

An Introduction to

Existential Graphs

(a diagrammatic first-order logic)

Alexander Heußner

Brussels — November 2011

Spoiler Alert / Warning

- take look at logics from broader perspective
 - ⇔ “philosophical touch” unavoidable
- suppose basic knowledge on first-order predicate calculus:
 $\wedge, \vee, \rightarrow, \exists, \forall, \dots$; Tarski-style semantics; natural deduction
- easy to grasp formalism (proof by audience participation)
- formal logic can be “intuitive” and fun 😊

Agenda

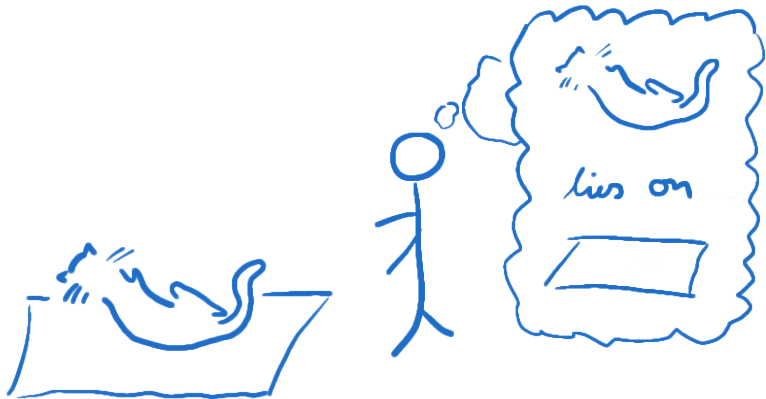
- ⇒ philosophical/historical background
- ⇒ Existential Graphs (EG)
 - basic notation and intuition
 - endoporeutic/game-based semantics
 - diagrammatic inference rules
- ⇒ retracing classical results of FOPC
- ⇒ conclusion & back to formal methods...

Agenda

- ⇒ philosophical/historical background
- ⇒ Existential Graphs (EG)
 - basic notation and intuition
 - endoporeutic/game-based semantics
 - diagrammatic inference rules
- ⇒ retracing classical results of FOPC
- ⇒ conclusion & back to formal methods...

Epistemology / Semiotics / Logics

Perception & Mental Model



Epistemology / Semiotics / Logics

Formalization

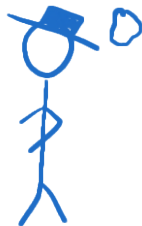


Epistemology / Semiotics / Logics

Reasoning

Cat on mat

mat is green



cat on
green
mat

Point of Departure

Question 1

How to (formally) represent [scientific¹] knowledge ?

¹including descriptions of domain experts like software engineers

Question 2

How to reason “correctly”² in this formal notational system ?

²such that formal deduction is not against “intuition”

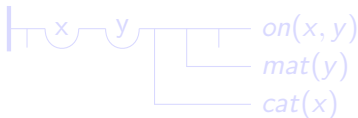
Historical Background

1854



Bool: propositional reasoning can be based on algebra

Frege: reasoning on arithmetic demands "unalgebraic" notation



"the following holds: there isn't any x and any y such that, if x is a *cat*, then x is not *on* y when y is a *mat*"

$$\neg \forall x : \forall y : cat(x) \rightarrow (mat(y) \rightarrow \neg on(x, y))$$

Peirce: general algebra of relations

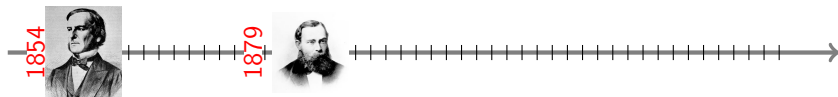
$$\sum_x \sum_y (cat_x \cdot mat_y \cdot on_{xy})$$

Peano: avoid arithmetic symbols

$$\exists x : \exists y : cat(x) \wedge mat(y) \wedge on(x, y)$$

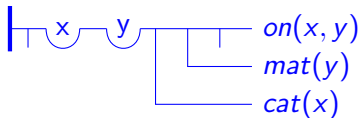
Peirce: introduces existential graphs

Historical Background



Boole: propositional reasoning can be based on algebra

Frege: reasoning on arithmetic demands "unalgebraic" notation



"the following holds: there isn't any x and any y such that, if x is a *cat*, then x is not *on* y when y is a *mat*"

$$\neg \forall x : \forall y : \text{cat}(x) \rightarrow (\text{mat}(y) \rightarrow \neg \text{on}(x, y))$$

Peirce: general algebra of relations

$$\sum_x \sum_y (\text{cat}_x \cdot \text{mat}_y \cdot \text{on}_{xy})$$

Peano: avoid arithmetic symbols

$$\exists x : \exists y : \text{cat}(x) \wedge \text{mat}(y) \wedge \text{on}(x, y)$$

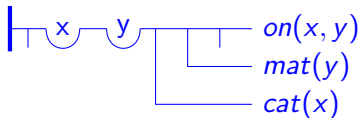
Peirce: introduces existential graphs

Historical Background



Boole: propositional reasoning can be based on algebra

Frege: reasoning on arithmetic demands "unalgebraic" notation



"the following holds: there isn't any x and any y such that, if x is a *cat*, then x is not *on* y when y is a *mat*"

$$\neg \forall x : \forall y : \text{cat}(x) \rightarrow (\text{mat}(y) \rightarrow \neg \text{on}(x, y))$$

Peirce: general algebra of relations

$$\sum_x \sum_y (\text{cat}_x \cdot \text{mat}_y \cdot \text{on}_{xy})$$

Peano: avoid arithmetic symbols $\exists x : \exists y : \text{cat}(x) \wedge \text{mat}(y) \wedge \text{on}(x, y)$

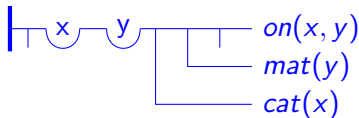
Peirce: introduces existential graphs

Historical Background



Boole: propositional reasoning can be based on algebra

Frege: reasoning on arithmetic demands "unalgebraic" notation



"the following holds: there isn't any x and any y such that, if x is a *cat*, then x is not *on* y when y is a *mat*"

$$\neg \forall x : \forall y : cat(x) \rightarrow (mat(y) \rightarrow \neg on(x, y))$$

Peirce: general algebra of relations

$$\sum_x \sum_y (cat_x \cdot mat_y \cdot on_{xy})$$

Peano: avoid arithmetic symbols $\exists x : \exists y : cat(x) \wedge mat(y) \wedge on(x, y)$

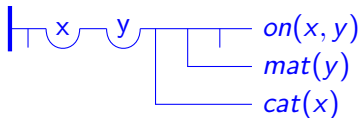
Peirce: introduces existential graphs

Historical Background



Boole: propositional reasoning can be based on algebra

Frege: reasoning on arithmetic demands "unalgebraic" notation



"the following holds: there isn't any x and any y such that, if x is a *cat*, then x is not *on* y when y is a *mat*"

$$\neg \forall x : \forall y : \text{cat}(x) \rightarrow (\text{mat}(y) \rightarrow \neg \text{on}(x, y))$$

Peirce: general algebra of relations

$$\sum_x \sum_y (\text{cat}_x \cdot \text{mat}_y \cdot \text{on}_{xy})$$

Peano: avoid arithmetic symbols $\exists x : \exists y : \text{cat}(x) \wedge \text{mat}(y) \wedge \text{on}(x, y)$

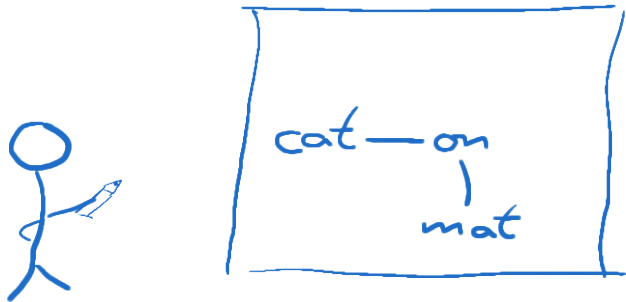
Peirce: introduces existential graphs

Peirce's Existential Graphs

I do not think I ever reflect in words: I employ visual diagrams, firstly, because this way of thinking is my natural language of self-communion, and secondly, because I am convinced that it is the best system for the purpose



Peirce's Existential Graphs



Agenda

- ⇒ philosophical/historical background
- ⇒ Existential Graphs (EG)
 - basic notation and intuition
 - endoporeutic/game-based semantics
 - diagrammatic inference rules
- ⇒ retracing classical results of FOPC
- ⇒ conclusion & back to formal methods...

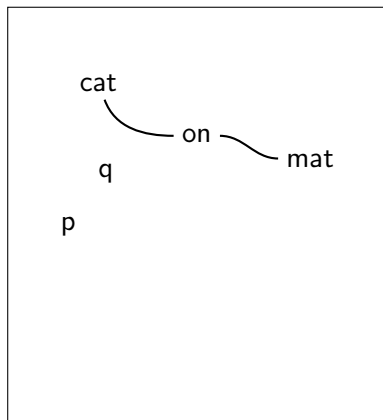
Existential Graphs : Basics

- sheet of assertion (\top)
- atoms ("n-ary relations")
- line of identity (\exists , "=")
- juxtaposition (\wedge)
- cut (\neg)
- cuts can be nested



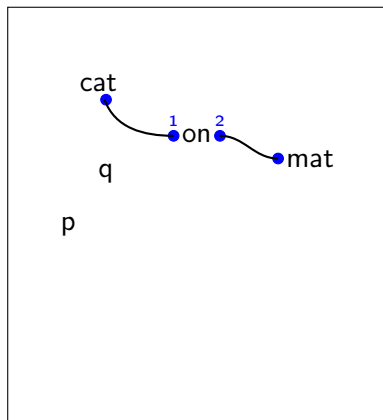
Existential Graphs : Basics

- sheet of assertion (\top)
- atoms ("n-ary relations")
- line of identity (\exists , "=")
- juxtaposition (\wedge)
- cut (\neg)
- cuts can be nested



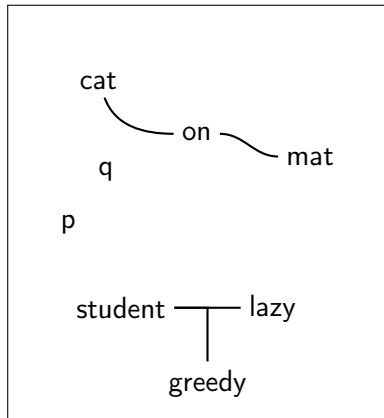
Existential Graphs : Basics

- sheet of assertion (\top)
- atoms ("n-ary relations")
- line of identity (\exists , "=")
- juxtaposition (\wedge)
- cut (\neg)
- cuts can be nested



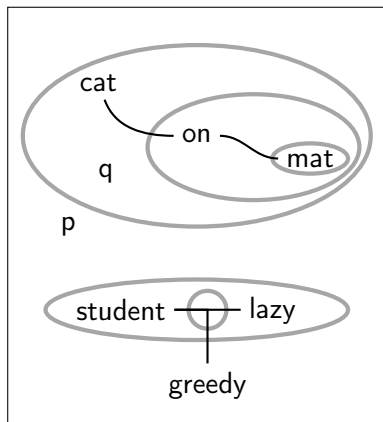
Existential Graphs : Basics

- sheet of assertion (\top)
- atoms (“n-ary relations”)
- line of identity (\exists , “=”)
- juxtaposition (\wedge)
- cut (\neg)
- cuts can be nested
odd \Rightarrow negative area (gray)
even \Rightarrow positive area (white)



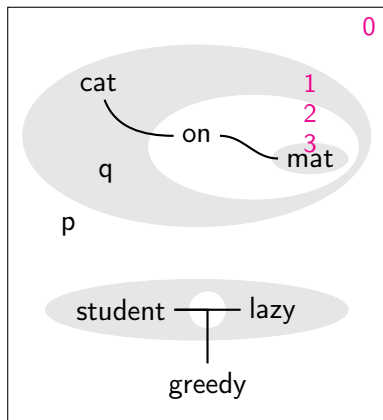
Existential Graphs : Basics

- sheet of assertion (\top)
- atoms ("n-ary relations")
- line of identity (\exists , "=")
- juxtaposition (\wedge)
- cut (\neg)
- cuts can be nested
 - odd \Rightarrow negative area (gray)
 - even \Rightarrow positive area (white)



Existential Graphs : Basics

- sheet of assertion (\top)
- atoms (“n-ary relations”)
- line of identity (\exists , “=”)
- juxtaposition (\wedge)
- cut (\neg)
- cuts can be nested
odd \Rightarrow negative area (gray)
even \Rightarrow positive area (white)



Examples

1. — student

2. — student

3. — student

Examples

1. — student

$\exists x : \textit{student}(x)$

2. — student

3. — student

Examples

1. — student

$\exists x : student(x)$

2. — student

3. — student

Examples

1. — student

$\exists x : student(x)$

2. — student

$\neg \exists x : student(x)$

3. — student

Examples

1. — student

$\exists x : student(x)$

2. — student

$\neg \exists x : student(x)$

3. — student

Examples

1. — student

$\exists x : student(x)$

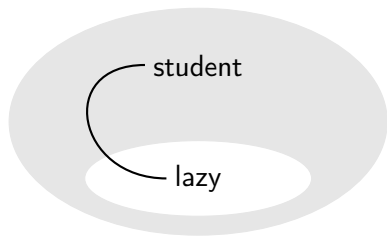
2. — student

$\neg \exists x : student(x)$

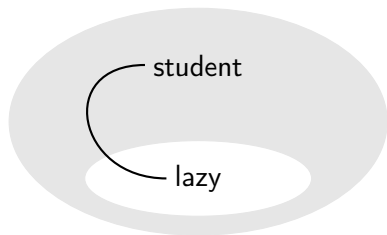
3. — student

$\exists x : \neg student(x)$

More Examples

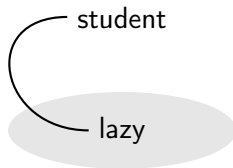
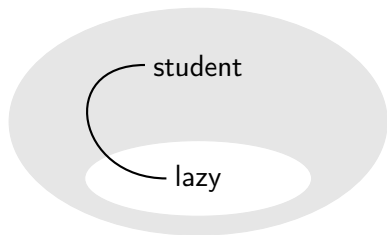


More Examples



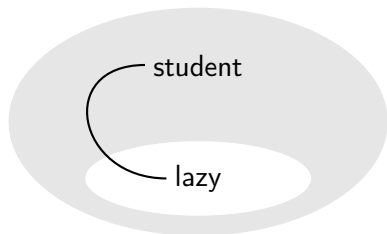
$$\neg \exists x : \textit{student}(x) \wedge \neg \textit{lazy}(x)$$
$$\equiv \forall x : \textit{student}(x) \rightarrow \textit{lazy}(x)$$

More Examples

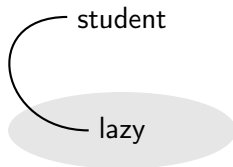


$$\neg \exists x : \textit{student}(x) \wedge \neg \textit{lazy}(x)$$
$$\equiv \forall x : \textit{student}(x) \rightarrow \textit{lazy}(x)$$

More Examples



$$\neg \exists x : \text{student}(x) \wedge \neg \text{lazy}(x)$$
$$\equiv \forall x : \text{student}(x) \rightarrow \text{lazy}(x)$$

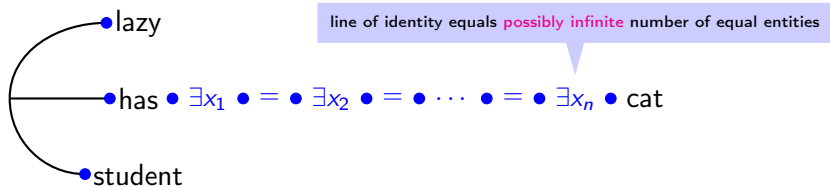


$$\exists x : \text{student}(x) \wedge \neg \text{lazy}(x)$$

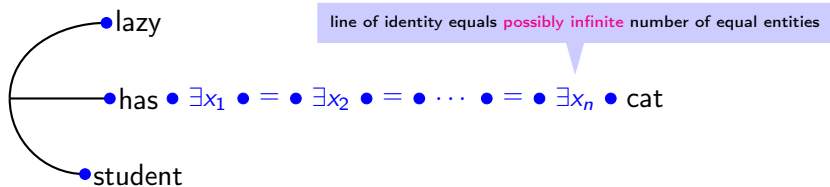
Even More Examples



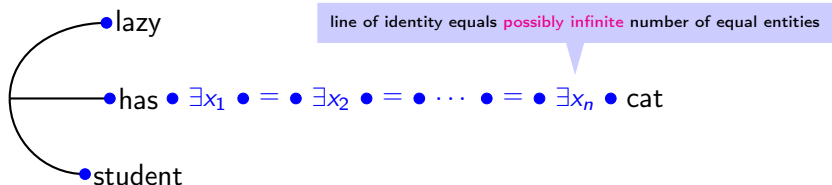
Even More Examples



Even More Examples

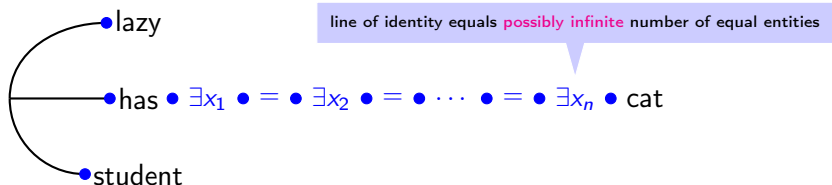


Even More Examples



$$\exists x : \exists y : p(x) \wedge p(y) \wedge \neg(x = y)$$

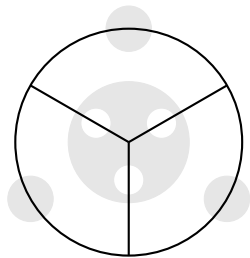
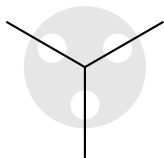
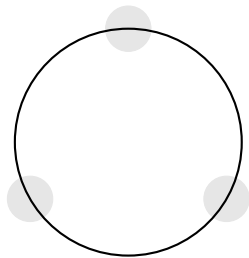
Even More Examples



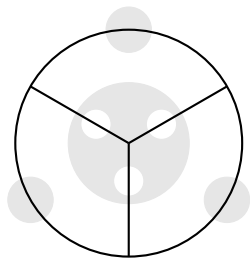
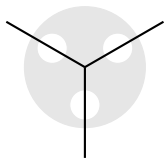
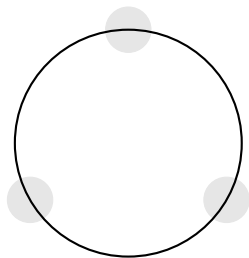
$$\exists x : \exists y : p(x) \wedge p(y) \wedge \neg(x = y)$$

$$\exists x : p(x) \wedge (\forall y : p(y) \rightarrow x = y)$$

Enough Examples !?



Enough Examples !?



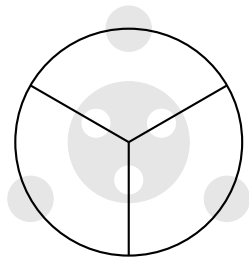
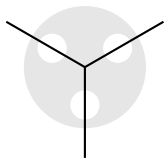
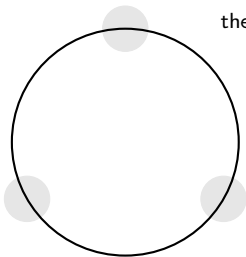
$$\exists x, y, z : x \neq y \wedge y \neq z \wedge z \neq x$$

there are at least three different things

Enough Examples !?

$\exists x, y, z : \neg \exists w : w \neq x \wedge y \neq x \wedge w \neq z$

there are at most three different things



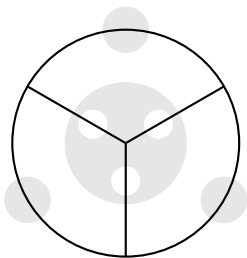
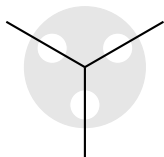
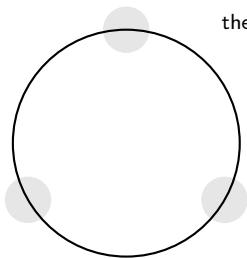
$\exists x, y, z : x \neq y \wedge y \neq z \wedge z \neq x$

there are at least three different things

Enough Examples !?

$\exists x, y, z : \neg \exists w : w \neq x \wedge y \neq x \wedge w \neq z$

there are at most three different things



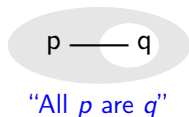
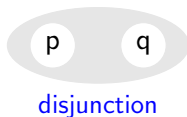
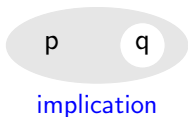
$\exists x, y, z : x \neq y \wedge y \neq z \wedge z \neq x$

there are at least three different things

there exist exactly three things

Intermediary Remarks

- reoccurring patterns



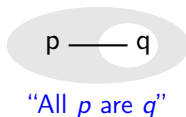
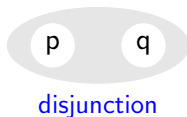
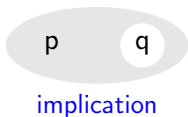
- juxtaposition is commutative & associative (free ride!)
- canonical representation for FOPC with equality



$$\begin{aligned} & p \rightarrow (q \vee r \vee s) \\ \equiv & (p \wedge \neg q) \rightarrow (r \vee s) \\ \equiv & (p \wedge \neg q \wedge \neg r) \rightarrow s \end{aligned}$$

Intermediary Remarks

- reoccurring patterns



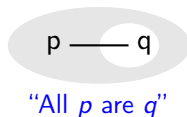
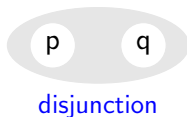
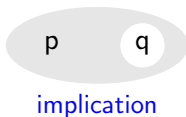
- juxtaposition is commutative & associative (free ride!)
- canonical representation for FOPC with equality



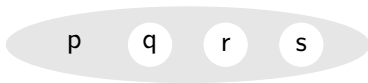
$$\begin{aligned} & p \rightarrow (q \vee r \vee s) \\ \equiv & (p \wedge \neg q) \rightarrow (r \vee s) \\ \equiv & (p \wedge \neg q \wedge \neg r) \rightarrow s \end{aligned}$$

Intermediary Remarks

- reoccurring patterns



- juxtaposition is commutative & associative (free ride!)
- canonical representation for FOPC with equality



$$\begin{aligned} p \rightarrow (q \vee r \vee s) \\ \equiv (p \wedge \neg q) \rightarrow (r \vee s) \\ \equiv (p \wedge \neg q \wedge \neg r) \rightarrow s \end{aligned}$$

Endoporeutic Semantics

- ⇒ map to standard FOPC syntax and its semantics and back ? 😞
- ⇒ “outside-in” evaluation method deriving truth value of a EG
- ⇒ lazy evaluation (can ignore certain subgraphs)
- ⇒ equivalent to two-person zero-sum perfect-information game
- ⇒ player **graphist** (proposer) vs. **grapheus** (sceptic)
- ⇒ start with one EG and a Tarski style model $M = (D, R)$
- ⇒ EG evaluates to true in M iff graphist has a **winning strategy**

Endoporeutic Semantics

- ⇒ map to standard FOPC syntax and its semantics and back ? 😞
- ⇒ “outside-in” evaluation method deriving truth value of a EG
- ⇒ lazy evaluation (can ignore certain subgraphs)

- ⇒ equivalent to two-person zero-sum perfect-information game
- ⇒ player **graphist** (proposer) vs. **grapheus** (sceptic)
- ⇒ start with one EG and a Tarski style model $M = (D, R)$
- ⇒ EG evaluates to true in M iff graphist has a **winning strategy**

Endoporeutic Semantics

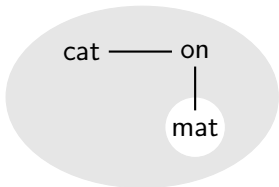
- ⇒ map to standard FOPC syntax and its semantics and back ? 😞
- ⇒ “outside-in” evaluation method deriving truth value of a EG
- ⇒ lazy evaluation (can ignore certain subgraphs)
- ⇒ equivalent to two-person zero-sum perfect-information game
- ⇒ player **graphist** (proposer) vs. **grapheus** (sceptic)
- ⇒ start with one EG and a Tarski style model $M = (D, R)$
- ⇒ EG evaluates to true in M iff graphist has a **winning strategy**

Endoporeutic Semantics

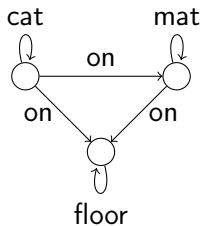
- ⇒ map to standard FOPC syntax and its semantics and back ? 😞
- ⇒ “outside-in” evaluation method deriving truth value of a EG
- ⇒ lazy evaluation (can ignore certain subgraphs)
- ⇒ equivalent to two-person zero-sum perfect-information game
- ⇒ player **graphist** (proposer) vs. **grapheus** (sceptic)
- ⇒ start with one EG and a Tarski style model $M = (D, R)$
- ⇒ EG evaluates to true in M iff graphist has a **winning strategy**

Example

Does



hold in

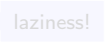


?

Endoporeutic Semantics

Game


repeat until one player wins:

- if EG is **empty** sheet of assertion, then **graphist wins**
- if EG consists of **one negative** area,
then **remove** outermost cut and **reverse roles** of players
- if EG consists of **more than one negative** areas,
sceptic **chooses one** and erases all others  laziness!
- if EG consists of subgraph G without negations and one or more negated graphs, then the graphist **proposes a graph homomorphism** of G into M (not necess. isomorphism!)

Endoporeutic Semantics

Game


repeat until one player wins:

- if EG is **empty** sheet of assertion, then **graphist wins**
- if EG consists of **one negative** area, then **remove** outermost cut and **reverse roles** of players
- if EG consists of **more than one negative** areas, sceptic **chooses one** and erases all others  **laziness!**
- if EG consists of subgraph G without negations and one or more negated graphs, then the graphist **proposes a graph homomorphism** of G into M (not neccess. isomorphism!)
 - if not possible, then **grapeus wins**
 - else this mapping leads to a partial homomorphism for future moves by assigning the same entity in D to the same line of identity, and we repeat the game on the EG where G is erased

Endoporeutic Semantics

Game


repeat until one player wins:

- if EG is **empty** sheet of assertion, then **graphist wins**
- if EG consists of **one negative** area, then **remove** outermost cut and **reverse roles** of players
- if EG consists of **more than one negative** areas, sceptic **chooses one** and erases all others  **laziness!**
- if EG consists of subgraph G without negations and one or more negated graphs, then the graphist **proposes a graph homomorphism** of G into M (not neccess. isomorphism!)
 - if not possible, then **grapheus wins**
 - else this mapping leads to a partial homomorphism for future moves by assigning the same entity in D to the same line of identity, and we repeat the game on the EG where G is erased

Endoporeutic Semantics

Game


repeat until one player wins:

- if EG is **empty** sheet of assertion, then **graphist wins**
- if EG consists of **one negative** area, then **remove** outermost cut and **reverse roles** of players
- if EG consists of **more than one negative** areas, sceptic **chooses one** and erases all others  **laziness!**
- if EG consists of subgraph G without negations and one or more negated graphs, then the graphist **proposes a graph homomorphism** of G into M (not neccess. isomorphism!)
 - if not possible, then **grapheus wins**
 - else this mapping leads to a partial homomorphism for future moves by assigning the same entity in D to the same line of identity, and we repeat the game on the EG where G is erased

Endoporeutic Semantics

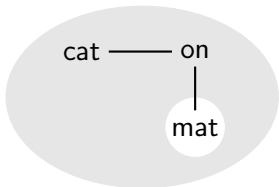
Game

repeat until one player wins:

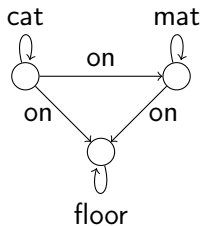
- if EG is **empty** sheet of assertion, then **graphist wins**
- if EG consists of **one negative** area, then **remove** outermost cut and **reverse roles** of players
- if EG consists of **more than one negative** areas, sceptic **chooses one** and erases all others  **laziness!**
- if EG consists of subgraph G without negations and one or more negated graphs, then the graphist **proposes a graph homomorphism** of G into M (not neccess. isomorphism!)
 - if not possible, then **grapeus wins**
 - else this mapping leads to a partial homomorphism for future moves by assigning the same entity in D to the same line of identity, and we repeat the game on the EG where G is erased

Example

Does



hold in



?

Inference Rules

basic axiom : empty sheet of assertion

- ①i any graph instance can be inserted in a **negative** area
- ①e any graph instance in a **positive** area can be erased

- ②i iterate (copy) any graph instance in the same or any enclosed nested area (lines of identity must be retained)
- ②e any graph instance generatable by ②i can be erased

- ③i a vacant ring shaped area (double cut) can be inserted anywhere
- ③e a double cut can be erased

Inference Rules

basic axiom : empty sheet of assertion

- ①i any graph instance can be inserted in a **negative** area
- ①e any graph instance in a **positive** area can be erased

- ②i iterate (copy) any graph instance in the same or any enclosed nested area (lines of identity must be retained)
- ②e any graph instance generatable by ②i can be erased

- ③i a vacant ring shaped area (double cut) can be inserted anywhere
- ③e a double cut can be erased

Inference Rules

basic axiom : empty sheet of assertion

- ①i any graph instance can be inserted in a **negative** area
- ①e any graph instance in a **positive** area can be erased

- ②i iterate (copy) any graph instance in the same or any enclosed nested area (lines of identity must be retained)
- ②e any graph instance generatable by ②i can be erased

- ③i a vacant ring shaped area (double cut) can be inserted anywhere
- ③e a double cut can be erased

Inference Rules

basic axiom : empty sheet of assertion

- ①i any graph instance can be inserted in a **negative** area
- ①e any graph instance in a **positive** area can be erased

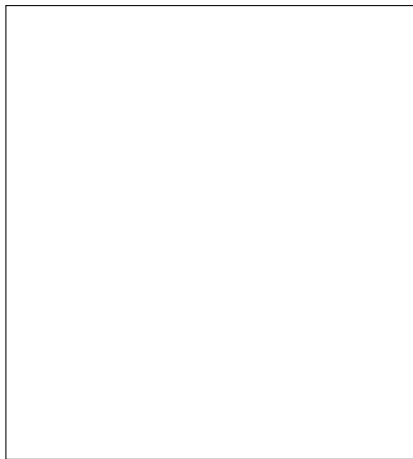
- ②i iterate (copy) any graph instance in the same or any enclosed nested area (lines of identity must be retained)
- ②e any graph instance generatable by ②i can be erased

- ③i a vacant ring shaped area (double cut) can be inserted anywhere
- ③e a double cut can be erased

Agenda

- ⇒ philosophical/historical background
- ⇒ Existential Graphs (EG)
 - basic notation and intuition
 - endoporeutic/game-based semantics
 - diagrammatic inference rules
- ⇒ retracing classical results of FOPC
- ⇒ conclusion & back to formal methods...

Re-inventing the Logical Wheel



⇒ sequence of rule applications:
axiom

recall:

Inference Rules

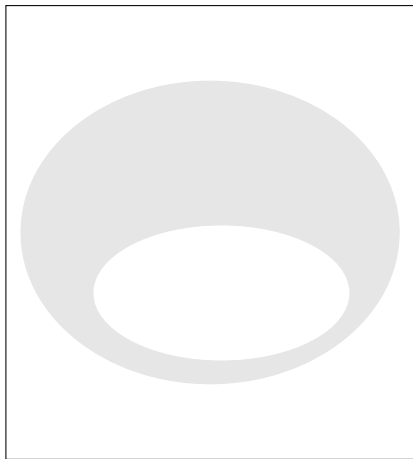
basic axiom : empty sheet of assertion

- ① any graph instance can be inserted in a **negative** area
- ①a any graph instance in a **postive** area can be erased

- ② iterate (copy) any graph instance in the same or any enclosed nested area (lines of identity must be retained)
- ②a any graph instance generatable by ② can be erased

- ③ a vacant ring shaped area (double cut) can be inserted anywhere
- ③a a double cut can be erased

Re-inventing the Logical Wheel



⇒ sequence of rule applications:
axiom; (3i)

recall:

Inference Rules

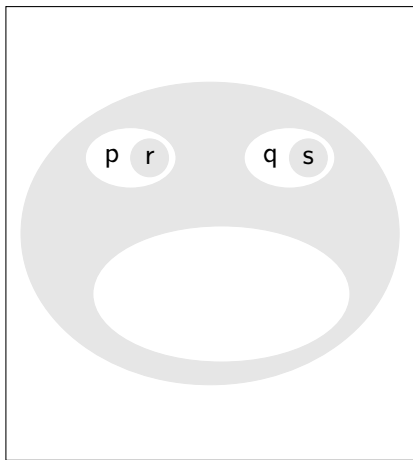
basic axiom : empty sheet of assertion

- (1i) any graph instance can be inserted in a **negative** area
- (1a) any graph instance in a **postive** area can be erased

- (2i) iterate (copy) any graph instance in the same or any enclosed nested area (lines of identity must be retained)
- (2a) any graph instance generatable by (2i) can be erased

- (3i) a vacant ring shaped area (double cut) can be inserted anywhere
- (3a) a double cut can be erased

Re-inventing the Logical Wheel



⇒ sequence of rule applications:
axiom; (3i); (1i)

recall:

Inference Rules

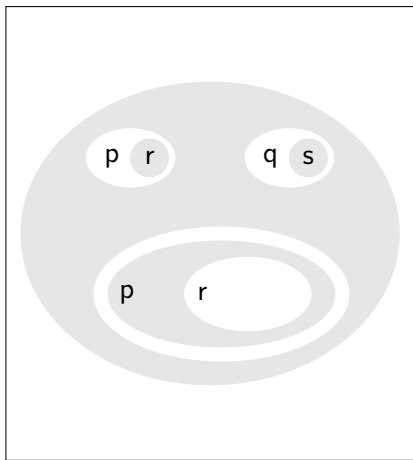
basic axiom : empty sheet of assertion

- (1i) any graph instance can be inserted in a **negative** area
- (1a) any graph instance in a **positive** area can be erased

- (2i) iterate (copy) any graph instance in the same or any enclosed nested area (lines of identity must be retained)
- (2a) any graph instance generatable by (2i) can be erased

- (3i) a vacant ring shaped area (double cut) can be inserted anywhere
- (3a) a double cut can be erased

Re-inventing the Logical Wheel



⇒ sequence of rule applications:
axiom; (3i); (1i); (2i)

recall:

Inference Rules

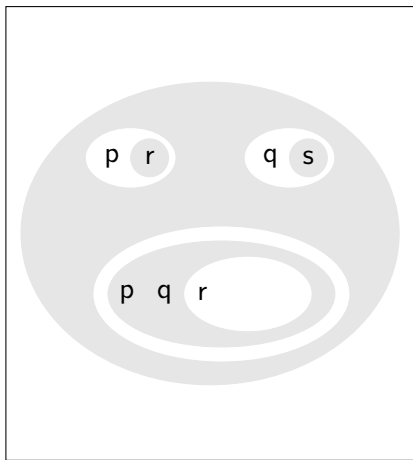
basic axiom : empty sheet of assertion

- (1i) any graph instance can be inserted in a **negative** area
- (1a) any graph instance in a **positive** area can be erased

- (2i) iterate (copy) any graph instance in the same or any enclosed nested area (lines of identity must be retained)
- (2a) any graph instance generatable by (2i) can be erased

- (3i) a vacant ring shaped area (double cut) can be inserted anywhere
- (3a) a double cut can be erased

Re-inventing the Logical Wheel



⇒ sequence of rule applications:
axiom; (3i); (1i); (2i); (1i)

recall:

Inference Rules

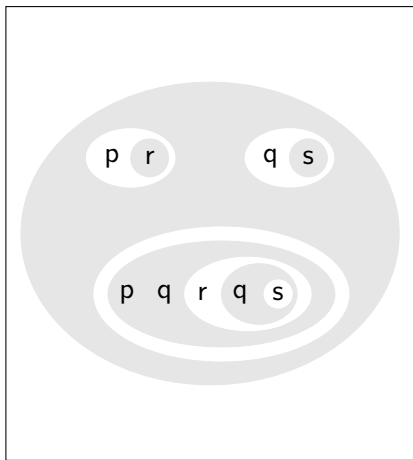
basic axiom : empty sheet of assertion

- (1i) any graph instance can be inserted in a **negative** area
- (1a) any graph instance in a **positive** area can be erased

- (2i) iterate (copy) any graph instance in the same or any enclosed nested area (lines of identity must be retained)
- (2a) any graph instance generatable by (2i) can be erased

- (3i) a vacant ring shaped area (double cut) can be inserted anywhere
- (3a) a double cut can be erased

Re-inventing the Logical Wheel



⇒ sequence of rule applications:
axiom; (3i); (1i); (2i); (1i); (2i)

recall:

Inference Rules

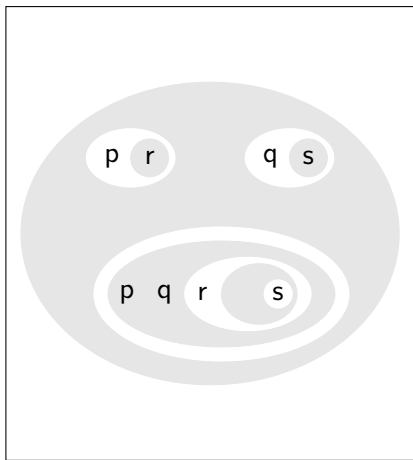
basic axiom : empty sheet of assertion

- (1i) any graph instance can be inserted in a **negative** area
- (1a) any graph instance in a **positive** area can be erased

- (2i) iterate (copy) any graph instance in the same or any enclosed nested area (lines of identity must be retained)
- (2a) any graph instance generatable by (2i) can be erased

- (3i) a vacant ring shaped area (double cut) can be inserted anywhere
- (3a) a double cut can be erased

Re-inventing the Logical Wheel



⇒ sequence of rule applications:
axiom; (3i); (1i); (2i); (1i); (2i); (2e)

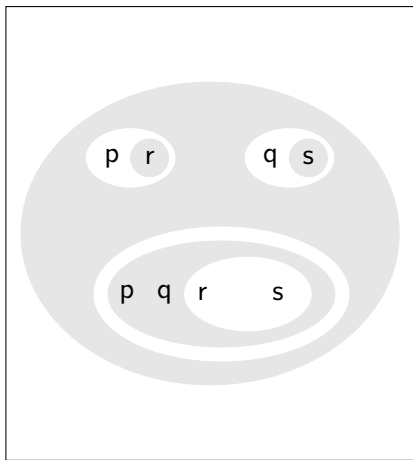
recall:

Inference Rules

basic axiom : empty sheet of assertion

- (1i) any graph instance can be inserted in a **negative** area
- (1e) any graph instance in a **positive** area can be erased
- (2i) iterate (copy) any graph instance in the same or any enclosed nested area (lines of identity must be retained)
- (2e) any graph instance generatable by (2i) can be erased
- (3i) a vacant ring shaped area (double cut) can be inserted anywhere
- (3e) a double cut can be erased

Re-inventing the Logical Wheel



⇒ sequence of rule applications:
axiom; (3i); (1i); (2i); (1i); (2i); (2e); (3e)

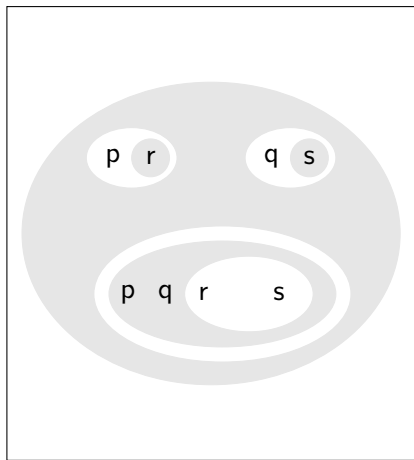
recall:

Inference Rules

basic axiom : empty sheet of assertion

- (1i) any graph instance can be inserted in a **negative** area
- (1e) any graph instance in a **positive** area can be erased
- (2i) iterate (copy) any graph instance in the same or any enclosed nested area (lines of identity must be retained)
- (2e) any graph instance generatable by (2i) can be erased
- (3i) a vacant ring shaped area (double cut) can be inserted anywhere
- (3e) a double cut can be erased

Re-inventing the Logical Wheel



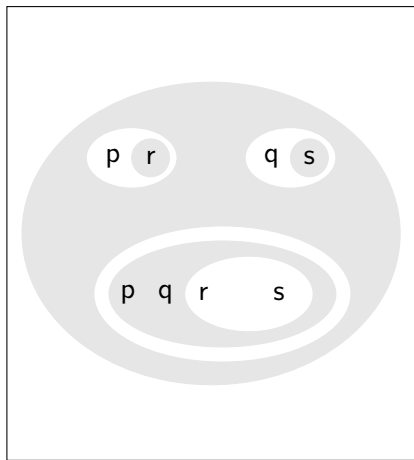
⇒ sequence of rule applications:
axiom; (3i); (1i); (2i); (1i); (2i); (2e); (3e)

⇒ derive **valid** formula:
 $[(p \rightarrow r) \wedge (q \rightarrow s)] \rightarrow [(p \wedge q) \rightarrow (r \wedge s)]$

⇒ Leibniz's Preclarum Theorem

⚠ [Principia Mathematica] needed proof in 43 steps (based on 5 axiom schemata, i.e., inf. number of axioms)

Re-inventing the Logical Wheel




⇒ sequence of rule applications:
axiom; (3i); (1i); (2i); (1i); (2i); (2e); (3e)

⇒ derive **valid** formula:
 $[(p \rightarrow r) \wedge (q \rightarrow s)] \rightarrow [(p \wedge q) \rightarrow (r \wedge s)]$

⇒ Leibniz's Preclarum Theorem

⚠ [Principia Mathematica] needed proof in 43 steps (based on 5 axiom schemata, i.e., inf. number of axioms)

Re-inventing the Logical Wheel

 Can you prove that $a \rightarrow (b \rightarrow a)$ is valid ?

recall:

Inference Rules


basic axiom : empty sheet of assertion

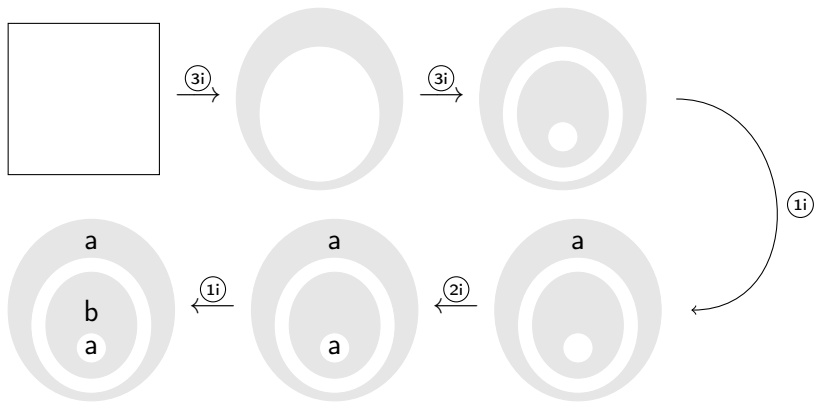
- ① any graph instance can be inserted in a **negative** area
- ①a any graph instance in a **postive** area can be erased

- ② iterate (copy) any graph instance in the same or any enclosed nested area (lines of identity must be retained)
- ②a any graph instance generatable by ② can be erased


- ③ a vacant ring shaped area (double cut) can be inserted anywhere
- ③a a double cut can be erased

Re-inventing the Logical Wheel

 Can you prove that $a \rightarrow (b \rightarrow a)$ is valid ?



Re-inventing the Logical Wheel

 Can you derive Frege's rule of *modus ponens* ?

recall:

Inference Rules

basic axiom : empty sheet of assertion

- ⑪ any graph instance can be inserted in a **negative** area
- ⑫ any graph instance in a **positive** area can be erased

- ⑬ iterate (copy) any graph instance in the same or any enclosed nested area (lines of identity must be retained)
- ⑭ any graph instance generatable by ⑬ can be erased

- ⑮ a vacant ring shaped area (double cut) can be inserted anywhere
- ⑯ a double cut can be erased

Re-inventing the Logical Wheel

 Can you derive Frege's rule of *modus ponens* ?

(hint: it is valid that from p and $p \rightarrow q$ follows q)

recall:

Inference Rules

basic axiom : empty sheet of assertion

- ⑪ any graph instance can be inserted in a **negative** area
- ⑫ any graph instance in a **positive** area can be erased

- ⑳ iterate (copy) any graph instance in the same or any enclosed nested area (lines of identity must be retained)
- ㉑ any graph instance generatable by ㉑ can be erased

- ㉓ a vacant ring shaped area (double cut) can be inserted anywhere
- ㉔ a double cut can be erased


Re-inventing the Logical Wheel

 Can you derive Frege's rule of *modus ponens*?

(hint: it is valid that from p and $p \rightarrow q$ follows q)



Re-inventing the Logical Wheel

 Can you derive Frege's rule of *modus ponens* ?

(hint: it is valid that from p and $p \rightarrow q$ follows q)



\Rightarrow now can apply modus ponens in diagrammatic proofs 

Cut-and-Paste Theorem

remark:

- inference rules only depend on whether area is negative/positive
- independent of how deeply nested

Cut-&-Paste Theorem

any proof possible on empty sheet of assertion can be “cut out” and “pasted” in any positive area of any depth

Re-inventing the Logical Wheel

Deduction Theorem

if q can be derived from p then $p \rightarrow q$.

Re-inventing the Logical Wheel

Deduction Theorem

if q can be derived from p then $p \rightarrow q$.



Agenda

- ⇒ philosophical/historical background
- ⇒ Existential Graphs (EG)
 - basic notation and intuition
 - endoporeutic/game-based semantics
 - diagrammatic inference rules
- ⇒ retracing classical results of FOPC
- ⇒ conclusion & back to formal methods...

Summary

We saw. . .

- system of EG “equivalent” to FOPC with equality
- purely diagrammatic rules of inference (“draw a proof”)
- most simple & symmetric formulation of FOPC’s inference rules
- “intuitive” formalization language (really ?)

But there is more !

- resolution algorithm for EG. . .
- graph-transformation view of inference and proofs. . .
- EG allow for reasoning in \mathbb{R} . . .
- category theory underpinning of EG. . .
- connection to boundary algebra. . .

Summary

We saw. . .

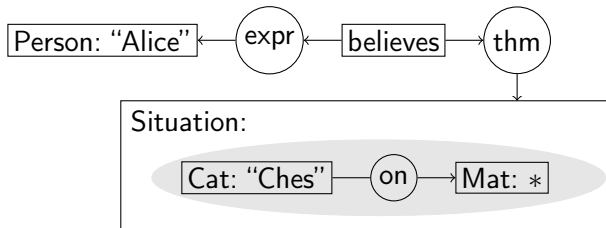
- system of EG “equivalent” to FOPC with equality
- purely diagrammatic rules of inference (“draw a proof”)
- most simple & symmetric formulation of FOPC’s inference rules
- “intuitive” formalization language (really ?)

But there is more !

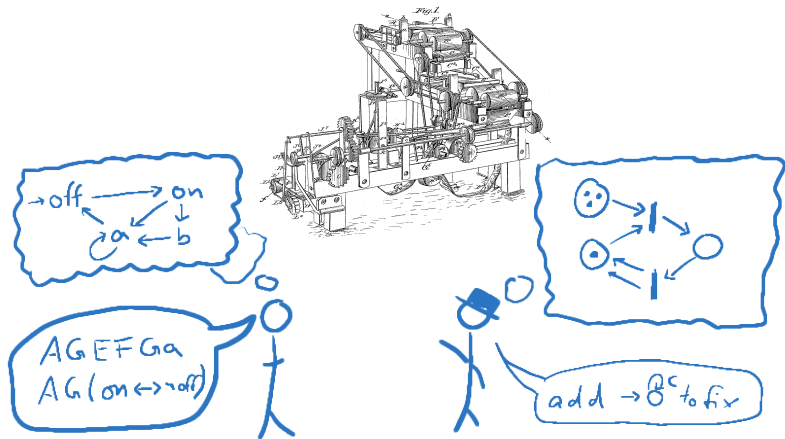
- resolution algorithm for EG. . .
- graph-transformation view of inference and proofs. . .
- EG allow for reasoning in \mathbb{R} . . .
- category theory underpinning of EG. . .
- connection to boundary algebra. . .

Derived Notions

- Peirce's γ -graphs (modal logic)
 - ⇒ introduce new kind of cut: \textcircled{p} for " $\Box p$ "
 - ⇒ extend inference rules...
- Sowa's conceptual graphs (knowledge representation)
 - ⇒ add types, type hierarchy, unique individuals,...
 - ⇒ distinguish concepts (boxes) & relations (circles)



Back to Formal Methods



Open Questions

- specification (e.g., as used in model checking or synthesis) relies on mental model of a software/hardware's behaviour
- formalization depends on (semi-)formal language
- μ -calculus et al. are expressive, ideal for algorithmic treatment, but not intuitive (= "close" to mental model)
- operational/relational model leads directly to graph-based models and graph-based reasoning
- why not focus directly logics on graphs/graph constraint systems/graph reasoning/etc. ?
- there are powerfull (wrt. to expressiveness, reasoning, etc.) formal diagrammatical languages !
- what are appropriate diagrammatical logics in our domain ?

Open Questions

- specification (e.g., as used in model checking or synthesis) relies on mental model of a software/hardware's behaviour
- formalization depends on (semi-)formal language
- μ -calculus et al. are expressive, ideal for algorithmic treatment, but not intuitive (= "close" to mental model)
- operational/relational model leads directly to graph-based models and graph-based reasoning
- why not focus directly logics on graphs/graph constraint systems/graph reasoning/etc. ?
- there are powerfull (wrt. to expressiveness, reasoning, etc.) formal diagrammatical languages !
- what are appropriate diagrammatical logics in our domain ?

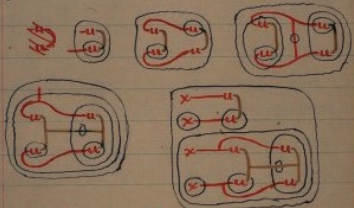
Open Questions

- specification (e.g., as used in model checking or synthesis) relies on mental model of a software/hardware's behaviour
- formalization depends on (semi-)formal language
- μ -calculus et al. are expressive, ideal for algorithmic treatment, but not intuitive (= "close" to mental model)
- operational/relational model leads directly to graph-based models and graph-based reasoning
- why not focus directly logics on graphs/graph constraint systems/graph reasoning/etc. ?
- there are powerfull (wrt. to expressiveness, reasoning, etc.) formal diagrammatical languages !
- what are appropriate diagrammatical logics in our domain ?

Open Questions

- specification (e.g., as used in model checking or synthesis) relies on mental model of a software/hardware's behaviour
- formalization depends on (semi-)formal language
- μ -calculus et al. are expressive, ideal for algorithmic treatment, but not intuitive (= "close" to mental model)
- operational/relational model leads directly to graph-based models and graph-based reasoning
- why not focus directly logics on graphs/graph constraint systems/graph reasoning/etc. ?
- there are powerfull (wrt. to expressiveness, reasoning, etc.) formal diagrammatical languages !
- what are appropriate diagrammatical logics in our domain ?

The whole theory of numbers depends upon five premises represented in this graph.



The red ligatures refer to numbers, the brown ligatures to any universe such that there is a relation that u may be understood to express, that will make the entire graph true.

x is to be understood to be replaceable by any named graph whatever without altering the truth of the entire graph.

References / Thanks

- Peirce's MS 514 with extensive commentary by J. Sowa
<http://www.jfsowa.com/peirce/ms514.htm>
- comprehensive bibliography for EG
http://existential-graphs.org/eg_readings.shtml
- MS464 from <http://www.unav.es/gep/Port/ms464/9.html>
- pictures (logicians, machine,...) from public domain