logic).

  Way, E.C. 1994. Conceptual Graphs – Past, Present, and Future. Procs. of ICCS'94.