Existential Graphs

The simplest notation for logic ever invented

John F. Sowa

10 January 2020

Summary

Existential graphs (EGs) are an excellent pedagogical tool.

- **They may be related to other logics, linear or graphic, by means of a translation to and from CLIP, a dialect of Common Logic. ***

EG rules of inference are notation-independent:

- **The same rules may be used for propositional logic, first-order logic, and many variations and extensions of FOL.**
- **They may be used with graphic and linear notations, including any subset of natural language that can be translated to FOL.**

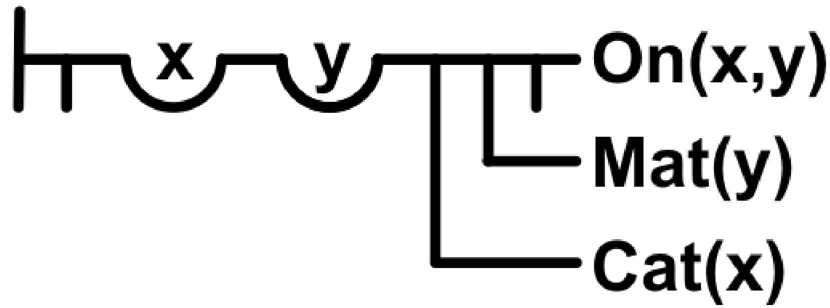
Peirce claimed that EGs represent "*a moving picture of the action of the mind in thought.*"

The psychologist Philip Johnson-Laird agreed: Peirce's EGs and rules of inference are a good candidate for a natural logic.

*** See "Language, Logic, and the Semantic Web": http://jfsowa.com/talks/eswc.pdf**

How to say “A cat is on a mat.”

Gottlob Frege (1879):

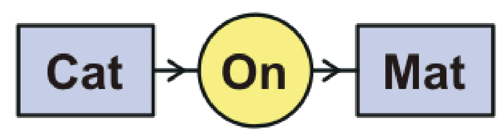


Charles Sanders Peirce (1885): $\Sigma_x \Sigma_y \text{Cat}_x \bullet \text{On}_{x,y} \bullet \text{Mat}_y$

Giuseppe Peano (1895): $\exists x \, \exists y \, \text{Cat}(x) \wedge \text{On}(x, y) \wedge \text{Mat}(y)$

Existential graph by Peirce (1897): Cat —— On —— Mat

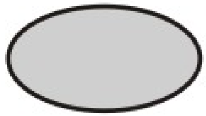
Conceptual graph (1976):



CLIP dialect of Common Logic: (∃ x y) (Cat x) (On x y) (Mat y).

Existential Graph Notation (1911)

Existence: —

Negation: 

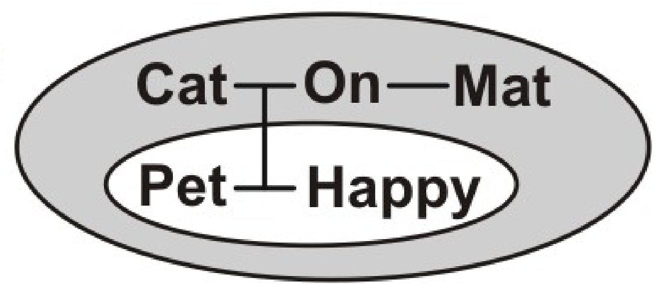
Relations: Cat• Mat• Happy• •On• •Under• •Give•

A cat is on a mat: Cat—On—Mat

Something is under a mat: —Under—Mat

Some cat is not on a mat: Cat—On—Mat

Some cat is on something that is not a mat: Cat—On—Mat

If a cat is on a mat, then it is a happy pet: 

The Core CLIP Notation

Existence: (∃ x) or (Exists x)

Negation: ~[ ] but ~[~[ ]] may be written [If [Then]]

Relations: (Cat x), (Mat x), (Pet x), (Happy x), (On x y), (Under x y)

A cat is on a mat: (∃ x y) (Cat x) (On x y) (Mat y).

Something is under a mat: (∃ x y) (Under x y) (Mat y).

Some cat is not on a mat: (∃ x) (Cat x) ~[(∃ y) (On x y) (Mat y)].

Some cat is on something that is not a mat:
(∃ x y) (Cat x) (On x y) ~[(Mat y)].

If a cat is on a mat, then it is a happy pet:
[If (∃ x y) (Cat x) (On x y) (Mat y)
[Then (Pet x) (Happy x)]].

EGs Without Negation

There is a man. *There is a king.* *There is a man who is a king.*

These examples represent existence and relations:

- **A** *line of identity* **states that something exists. In CLIP, that is (∃x).**
- **Relations in CLIP are represented as (king x), (on x y), (under x y).**

Translating the above EGs to CLIP:

(∃x) (man x). *There is something, which is a man*.

(∃x) (king x). *There is something, which is a king*.

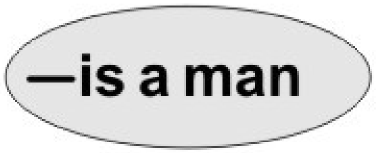
(∃x) (man x) (king x). *There is something, which is a man and a king*.

An option that uses a relation name to *restrict* **the quantifier:**

(∃ x:man) (king x). *Some man is a king*.

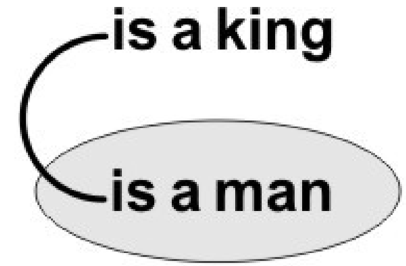
Peirce drew the six EGs in this slide and the next in the manuscript R145, page 21.

EGs With Negation



There is no man.

Something is not a man.



There is a king who is not a man.

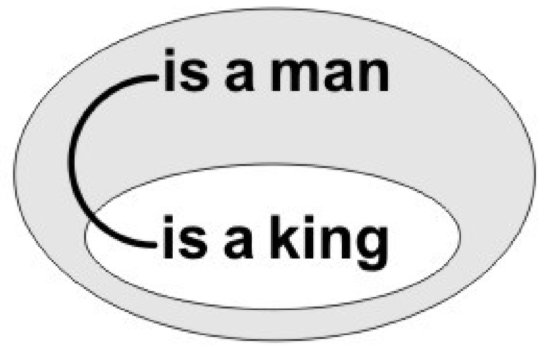
A shaded oval states that the nested graph or subgraph is false:

- **When a line of identity is extended into an oval enclosure, existence is declared in the area that contains the outermost point of the line.**
- **In CLIP, a shaded oval is represented by a tilde ~ and square brackets [ ].**

Translations to CLIP (with two options for the EG on the right):

~[(∃x) (man x)].	*It's false that there is a man*.
(∃x) ~(man x).	*There is something which is not a man*.
(∃x) (king x) ~(man x).	*There is something which is a king and not a man*.
(∃x x:king) ~(man x).	*Some king is not a man*.

Nested Ovals



There is no man who is not a king.

If there is a man, then he is a king.

Every man is a king.

An oval nested in an even number of negations is unshaded.

- **Since a double negation is positive, evenly nested areas are positive.**
- **A nest of two ovals represents an if-then statement.**

With two or more negations, an EG may be translated to CLIP or to English in several equivalent ways:

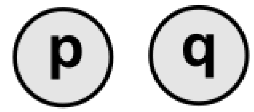
~[(∃x) (man x) ~(king x)].	***It's false that there is a man who is not a king.***
[If (∃ x:man) [Then (king x)]].	***If there is a man, then he is a king*.**
(∀ x:man) (king x).	***Every man x is a king*.**

Boolean Combinations

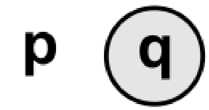
Areas nested inside an odd number of negations are shaded.

p q

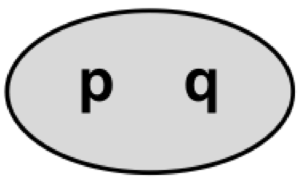
p and q



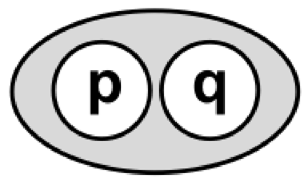
not p and not q



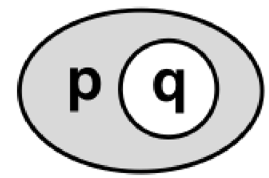
p and not q



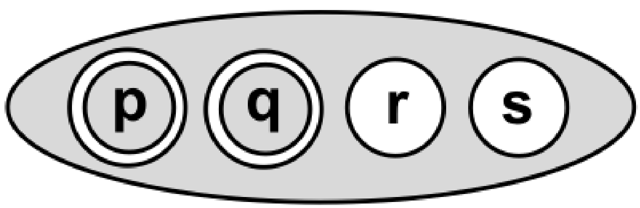
not (p and q)



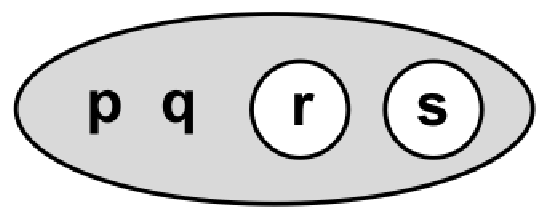
p or q



if p, then q



not p or not q or r or s

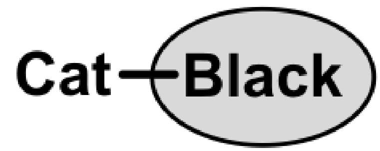


if p and q, then r or s

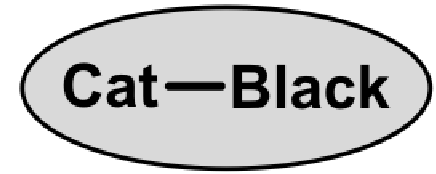
The Scope of Quantifiers

The scope is determined by the outermost point of any line.

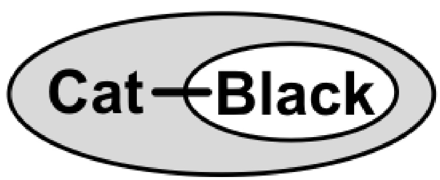
Some cat is black.



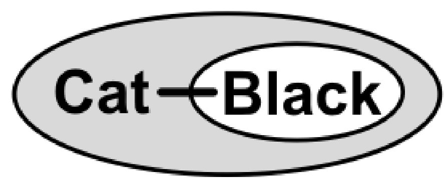
Some cat is not black.



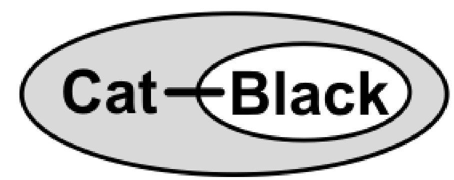
No cat is black.



It is false that some cat is not black.



If there is a cat, then it is black.



Every cat is black.

EGs in these four patterns represent the four sentence types in Aristotle's syllogisms: **http://www.jfsowa.com/talks/aristo.pdf**

Epistemology

Before deduction is possible, the premises must be derived by observation or some method of reasoning:

- **All observations, including the raw data of every science, can be stated with just two operators: existence $\exists$ and conjunction $\wedge$.**
- **The universal quantifier $\forall$ is derived by induction from observations and the assumption that those observations exhaust all possible cases.**
- **Implication $\supset$ cannot be observed. "Post hoc, ergo propter hoc" is a classical fallacy.**
- **For disjunction $\vee$, only one option at a time can be observed. The possibility of alternatives must be inferred by some method..**
- **Even negation ~ must be inferred. Absence of observation is not a proof of absence.**

Conclusion: Existence, conjunction, and negation are closer to observation than the other logical operators:

- **Existence and conjunction are the only ones that can be observed.**
- **If a proposition p is expected, a negation of p may be inferred from a failure to observe p.**

Reasoning

Are the primitives of reasoning and epistemology the same?

- For syllogisms, Aristotle used only ∀, ∃, and ~ in statements, but he assumed a conjunction ∧ of the premises and ⊃ for the conclusion.
- The Stoics introduced hypothetical syllogisms with ⊃ and disjunctive syllogisms with ∨.
- Ockham specified a model-theoretic semantics for a subset of Latin with all these operators, but he didn't make any claims about primitives.

Both Peirce and Frege assumed that a sign for ⊃ was necssary:

- For his *Begriffsschrift*, Frege's only operators were ∀, ⊃, and ~.
- For the *Principia*, Whitehead and Russell adopted Frege's primitives, axioms, and rules of inference. But their proof procedure was the worst ever inflicted upon innocent students.
- In 1934, Gentzen developed much improved proof procedures, but his derivations were distorted by the limitations of the *Principia* notation.

In 1911, Peirce adopted the primitives of epistemology for EGs.

- His 1911 proof procedure is a simplification and generalization of Gentzen's methods of 1934.

Syntax of Existential Graphs

Example: Cat — On — Mat

- **Two lines mean** ***There exist something x and something y*****.**
- **Cat and Mat are** ***monadic*** **relations. On is a** ***dyadic*** **relation.**

Five syntactic features:

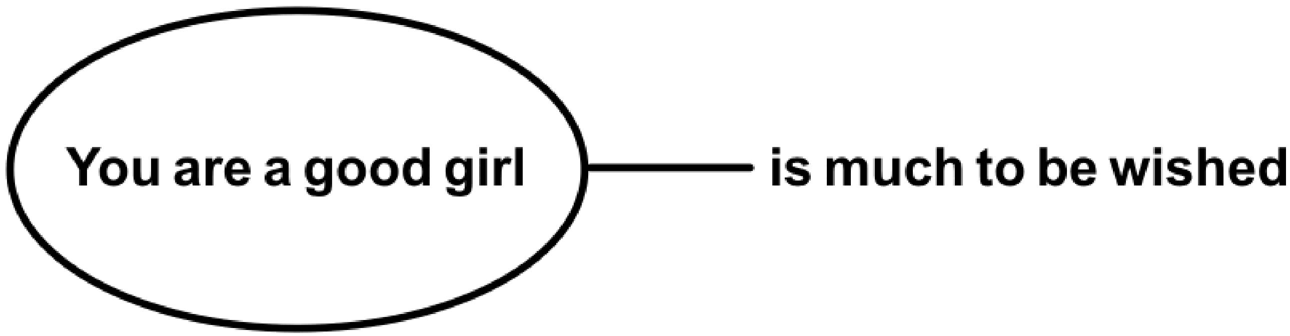
- **Relation: A name with zero or more** ***pegs*** **for attaching lines.**
- **Existence: A** ***line of identity*** **that says** ***Something exists*****.**
- **Conjunction: Two or more graphs in the same area.**
- **Metalanguage: An** ***oval*** **that covers some area.**
- **Negation: A** ***shaded oval*** **that represents the operator** ***not*****.**

Five combinations:

- **Proposition: A graph of lines attached to the pegs of relations.**
- **Identity: Two or more connected lines (called a** ***ligature*****).**
- **Denial: A shaded oval that denies the EG it covers.**
- **Complex Boolean operators: Nests of two or more negations.**
- **Metalanguage: A line that connects an oval to a relation.**

Metalanguage

A relation attached to an oval makes a metalevel comment about the proposition expressed by the nested graph. *



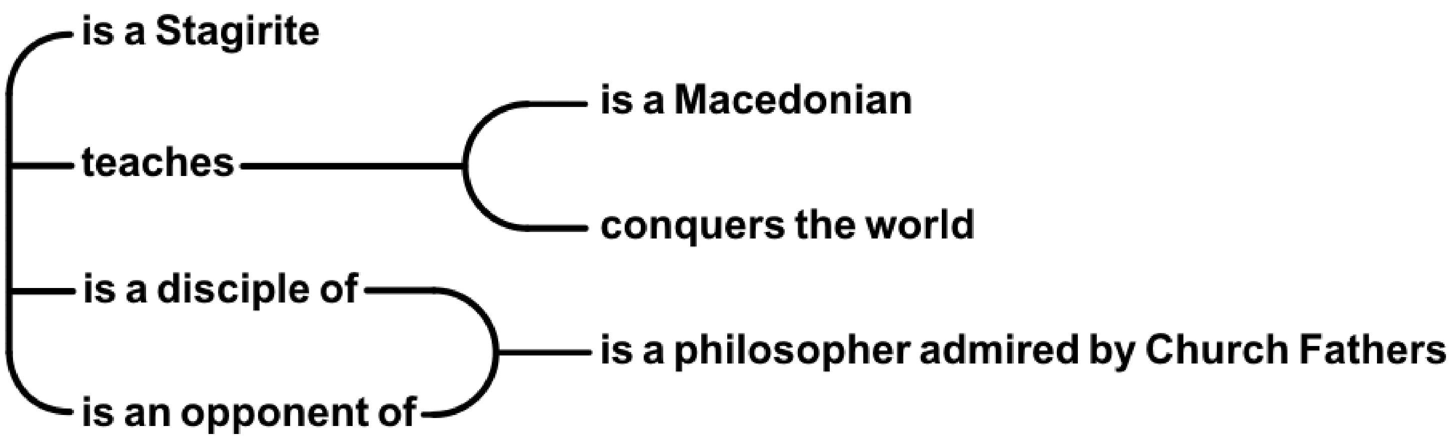
Peirce allowed the names of relations to contain blanks.

The relation named 'You are a good girl' has zero pegs.
It is an EG that expresses a proposition p.

The relation named 'is much to be wished' has one peg, which is attached to a line, which says that the proposition p exists.

*** From Charles Sanders Peirce, *Reasoning and the Logic of Things*, The Cambridge Conferences Lectures of 1898, Harvard University Press, p. 151.**

One of Peirce’s Examples



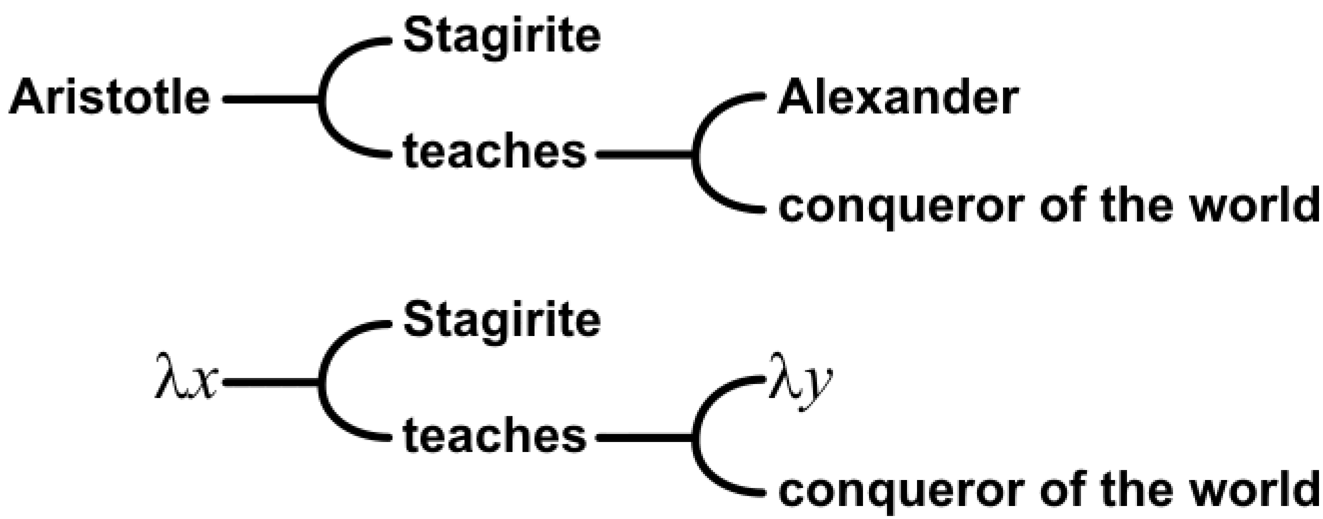
Peirce’s translation to English: ***“There is a Stagirite who teaches a Macedonian conqueror of the world and who is at once a disciple and an opponent of a philosopher admired by Fathers of the Church.”***

A translation to CLIP:

> **(∃ x y z) ("is a Stagirite" x) (teaches x y) ("is a Macedonian" y) ("conquers the world" y) ("is a disciple of" x z) ("is an opponent of" x z) (“is a philosopher admired by church fathers” z).**

Without negation, CLIP can represent the content of a relational database or the graph databases of the Semantic Web.

Lambda Abstraction



The top EG says ***Aristotle is a Stagirite who teaches Alexander who conquers the world*****.**

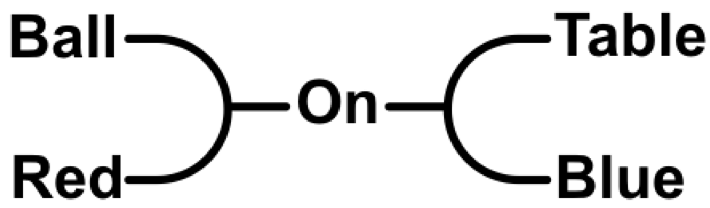
In the EG below it, the names Aristotle and Alexander are erased, and their places are marked with the Greek letter λ.

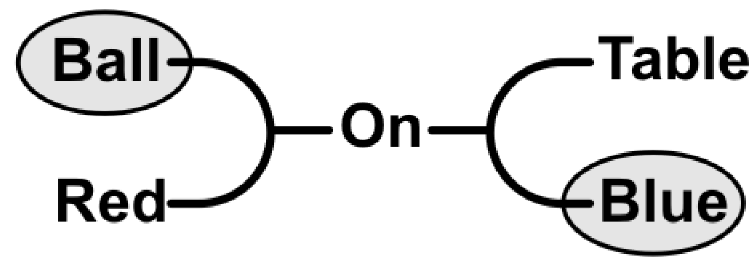
That EG represents a dyadic relation: ___ ***is a Stagirite who teaches*** **___** ***who conquers the world*****.**

Peirce used an underscore to mark those empty places, but Alonzo Church marked them with λ.

Translating EGs to and from English

Most existential graphs can be read in several equivalent ways.





Left graph:

A red ball is on a blue table.

Some ball that is red is on some table that is blue.

Right graph:

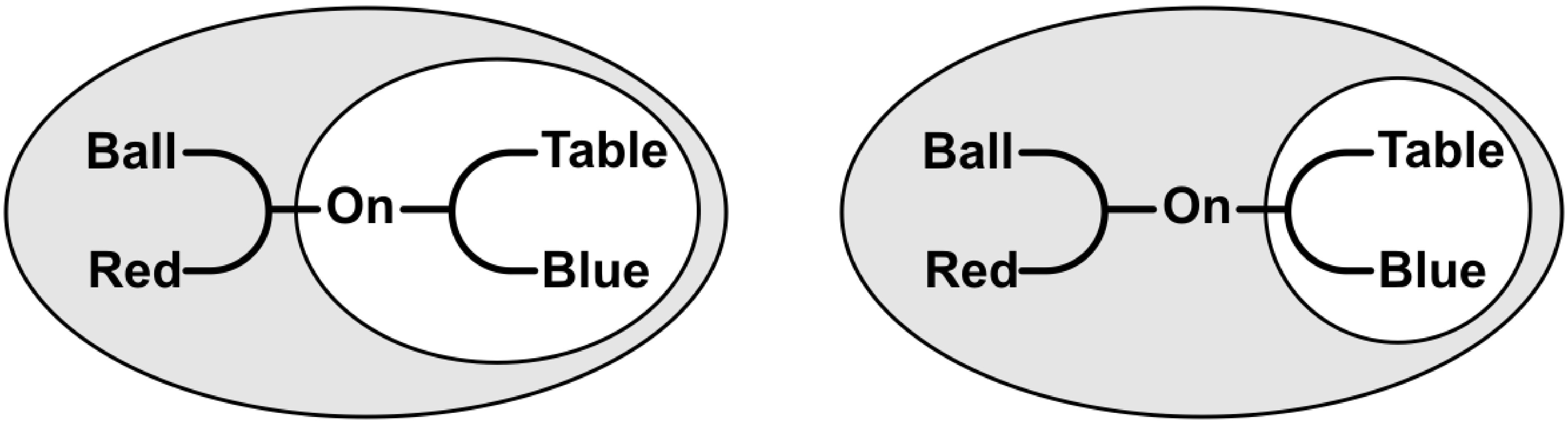
Something red that is not a ball is on a table that is not blue.

A red non-ball is on a non-blue table.

On some non-blue table, there is something red that is not a ball.

Scope of Quantifiers and Negations

Ovals define the scope for both quantifiers and negations.



Left graph:

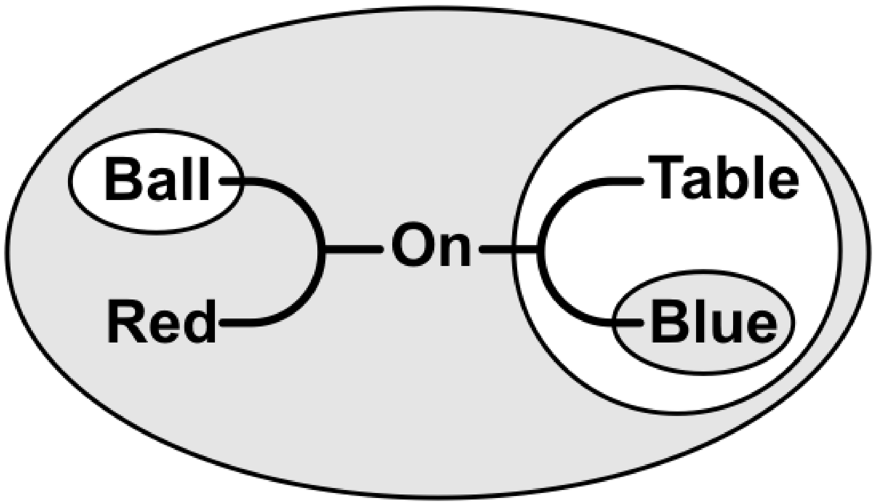
If there is a red ball, then it is on a blue table.

Every red ball is on some blue table.

Right graph:

If a red ball is on something x, then x is a blue table.

EGs With Multiple Nested Negations



The many ways of reading an EG are logically equivalent:

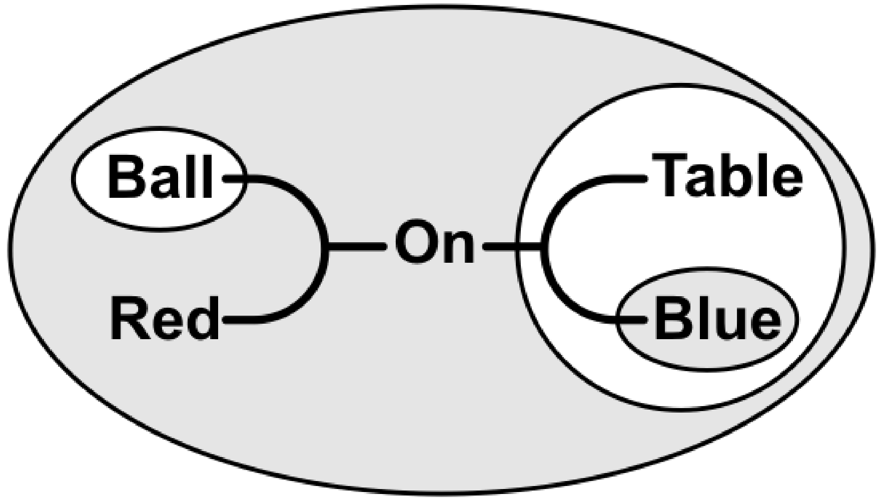
If something red that is not a ball is on something y, then y is a table that is not blue.

If a red thing x is on something y, then either x is a ball, or y is a table that is not blue.

If a red thing x is on something that is not a non-blue table, then x is ball.

Therefore, EGs are a good canonical form for expressing the common meanng. See http://www.jfsowa.com/logic/proposit.pdf

Core CLIP and Extended CLIP



If something red that is not a ball is on something y,
then y is a table that is not blue.

Core CLIP has a one-to-one mapping to and from every EG:

~[(∃x) (∃ y)   (Red x) ~[(Ball x)] (On x y) ~[(Table y) ~[(Blue]]].

A literal reading of Core CLIP is as hard to understand as CLIP itself:
It's false that there exists an x and a y such that x is red and x is not
a ball and x is on y and it's false that y is a table that is not blue.

Extended CLIP simplifies some features and adds readable key words:

[If (∃ x:Red y) ~(Ball x) (On x y) [Then (Table y) ~(Blue y)]].

Coreference Nodes in CLIP

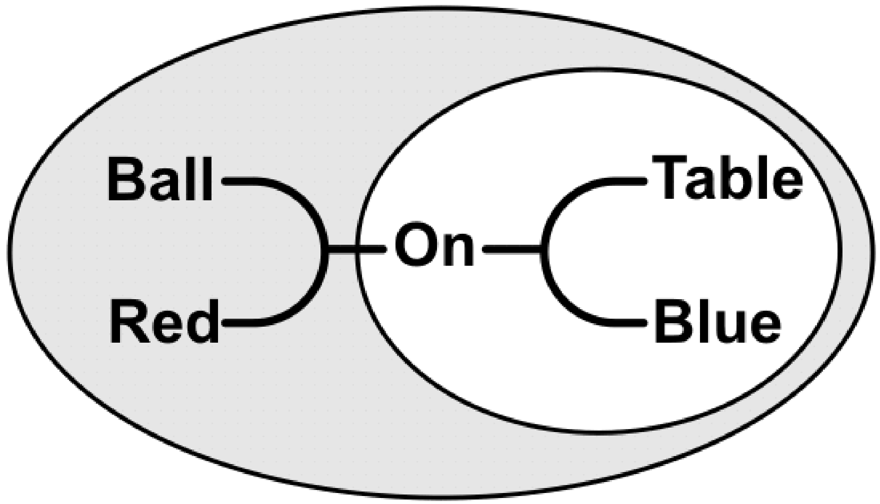
Coreference nodes show how lines are extended into a nested area and how they are connected to form ligatures:

- Quantifier node: (∃x) represents the outermost point of the line x.
- Identity (= x y) is a coreference node of lines x and y. It shows that both names refer to the same entity.
- Teridentity (= x y z) is a coreference node for a ligature of three lines. It is equivalent to pair of identities, (= x y) and (= y z).
- Coreference nodes may connect any number of lines.
- A coreference node with just one line (= x) shows an extension of a line x. It is often used as an intermediate step in a proof.

With coreference nodes, CLIP can show how an EG may be derived by connecting the lines of simpler EGs.

If two or more names in the same area are coreferent, the name of the line whose quantifier occurs before (to the left of) all the others may replace the names of the others.

Representing Ligatures in CLIP



This EG has two ligatures, each with three branching lines.

Representing each branch with a distinct name in CLIP:

[If (∃ x y z) (Ball x) (Red y) (= x y z)
[Then (∃ u v w) (On z u) (Table v) (Blue w) (= u v w)]]

After replacing coreferent names:

[If (∃ x) (Ball x) (Red x) (= x x x)
[Then (∃ u) (On x u) (Table u) (Blue u) (= u u u)]]

Then delete the irrelevant (= x x x) and (= u u u).

Translating the Word *is* to Logic

Three different translations in English or CLIP:

- **Existence:** *There is x.* ↔ **(∃x).**
- **Predication:** *x is a cat.* ↔ **(Cat x).**
- **Identity:** *x is y.* ↔ **(= x y).**

Do these three translations imply that English is ambiguous?

Or is the syntax of linear notations too complex?

In EGs, all three uses of the word *is* map to a line of identity:

- **Existence:** *There is x.* ↔
- **Predication:** *x is a cat.* ↔ **Cat**
- **Identity:** *x is y.* ↔ **(a ligature of two lines)**

As Peirce said, EGs are more iconic than predicate calculus: they show relationships more clearly and directly.

Issues of Mapping Language to Logic

Hans Kamp observed that the features of predicate calculus PC) do not have a direct mapping to and from natural languages. *

Pronouns can cross sentence boundaries, but variables cannot.

- **Example:** ***Pedro is a farmer. He owns a donkey.***
- **PC: (∃x)(Pedro(x) ∧ farmer(x)). (∃y)(∃z)(owns(y,z) ∧ donkey(z)).**
- **There is no operator that can relate x and y in different formulas.**

In English, quantifiers in the if-clause govern the then-clause.

- **Example:** ***If a farmer owns a donkey, then he beats it.***
- **But in predicate calculus, the quantifiers must be moved to the front.**
- **CLIP supports both options: English-like and PC-like.**
  If (∃ x y) (farmer x) (donkey y) (owns x y) [Then (beats x y)]].
  (∀ x y) If (farmer x) (donkey y) (owns x y) [Then (beats x y)]].

Note: Proper names are rarely unique identifiers. Both Kamp and Peirce represented names by monadic predicates.

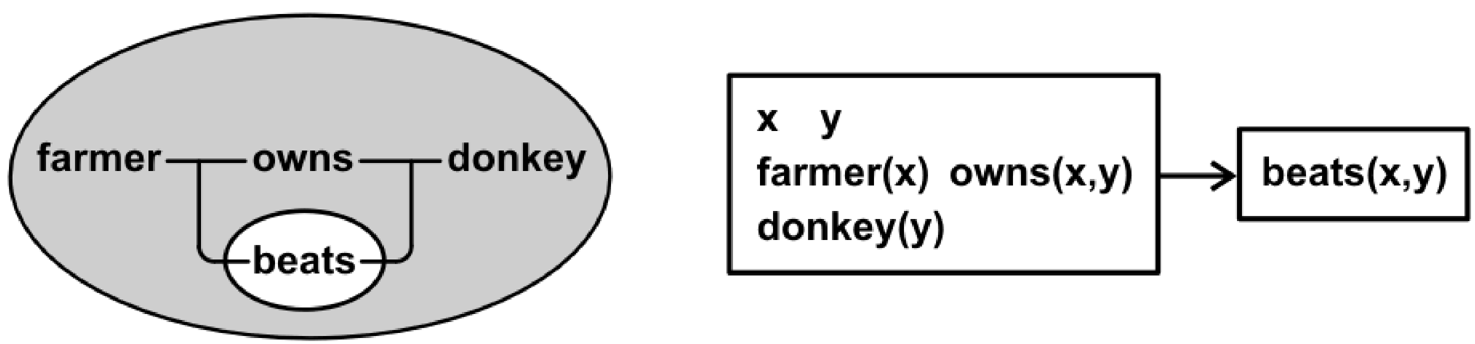
*** Hans Kamp & Uwe Reyle (1993) *From Discourse to Logic*, Dordrecht: Kluwer.**

Quantifiers in EG and DRS

Peirce and Kamp independently chose isomorphic structures.

- **Peirce chose nested ovals for EG with lines to show coreference.**
- **Kamp chose boxes for DRS with variables to show coreference.**
- **But the boxes and ovals are isomorphic: they have the same constraints on the scope of quantifiers, and they support equivalent operations.**

Example: ***If a farmer owns a donkey, then he beats it.***



In these examples, the same CLIP represents the EG and the DRS:
[If (∃ x y) (farmer x) (owns x y) (donkey y) [Then (beats x y)]].

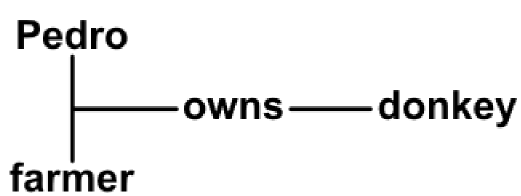
Combining EG Graphs or DRS Boxes

Two English sentences, ***Pedro is a farmer. He owns a donkey,*** **are represented by EG graphs (left) and DRS boxes (right):**

x
Pedro(x) farmer(x)

y z
owns(y,z) donkey(z)

Combine them by connecting EG lines or merging DRS boxes:



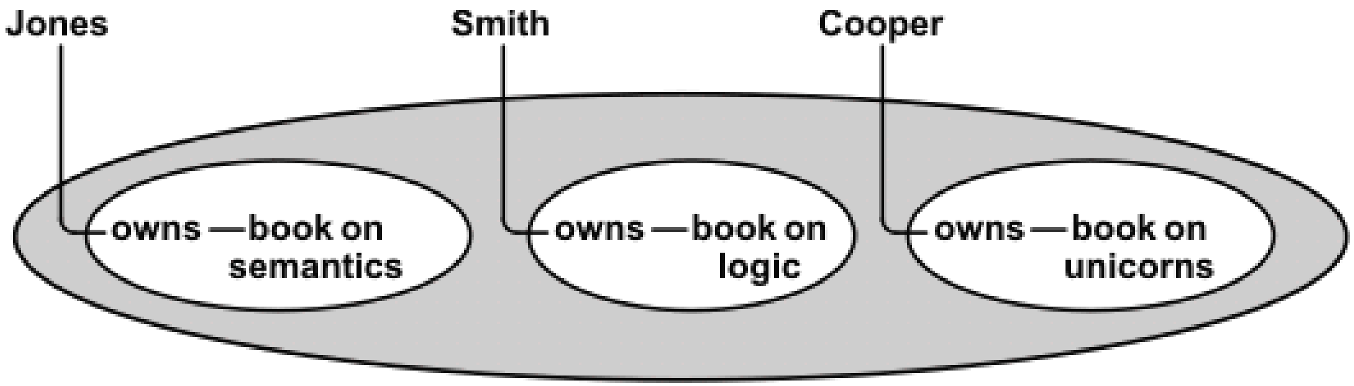
x y z x=y
Pedro(x) farmer(x)
owns(y,z) donkey(z)

Equivalent operations on EG and DRS produce the same CLIP:
(∃ x y z) (Pedro x) (farmer x) (= x y) (owns y z) (donkey z).

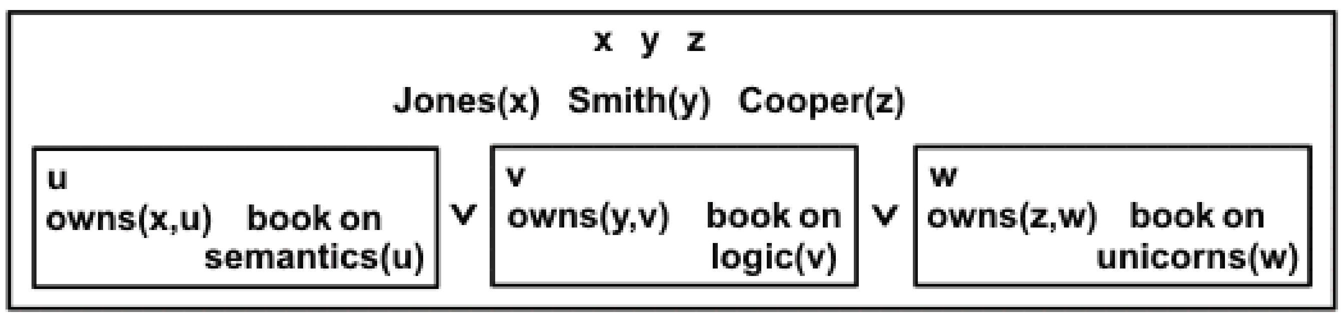
Disjunction in EG, DRS, and CLIP

Kamp and Reyle (1993): ***"Either Jones owns a book on semantics, or Smith owns a book on logic, or Cooper owns a book on unicorns."***

EG:



DRS:



CLIP:

(∃ x y z) (Jones x) (Smith y) (Cooper z)
[Or [(∃ u) (owns x u) ("book on semantics" u)]
[(∃ v) (owns y v) ("book on logic" v)]
[(∃ w) (owns z w) ("book on unicorns" w)]].

Peirce’s Rules of Inference

Peirce’s rules support the simplest, most general reasoning method ever invented for any logic.

Three pairs of rules, which insert or erase a graph or subgraph:

1. **Insert/Erase: Insert anything in a negative area; erase anything in a positive area.**
2. **Iterate/Deiterate: Iterate (copy) anything in the same area or any nested area; deiterate (erase) any iterated copy.**
3. **Double negation: Insert or erase a double negation (pair of ovals with nothing between them) around anything in any area.**

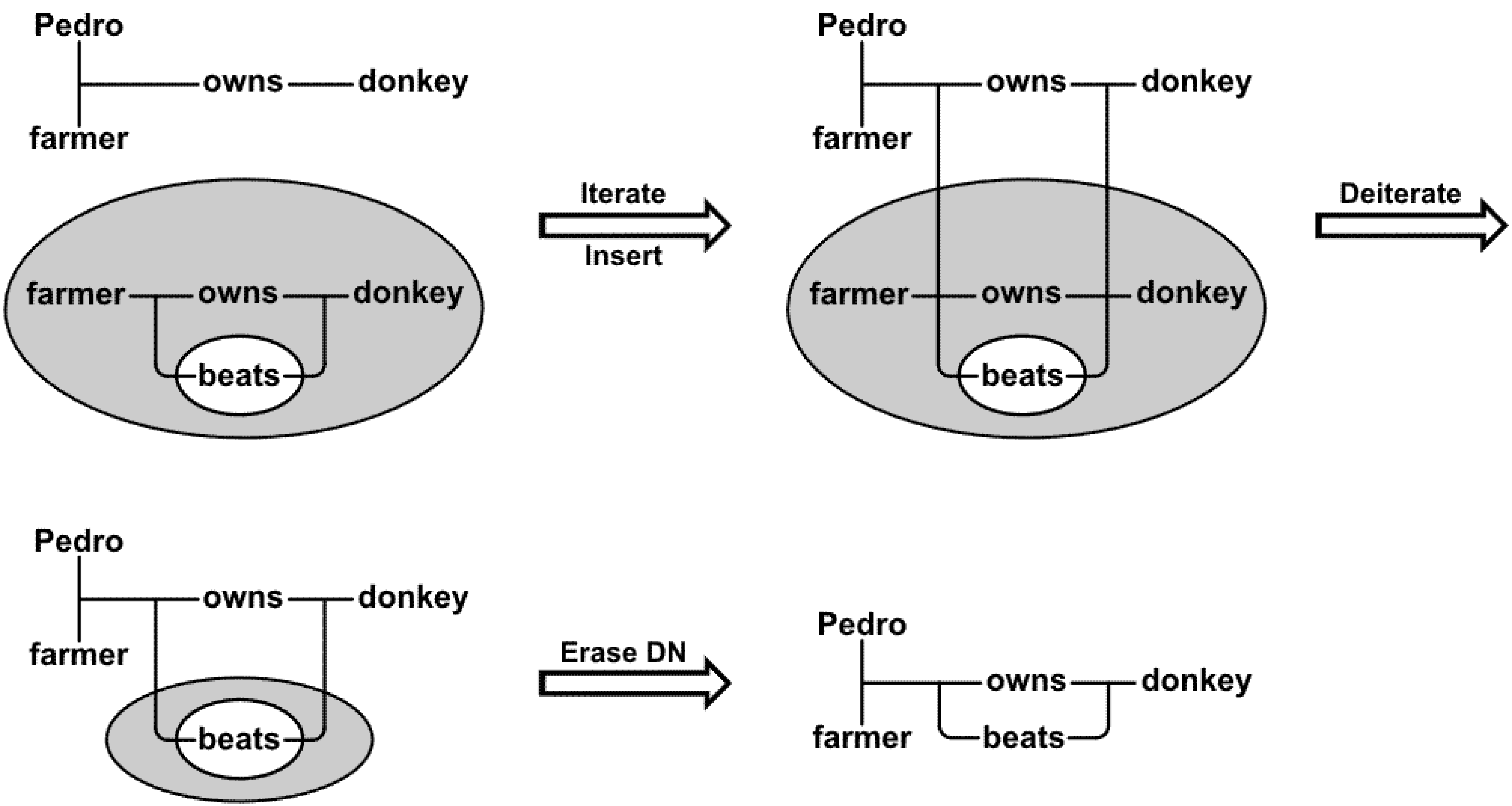
These rules are stated in terms of EGs.

But they can be adapted to many notations, including CLIP, DRS, predicate calculus, various diagrams, and natural languages.

For details, see Reasoning with diagrams and images,

http://www.collegepublications.co.uk/downloads/ifcolog00025.pdf .

A Proof by Peirce’s Rules



Conclusion: ***Pedro is a farmer who owns and beats a donkey*.**

Proving a Theorem

Peirce's only axiom is the empty graph – a blank sheet of paper.

- **The empty graph cannot say anything false.**
- **Therefore, the empty graph is always true.**
- **Silence is golden.**

A theorem is a proposition that is proved from the empty graph.

- **For the first step, only one rule can be applied: draw a double negation around a blank area.**
- **The next step is to insert the hypothesis into the negative area.**

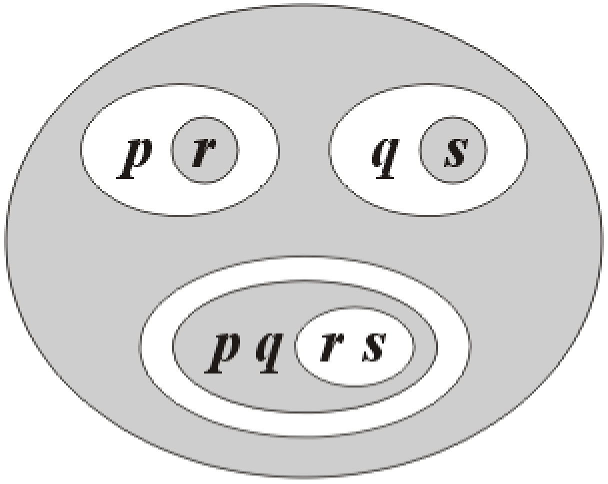
The Praeclarum Theorema (splendid theorem) by Leibniz:

PC: $((p \supset r) \wedge (q \supset s)) \supset ((p \wedge q) \supset (r \wedge s))$.

In the *Principia Mathematica*, Whitehead and Russell took 43 steps to prove this theorem.

With Peirce's rules, the proof takes only 7 steps.

Praeclarum Theorema

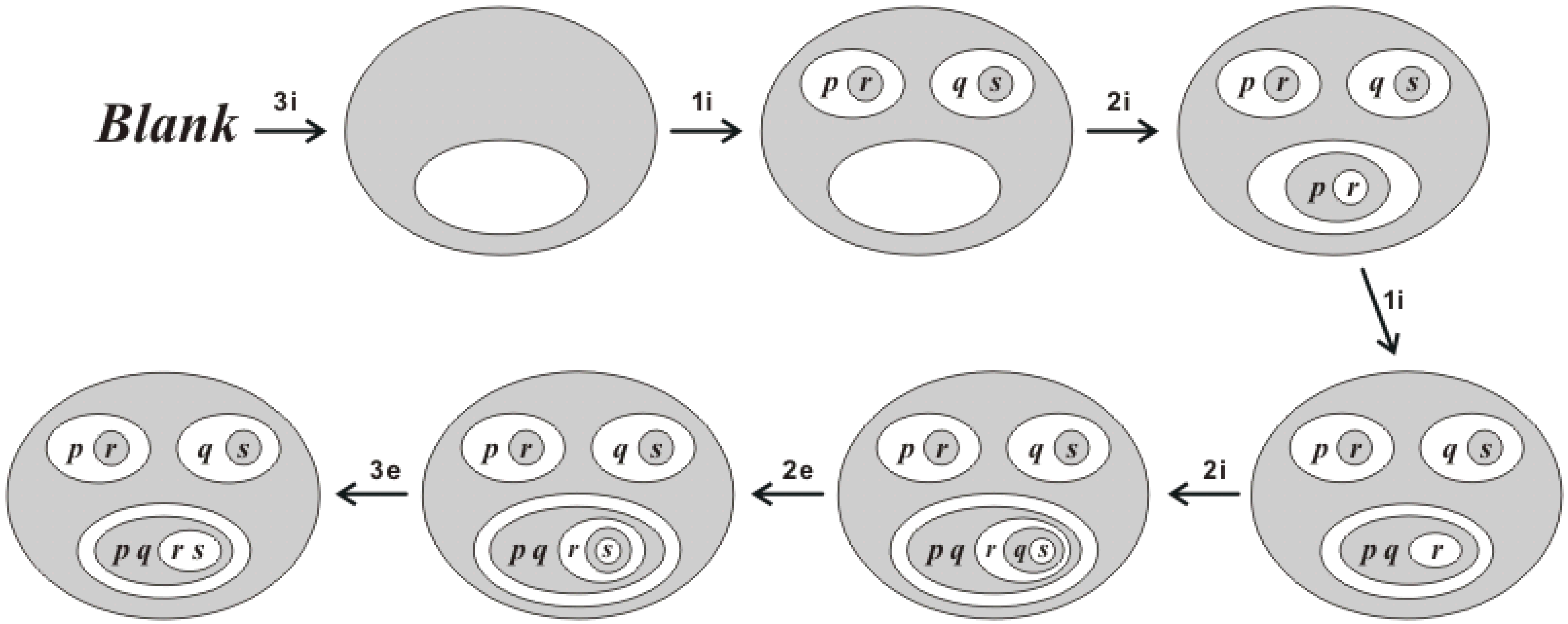


PC: $((p \supset r) \wedge (q \supset s)) \supset ((p \wedge q) \supset (r \wedge s))$

Note that the if-parts of $(p \supset r)$ and $(q \supset s)$ are white, because those areas are nested two levels deep.

But the if-part of $(p \wedge q) \supset (r \wedge s)$ is shaded, because that area is nested three levels deep.

Proof of the Praeclarum Theorema



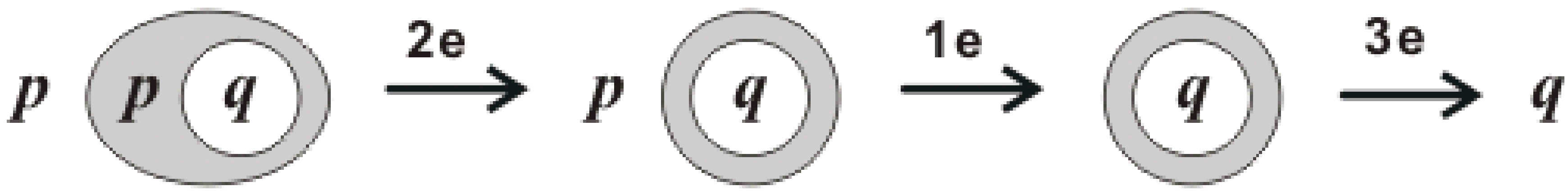
Each step is labeled with the number of the rule:

3i, insert double negation. 1i, insert (($p \supset r$) $\wedge$ ($q \supset s$)). 2i, iterate ($p \supset r$).
1i, insert *q*. 2i, iterate ($q \supset s$). 2e, deiterate q. 3e, erase double negation.

For humans, perception determines which rule to apply.

Look ahead to the conclusion to see which rule would make the current graph look more like the target graph.

Derived Rules of Inference



Proof of modus ponens: Given (p) and [If (p) [Then (q)]]:

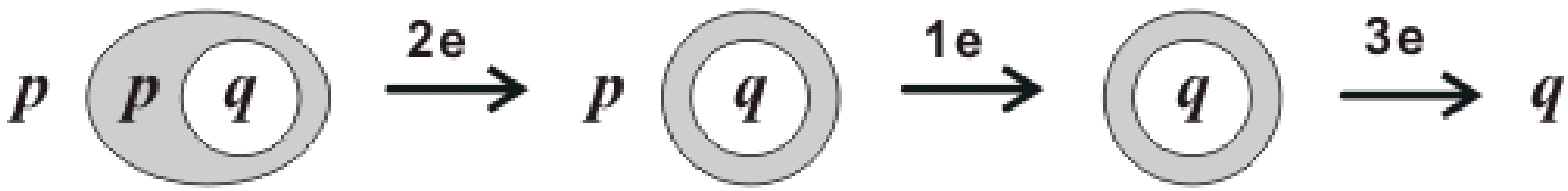
2e, deiterate nested p. 1e, erase p. 3e, erase double negation.

Therefore, modus ponens may be used as a derived rule of inference in any proof by Peirce's rules.

In general,

- **All rules and proof procedures of classical first-order logic may be derived by a proof that uses Peirce's rules.**
- **Therefore, any or all of those rules may be used as derived rules in any proof that uses EGs.**
- **With appropriate constraints, Peirce's rules may also be adapted to higher-order logics, nonmonotonic logics, intuitionistic logics, etc.**

Proof of Modus Ponens in CLIP



Proof of modus ponens in CLIP:

0. Given: (p) and ~[(p) ~[(q)]].
1. By 2e, deiterate (erase) the nested (p): (p) ~[~[(q)]].
2. By 1e, erase (p) in a positive area: ~[~[(q)]].
3. By 3e, erase the double negation: (q).

Observations:

- CLIP can represent the full semantics of ISO Common Logic (CL).
- CL is a superset of a wide range of logics used in computer systems.
- Therefore, EGs and CLIP proofs can be used to represent a wide range of logics and proofs in computer science and systems.

Proof of the Praeclarum Theorema in CLIP

1. By 3i, draw a double negation around the blank: ~[~[ ]].

2. By 1i, insert the hypothesis in the negative area:
   ~[~[(p) ~[(r)]] ~[(q) ~[(s)]] ~[ ]].

3. By 2i, iterate the left part of the hypothesis into the conclusion:
   ~[~[(p) ~[(r)]] ~[(q) ~[(s)]] ~[~[(p) ~[(r)]]]].

4. By 1i, insert (q):
   ~[~[(p) ~[(r)]] ~[(q) ~[(s)]] ~[~[(p) (q) ~[(r)]]]].

5. By 2i, iterate the right part of the hypothesis into the innermost area:
   ~[~[(p) ~[(r)]] ~[(q) ~[(s)]] ~[~[(p) (q) ~[(r) ~[(q) ~[(s)]]]]].

6. By 2e, deiterate (q):
   ~[~[(p) ~[(r)]] ~[(q) ~[(s)]] ~[~[(p) (q) ~[(r) ~[~[(s)]]]]].

7. By 3e, erase the double negation to generate the conclusion:
   ~[~[(p) ~[(r)]] ~[(q) ~[(s)]] ~[~[(p) (q) ~[(r) (s)]]].

8. Replace the negations by the keywords 'If' and 'Then':
   [If [If (p) [Then (r)]] [If (q) [Then (s)]] [Then [If (p) (q) [Then (r) (s)]]].

Applying Peirce's Rules to Other Notations

With minor changes, Peirce's rules can be used with many logic notations, including controlled subsets of natural languages.

Definition: Proposition X is more general (or specialized) than Y iff the models for X are a proper superset (subset) of the models for Y.

Modified version of Peirce's first pair of rules:

- Insert: In a negative context, any propositional expression may be replaced by a more specialized expression.
- Erase: In a positive context, any propositional expression may be replaced by a more general expression.

The rules of Iterate/Deiterate and Double Negation are unchanged.

This modification holds for existential graphs, since erasing any subgraph makes a graph more general.

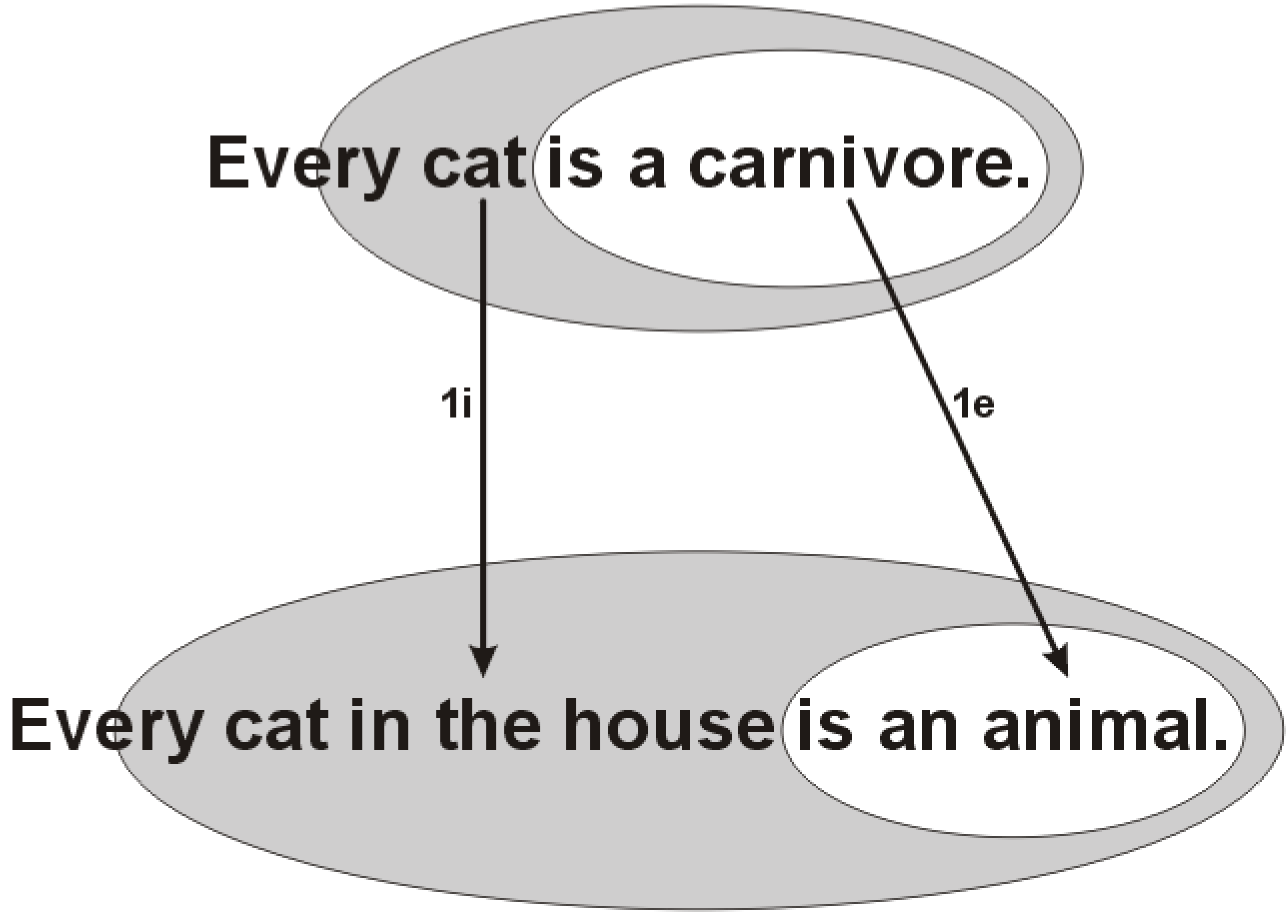
But this version can be easier to apply to other notations.

Peirce’s Rules Applied to English

Mark the positive and negative parts of each sentence by overlaying shaded ovals.

Rule 1i specializes 'cat' to 'cat in the house'.

Rule 1e generalizes 'carnivore' to 'animal'.



This method of reasoning is sound for sentences that can be mapped to a formal logic. It can also be used on propositional parts of sentences that contain some nonlogical features.

A Proof in English

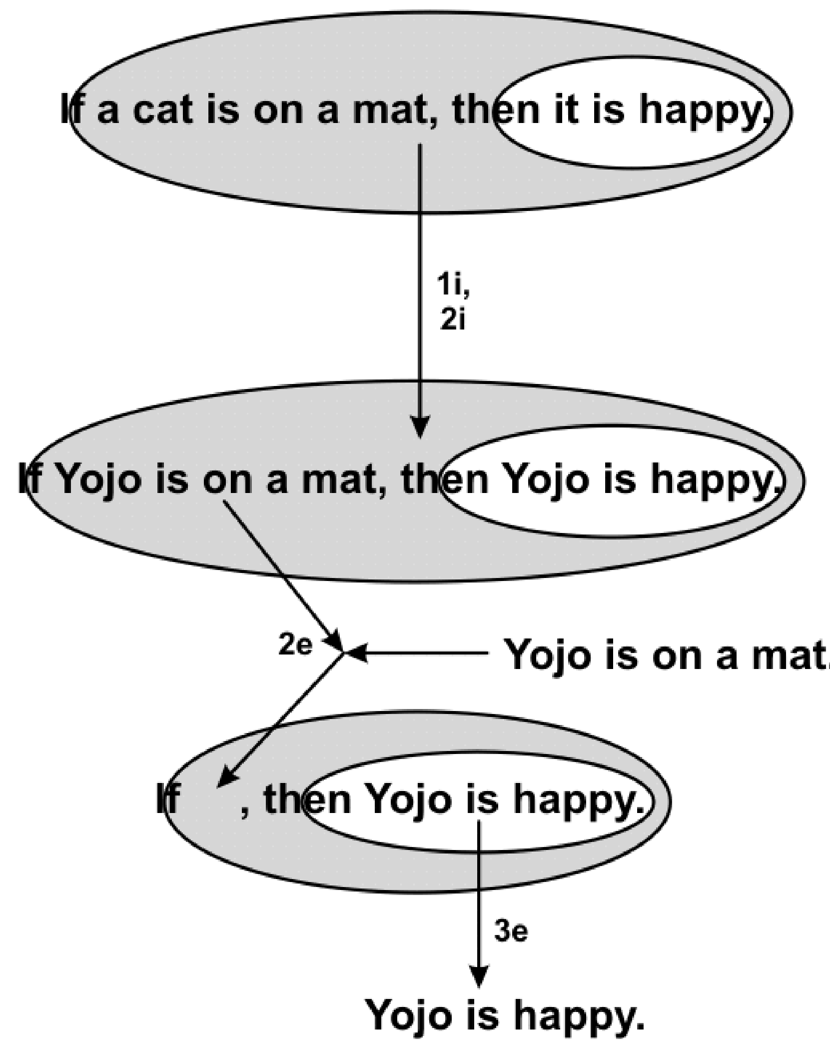
Use shading to mark positive and negative parts of each sentence.

Rule 1i specializes 'a cat' to 'Yojo', and Rule 2i iterates 'Yojo' to replace the pronoun 'it'.

Rule 2e deiterates the nested copy of the sentence 'Yojo is on a mat'.

As a result, there is nothing left between the inner and outer negation of the if-then nest.

Finally, Rule 3e erases the double negation to derive the conclusion.



Natural Deduction

Gerhard Gentzen developed a proof procedure that he called *natural deduction*.

But Peirce's method is a version of natural deduction that is simpler and more general than Gentzen's:

Peirce's Method	Gentzen's Method
6 rules	16 rules
3 symmetric pairs	Many irregularities
Simple operations	Requires provability
Straight-line proofs	Complex bookkeeping
Date: 1897-1909	Date: 1935

For a proof of equivalence, see http://www.jfsowa.com/pubs/egtut.pdf

Gentzen’s Natural Deduction

	Introduction Rules	Elimination Rules
$\wedge$	$\frac{A,\ B}{A \wedge B}$	$\frac{A \wedge B}{A}$ $\quad$ $\frac{A \wedge B}{B}$
$\vee$	$\frac{A}{A \vee B}$ $\quad$ $\frac{B}{A \vee B}$	$\frac{A \vee B,\ \ A \vdash C,\ \ B \vdash C}{C}$
$\supset$	$\frac{A \vdash B}{A \supset B}$	$\frac{A,\ \ A \supset B}{B}$
$\sim$	$\frac{A \vdash \bot}{\sim A}$ $\quad$ $\frac{\bot}{A}$	$\frac{A,\ \ \sim A}{\bot}$ $\quad$ $\frac{\sim\sim A}{A}$
$\forall$	$\frac{A(a)}{(\forall x)A(x)}$	$\frac{(\forall x)A(x)}{A(t)}$
$\exists$	$\frac{A(t)}{(\exists x)A(x)}$	$\frac{(\exists x)A(x),\ \ A(a) \vdash B}{B}$

Like Peirce, Gentzen assumed only one axiom: a blank sheet of paper.

But Gentzen had more operators and more complex, nonsymmetric pairs of rules for inserting or erasing operators.

Role of the Empty Sheet

Both Peirce and Gentzen start a proof from an empty sheet.

In Gentzen's syntax, a blank sheet is not a well-formed formula.

- Therefore, no rule of inference can be applied to a blank.
- The method of making and discharging an assumption is the only way to begin a proof.

But in EG syntax, an empty graph is a well-formed formula.

- Therefore, a blank may be enclosed in a double negation.
- Then any assumption may be inserted in the negative area.

Applying Peirce's rules to predicate calculus:

- Define a blank as a well-formed formula that is true by definition.
- Define the positive and negative areas for each Boolean operator.
- Show that each of Gentzen's rules is a derived rule of inference in terms of Peirce's rules.

Then any proof by Gentzen's rules is a proof by Peirce's rules.

Theoretical Issues

Peirce's rules have some remarkable properties:

- **Simplicity: Each rule inserts or erases a graph or subgraph.**
- **Symmetry: Each rule has an exact inverse.**
- **Depth independence: Rules depend on the positive or negative areas, not on the depth of nesting.**

They allow short proofs of remarkable theorems:

- **Reversibility Theorem. Any proof from *p* to *q* can be converted to a proof of ~*p* from ~*q* by negating each step and reversing the order.**
- **Cut-and-Paste Theorem. If *q* can be proved from *p* on a blank sheet, then in any positive area where *p* occurs, *q* may be substituted for *p*.**
- **Resolution and natural deduction: Any proof by resolution can be converted to a proof by Peirce's version of natural deduction by negating each step and reversing the order.**

For proofs of these theorems and further discussion of the issues, see Section 6 of **http://www.jfsowa.com/pubs/egtut.pdf**

A Problem in Automated Reasoning

Larry Wos (1988), a pioneer in automated reasoning methods, stated 33 unsolved problems. His problem 24:

> ***Is there a mapping between clause representation and natural-deduction representation (and corresponding inference rules and strategies) that causes reasoning programs based respectively on the two approaches or paradigms to attack a given assignment in an essentially identical fashion?***

The answer in terms of Peirce's rules is yes:

- **The inference rules for Gentzen's clause form and natural deduction are derived rules of inference in terms of the EG rules.**
- **Any proof in clause form (by resolution) can be converted, step by step, to a proof by EG rules.**
- **Any such proof can be converted to a proof by Peirce's version of natural deduction by negating each step and reversing the order.**
- **Convert the proof by Peirce's rules to a proof by Gentzen's rules.**

Alpha, Beta, and Gamma Graphs

Peirce classified EGs in three categories:

- **Alpha graphs use only conjunction and negation to represent propositional logic.**
- **Beta graphs add the existential quantifier to represent full FOL.**
- **Gamma graphs extend EGs with metalanguage, modal logic, and higher-order logic.**

The semantics of CGIF and the CLIP subset is defined by the ISO standard 24707 for Common Logic (CL).

- **For Alpha and Beta, CL model theory is consistent with Peirce's version, which he called *endoporeutic*.**
- **CL semantics also supports quantification over relations in a way that is compatible with Peirce's version.**
- **But extensions to CLIP are needed for other Gamma features.**

For details, see Section 5 of http://www.jfsowa.com/pubs/eg2cg.pdf

Psychology

Endorsement by the psychologist Philip Johnson-Laird (2002):

> **“Peirce’s existential graphs... establish the feasibility of a diagrammatic system of reasoning equivalent to the first-order predicate calculus.”**
>
> **“They anticipate the theory of mental models in many respects, including their iconic and symbolic components, their eschewal of variables, and their fundamental operations of insertion and deletion.”**
>
> **“Much is known about the psychology of reasoning... But we still lack a comprehensive account of how individuals represent multiply-quantified assertions, and so the graphs may provide a guide to the future development of psychological theory.”**

Johnson-Laird published many papers about mental models.

His comments on that topic are significant, especially in combination with the other properties of the graphs.

Mental Maps, Images, and Models

The neuroscientist Antonio Damasio (2010):

> **"The distinctive feature of brains such as the one we own is their uncanny ability to create maps... But when brains make maps, they are also creating images, the main currency of our minds. Ultimately consciousness allows us to experience maps as images, to manipulate those images, and to apply reasoning to them."**

The maps and images form mental models of the real world or of the imaginary worlds in our hopes, fears, plans, and desires.

Words and phrases of language can be generated from them.

They provide a "model theoretic" semantics for language that uses perception and action for testing models against reality.

Like Tarski's models, they define the criteria for truth, but they are flexible, dynamic, and situated in the daily drama of life.

Reasoning with Mental Models

From Damasio and other neuroscientists:

- Mental models are patterns in the sensory projection areas that resemble patterns generated during perception.
- But the stimuli that generate mental models come from the frontal lobes, not from sensory input.
- The content of the mental models is generated by assembling fragments of earlier perceptions in novel combinations.

From suggestions by Johnson-Laird:

- The nodes of an existential graph could represent images or fragments of images from long-term memory.
- The connecting lines of an EG would show how those fragments are assembled to form a mental model.
- The logical features of EGs could be used to represent rules and constraints for reasoning about those models.

Teaching Logic

EGs are an excellent pedagogical tool for teaching logic at every level from beginners to the most advanced.

For people who were exposed to predicate calculus and hate it:

- **First hour: EG syntax (along the lines of slides 4 to 22).**
- **Second hour: Theorem proving (with more examples than 28 to 35).**
- **Third hour: Draw EGs and ask the class how to prove them.**
- **After 3 hours, they say it's the first time they understood logic.**

For advanced students:

- **Present all slides in one-hour, followed by a half-hour discussion.**

Observation by Don Roberts at the University of Waterloo:

- **Students who start with EGs and move to predicate calculus score higher on exams than students who study only predicate calculus.**
- **The biggest improvement is in their ability to prove theorems.**

Summary

Existential graphs (EGs) are an excellent pedagogical tool.

EG rules of inference are notation-independent:

- **The same rules can be used for propositional logic, first-order logic, and many variations and extensions of FOL.**
- **They can be used with graphic and linear notations, including versions of natural language.**
- **They could even be implemented in neural networks..**

Peirce claimed that EGs represent “a moving picture of the action of the mind in thought.”

The psychologist Philip Johnson-Laird agreed: Peirce’s EGs and rules of inference are a good candidate for a natural logic.

Related Readings

Sowa, John F. (2011) Peirce's tutorial on existential graphs,
http://www.jfsowa.com/pubs/egtut.pdf

Sowa, John F. (2013) From existential graphs to conceptual graphs,
http://www.jfsowa.com/pubs/eg2cg.pdf

Sowa, John F. (2015) Slides for a tutorial on natural logic,
http://www.jfsowa.com/talks/natlog.pdf

Johnson-Laird, Philip N. (2002) Peirce, logic diagrams, and the elementary operations of reasoning, *Thinking and Reasoning* 8:2, 69-95. http://mentalmodels.princeton.edu/papers/2002peirce.pdf

Pietarinen, Ahti-Veikko (2009) Peirce's development of qantification theory,
http://www.helsinki.fi/peirce/PEA/Pietarinen%20%2d%20Peirce%27s%20Development.pdf

Pietarinen, Ahti-Veikko (2003) Peirce's magic lantern of logic: Moving pictures of thought,
http://www.helsinki.fi/science/commens/papers/magiclantern.pdf

Pietarinen, Ahti-Veikko (2011) Moving pictures of thought II, *Semiotica* 186:1-4, 315–331,
http://www.helsinki.fi/~pietarin/publications/Semiotica-Diagrams-Pietarinen.pdf

Sowa, John F. (2010) Role of logic and ontology in language and reasoning,
http://www.jfsowa.com/pubs/rolelog.pdf

Sowa, John F. (2006) Peirce's contributions to the 21st Century,
http://www.jfsowa.com/pubs/csp21st.pdf

ISO/IEC standard 24707 for Common Logic,
http://standards.iso.org/ittf/PubliclyAvailableStandards/c039175_ISO_IEC_24707_2007(E).zip

For other references, see the general bibliography,
http://www.jfsowa.com/bib.htm