Existential Graphs

The simplest notation for logic ever invented

John F. Sowa

10 March 2017
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.
How to say “A cat is on a mat.”

Gottlob Frege (1879):

Charles Sanders Peirce (1885):

Giuseppe Peano (1895):

Charles Sanders Peirce (1897):

Conceptual graph (1976):
A Minimal Notation for FOL

A line for existence. An oval for negation. Conjunction is implicit.

Existence: —

Negation: 🔰

Relations: Cat- -On- -Under- -With- -Mat

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
EGs Without Negation

—is a man                     —is a king

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

The *existential graphs* (EGs) above have three kinds of symbols:

- Short lines, called *lines of identity*, state that something exists. In the algebraic notation, they are represented by $\exists$ followed by a letter.
- Character strings name *relations*.
- A *ligature* connects lines of identity. It corresponds to an equal sign $=$.

The three EGs may be translated to Peirce-Peano algebra:

- $\exists x \text{ Man}(x)$.  *There is an* $x$, *and* $x$ *is a man.*
- $\exists x \text{ King}(x)$.  *There is an* $x$, *and* $x$ *is a king.*
- $\exists x (\text{Man}(x) \land \text{King}(x))$.  *There is an* $x$, *$x$ is a man, and* $x$ *is a king.*

Peirce drew the six EGs in this slide and the next in MS 145, page 21.
EGs With Negation

- 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 the algebraic formulas, negation is represented by the symbol $\sim$.

- Translation of these three EGs to algebraic formulas:
  - $\sim \exists x \text{ Man}(x)$. It is false that there is an $x$ who is a man.
  - $\exists x \sim \text{Man}(x)$. There is an $x$, and $x$ is not a man.
  - $\exists x (\text{King}(x) \land \sim \text{Man}(x))$. There is an $x$, $x$ is a king, and $x$ is not a man.
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.

An EG with two or more ovals may be translated to English or algebraic formulas in several equivalent ways:

- \( \neg \exists x \ (\text{Man}(x) \land \neg \text{King}(x)) \). There is no \( x \) who is a man and not a king.
- \( \forall x \ (\text{Man}(x) \supset \text{King}(x)) \). For every \( x \), if \( x \) is a man, then \( x \) is a king.
- \( \forall x \ (\text{Man}(x) \supset \text{King}(x)) \). Every man \( x \) is a king.
Boolean Combinations

Areas nested inside an odd number of negations are shaded.

\[ p \quad q \]
\[ p \quad q \quad q \]
\[ p \quad q \quad p \]

- \( p \) and \( q \)
- \( \neg p \) and \( \neg q \)
- \( p \) and \( \neg q \)

\[ p \quad q \quad p \quad q \]
\[ p \quad q \quad p \quad q \quad p \]
\[ p \quad q \quad p \quad q \quad p \quad q \]

- \( \neg (p \text{ and } q) \)
- \( p \) or \( q \)
- if \( p \), then \( q \)

\[ p \quad q \quad r \quad s \]
\[ p \quad q \quad r \quad s \]

- \( \neg p \) or \( \neg q \) 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.

EGs in these four patterns represent the four sentence types in Aristotle’s syllogisms: [Link to PDF]

- Cat—Black
  - Some cat is black.

- Cat—Black
  - Some cat is not black.

- Cat—Black
  - No cat is black.

- Cat—Black
  - It is false that some cat is not black.

- Cat—Black
  - If there is a cat, then it is black.

- Cat—Black
  - Every cat is black.
Syntax of Existential Graphs

Example: Cat — On — Mat

- Two lines mean there exists something \( x \) and something \( y \).
- Cat and Mat are monadic relations. On is a dyadic relation.

Four primitives:

- Relation: A character string with zero or more pegs.
- Existence: A line that means “There exists something.”
- Conjunction: Two or more graphs in the same area.
- Metalanguage: An oval that encloses some graph or subgraph.

Four combinations:

- Argument: A line attached to a peg of some relation.
- Equality: Two or more connected lines (called a ligature).
- Metacomment: A line that connects an oval to a relation.
- Negation: A shaded oval that represents the comment not.
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.

Choice of Primitives

Motivation:

- Frege: Choosing the operators negation, implication, and the universal quantifier in order to state his rules of inference.
- Peirce (1885): Combining Boolean operators, relational algebra, and the quantifiers used in syllogisms.
- Peirce (1897): Integrating logic and semiotic.

Reasons for those choices:

- Frege believed that deduction was the essence of logic and that the notation should be based on the operators used in deduction.
- Peirce (1885) combined the Aristotelian tradition with a Boolean (algebraic) style of notation.
- Peirce (1897) integrated EG notation and methods of reasoning with every aspect of his philosophy – especially semiotic.

See “Peirce the Logician” by Hilary Putnam for an overview of the contributions to logic by Peirce and Frege:  http://www.jfsowa.com/peirce/putnam.htm
Epistemological Issues

Before deduction is possible, the premises must be derived by observation, induction, abduction, or analogy:

- All observations, including the raw data of every science, can be stated with just the operators of 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 $\lor$, only one option at a time can be observed. The possibility of alternatives must be inferred.
- Even negation $\neg$ 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.
- With negation, the others can be defined.
The RDF Subset of Logic

English: “There exists an x, which is a Stagirite, x teaches some y, which is a Macedonian that conquers the world, x is a disciple of some z, which is a philosopher admired by Church Fathers, and x is an opponent of z.”

The only logical operators are existence and conjunction:

\[ \exists x \exists y \exists z (\text{isaStagirite}(x) \land \text{teaches}(x,y) \land \text{isaMacedonian}(y) \land \text{conquersTheWorld}(y) \land \text{isaDiscipleOf}(x,z) \land \text{isanOpponentOf}(x,z) \land \text{isaPhilosopherAdmiredByChurchFathers}(z)). \]

This subset of logic can represent the content of any relational DB or any RDF DB (including blank nodes in RDF).
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 $\lambda$.

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 $\lambda$. 
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.
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 meaning. See [http://www.jfsowa.com/logic/proposit.pdf](http://www.jfsowa.com/logic/proposit.pdf)
If something red that is not a ball is on something y, then y is a table that is not blue.

\(~[[*x] [*y] (Red ?x) ~[(Ball ?x)] (On ?x ?y) ~[(Table ?y) ~[(Blue ?y)]]\]

A one-to-one mapping of EG features to EGIF features: *

- Two ligatures of connected lines map to [*x] and [*y].
- Four ovals map to four negations, represented as \(~[\ ]\).
- Five EG relation names map to five EGIF relation names.
- Six pegs of the relations map to six occurrences of ?x or ?y.

* See http://www.jfsowa.com/pubs/egtut.pdf
Mapping EG to Predicate Calculus

EGIF:

\[ \neg[\neg \exists x \neg \exists y (\text{Red } x) \neg (\text{Ball } x) (\text{On } x y) \neg (\text{Table } y) \neg (\text{Blue } y))] \]

Translation of EGIF to predicate calculus:

\[ \neg(\exists x)(\exists y)(\text{Red}(x) \land \neg \text{Ball}(x) \land \text{On}(x,y) \land \neg(\text{Table}(y) \land \neg \text{Blue}(y))) \]

Relating EGIF to predicate calculus and ISO Common Logic.

- EG and EGIF have fewer operators, and conjunction is implicit.
- Since EG areas have no linear order, EGIF areas are also unordered.
- Labels like \(?x\) and \(?y\) show how lines and nodes are connected.
- Both EGIF and predicate calculus are subsets of Common Logic.
Coreference Nodes in EGIF

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

- **Defining node**: [*x*] represents the outermost point of a line.
- **Extension node**: [?x] shows an extension of the line x.
- **Identity**: [?x ?y] shows that lines x and y are connected to form a ligature; it corresponds to an equality $x=y$.
- **Teridentity**: [?x ?y ?z] shows a connection of lines x, y, and z; it corresponds to a pair of equalities, $x=y$ and $y=z$.
- **Coreference nodes** may connect any number of lines.

With coreference nodes, EGIF can show how each ligature is formed by connecting lines.

Rules of inference can simplify the EGIF while preserving truth.
Representing Connections in EGIF

Two ligatures, each with three branching lines.

Representing each branch with a distinct defining node in EGIF:

\[ [\text{Ball} \ ?x] [\text{Red} \ ?y] [\text{On} \ ?x \ ?y \ ?z] \]
\[ [\text{On} \ ?z \ ?u] [\text{Table} \ ?v] [\text{Blue} \ ?w] \]

Rules of inference can simplify the EGIF:

\[ [\text{Ball} \ ?x] [\text{Red} \ ?x] [\text{On} \ ?x \ ?u] [\text{Table} \ ?u] [\text{Blue} \ ?u] \]

Mapping Language to Logic

Hans Kamp observed that the algebraic notation does not have a simple mapping to and from natural languages. *

Pronouns can cross sentence boundaries, but variables cannot.

- Example: *Pedro is a farmer. He owns a donkey.*
  
- $$(\exists x)(\text{Pedro}(x) \land \text{farmer}(x)). \ (\exists y)(\exists z)(\text{owns}(y,z) \land \text{donkey}(z)).$$
  
- There is no operator that can relate $x$ and $y$ in different formulas.

The rules for scope of quantifiers are different.

- Example: *If a farmer owns a donkey, then he beats it.*
  
- In English, quantifiers in the if-clause govern the then-clause.
  
- But in predicate calculus, the quantifiers must be moved to the front.
  
- Formula: $$(\forall x)(\forall y)((\text{farmer}(x) \land \text{donkey}(y) \land \text{owns}(x,y)) \supset \text{beats}(x,y)).$$

In narratives, the default operator between NL sentences is usually equivalent to *and then.*

Translating the Word *is* to Logic

Three different translations in the algebraic notation:

- **Existence:** There is $x$. $\leftrightarrow \exists x$
- **Predication:** $x$ is a cat. $\leftrightarrow \text{Cat}(x)$
- **Identity:** $x$ is $y$. $\leftrightarrow x = y$

Do these three translations imply that English is ambiguous?

Or is the algebraic notation unnecessarily complex?

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

- **Existence:** There is $x$. $\leftrightarrow \text{—}$
- **Predication:** $x$ is a cat. $\leftrightarrow \text{—Cat}$
- **Identity:** $x$ is $y$. $\leftrightarrow \text{——}$ (a ligature of two lines)

As Peirce said, EGs are more iconic than the algebraic notation: they relate language to logic more clearly and directly.

Frege and Russell were misled by their notations.
Kamp invented Discourse Representation Structure (DRS) as a logic with a simpler mapping to and from NLs. EGs (left) and DRSes (right) support equivalent operations. By connecting EG lines or merging DRS boxes,
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 have the same mapping to NLs.

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

In both EG and DRS, quantifiers in the *if*-area are existential, and they include the *then*-area within their scope.
Disjunctions in EG and DRS

Example by 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.*
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 syntaxes, including DRS, predicate calculus, frame notations, and even natural language.

For details, see http://www.jfsowa.com/pubs/egtut.pdf.
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:

\[ ((p \supset r) \land (q \supset s)) \supset ((p \land q) \supset (r \land s)) \]

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

\[ ((p \supset r) \land (q \supset s)) \supset ((p \land q) \supset (r \land s)) \]

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 \((p \land q) \supset (r \land s)\) 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 \(((p \supset r) \land (q \supset s))\).  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 from a blank sheet of assertion:

3i, insert double negation.  1i, insert \( p \).  2i, iterate \( p \).

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 EGIF

Proof of modus ponens from a blank sheet of assertion:

1. By 3i, draw a double negation around the blank: \( \sim[\sim]\)
2. By 1i, insert the hypothesis in the negative area: \(\sim[(p)\sim]\)
3. By 2i, iterate the hypothesis into the conclusion: \(\sim[(p)\sim(p)]\)

Observations:

- EGIF is not as readable as the graphic notation for EGs.
- But there is a one-to-one correspondence of proofs in EG and EGIF.
- CGIF is a standard dialect of ISO Common Logic (CL).
- CL is a superset of a wide range of logics, including RDF and OWL.
- Therefore, EGs and EG proofs can be mapped to and from EGIF and a wide range of logics used in computer science and systems.
Proof of the Praeclarum Theorema in EGIF

1. By 3i, draw a double negation around the blank:  \( \neg[ \neg[ ] ] \)

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

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

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

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

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

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

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

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

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

Every cat is a carnivore.

Every cat in the house is an 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:

<table>
<thead>
<tr>
<th>Peirce’s Method</th>
<th>Gentzen’s Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 rules</td>
<td>16 rules</td>
</tr>
<tr>
<td>3 symmetric pairs</td>
<td>Many irregularities</td>
</tr>
<tr>
<td>Simple operations</td>
<td>Requires provability</td>
</tr>
<tr>
<td>Straight-line proofs</td>
<td>Complex bookkeeping</td>
</tr>
<tr>
<td>Date: 1897-1909</td>
<td>Date: 1935</td>
</tr>
</tbody>
</table>

For a proof of equivalence, see http://www.jfsowa.com/pubs/egtut.pdf
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.
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.
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 \( \sim p \) from \( \sim q \) 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](http://www.jfsowa.com/pubs/egtut.pdf)
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 EGIF 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 EGIF are needed for other Gamma features.

For details, see Section 5 of [http://www.jfsowa.com/pubs/eg2cg.pdf](http://www.jfsowa.com/pubs/eg2cg.pdf)
Psychological Issues

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.
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.
Sowa, John F. (2011) Peirce’s tutorial on existential graphs,  

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,  

Johnson-Laird, Philip N. (2002) Peirce, logic diagrams, and the elementary operations of reasoning,  

Pietarinen, Ahti-Veikko (2009) Peirce’s development of quantification theory,  

Pietarinen, Ahti-Veikko (2003) Peirce’s magic lantern of logic: Moving pictures of thought,  

Pietarinen, Ahti-Veikko (2011) Moving pictures of thought II, Semiotica 186:1-4, 315–331,  

Sowa, John F. (2010) Role of logic and ontology in language and reasoning,  

Sowa, John F. (2006) Peirce’s contributions to the 21st Century,  

ISO/IEC standard 24707 for Common Logic,  

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