Constants and Functions in Peirce's Existential Graphs

Frithjof Dau

University of Wollongong, Australia
dau@uow.edu.au

Abstract. The system of Peirce's existential graphs is a diagrammatic version of first order logic. To be more precisely: As Peirce wanted to develop a logic of *relatives* (i.e., relations), existential graphs correspond to first order logic with relations and identity, but without constants or functions. In contemporary elaborations of first order logic, constants and functions are usually employed. In this paper, it is described how the syntax, semantics and calculus for Peirce's existential graphs has to be extended in order to encompass constants and functions as well.

1 Motivation and Introduction

It is well-known that Peirce's (1839-1914) extensively investigated a *logic of relations* (which he called 'relatives'). Much of the third volume of the collected papers [HB35] is dedicated to this topic (see for example "Description of a Notation for the Logic of Relatives, Resulting From an Amplification of the Conceptions of Boole's Calculus of Logic" (3.45–3.149, 1870) "On the Algebra of Logic" (3.154–3.251, 1880), "Brief Description of the Algebra of Relatives" (3.306–3.322, 1882), and "the Logic of Relatives" (3.456–3.552, 1897)). As Burch writes, in Peirce's thinking 'reasoning is primarily, most elementary, reasoning about *relations*' ([Bur91], p. 2, emphasis by Burch).

Starting in 1896, Peirce invented a diagrammatic form of formal logic, namely his system of existential graphs [Zem64, Rob73, Shi02, PS00, Dau06b]. The Beta part of this system corresponds to first order logic (FO) [Zem64, Dau06b]. To be more precisely: As Peirce investigated a logic of relations, the Beta part of existential graphs is equivalent to FO with relations and identity, but without constants or functions. In contrast to that, the contemporary symbolic formalizations of FO are intended to represent statements about objects, relations, and functions. This paper attempts to show how system of existential graphs has to be extended in order to cover constants and and functions as well.

This paper is part of the author's research on Sowa's conceptual graphs and Peirce's existential graphs [Dau02, Dau03, Dau06d, Dau06a, Dau06c, Dau06b]. It aims to provide a sufficiently formal elaboration of the paper's goal. For this reason, a formal elaboration of existential graphs, including their syntax, semantics, and calculus, would be needed. Due to space limitations, this is by no means possible. To resolve this problem, only those definitions and theorems of the author's treatises [Dau03, Dau06b] which are needed to keep this paper almost self-contained will be given. For the motivation of the definitions, the proofs of the theorems etc, please look at the treatises. Particularly, the scrutiny will not be carried out on a formal elaboration of existential graphs, but on discrete structures called EXISTENTIAL GRAPH INSTANCES (EGIS) instead. It shall be shortly explained how EGIs are related to the concept graphs with cuts (CGwCs) of [Dau03] and the formal existential graphs of [Dau06b].

CGwCs [Dau03] are a formal elaboration of simple conceptual graphs [Sow84, Sow92, Sow00, CM92, CM95], where the cuts of Peirce's existential graphs are added to allow for negation of subgraphs. They are elaborated in terms of mathematical graph theory. The system of CGwCs is equivalent to FO with constants, relations and identity, but without function names. It will turn out that EGIs are a restricted form of CGwCs.

In contrast to CGwCs and formulas of FO, existential graphs are not per se discrete structures. To formalize them, [Dau06b] takes a two-step approach. First, discrete structures, namely the herein presented EGIs, are introduced. An EGI can be best understood as one (of many) possible discrete formalizations of a given existential graph. Then all different EGIs which formalize the same (naive) existential graph are aggregated in a class, and each of these classes is called a *formal* existential graph. For further details, see [Dau06b].

The organization of the paper is as follows. Sec. 2 provides a short overview of the definitions and theorems of [Dau03, Dau06b] which are needed in this paper for defining the syntax and semantics of EGIs. The main task is to extend the calculus as well. In Sec. 3, the general methodology for extending the calculus is provided. Then, using this methodology, new rules for constants and function names are provided in Sec. 4, and their soundness and completeness if proven. In Sec. 5, a short example for a formal proof within the extended system of EGIs is given. Finally, Sec. 6 discusses the results of the paper and provides an outlook to further research.

2 Syntax and Semantics

The underlying structure for EGIs and CGwCs are relational graphs with cuts.

Definition 1 (Relational Graphs with Cuts). *A* RELATIONAL GRAPH WITH CUTS *is a structure* $(V, E, \nu, \top, Cut, area)$*, where*

- V*,* E *and* Cut *are pairwise disjoint, finite sets whose elements are called* VERTICES EDGES *and* CUTS*, respectively,*
- $\nu : E \to \bigcup_{k\in\mathbb{N}_0} V^k$ *is a mapping,*
- $\top$ *is a single element with* $\top \notin V \cup E \cup Cut$*, the* SHEET OF ASSERTION*, and*
- $area : Cut \cup \{\top\} \to \mathfrak{P}(V \cup E \cup Cut)$ *is a mapping which satisfies a)* $c_1 \neq c_2 \Rightarrow area(c_1) \cap area(c_2) = \emptyset$ *, b)* $V \cup E \cup Cut = \bigcup_{d\in Cut\cup\{\top\}} area(d)$*, and*

c) $c \notin area^n(c)$ for each $c \in Cut \cup \{\top\}$ and $n \in \mathbb{N}$ (with $area^0(c) := \{c\}$ and $area^{n+1}(c) := \bigcup\{area(d) \mid d \in area^n(c)\}$).

For an edge $e \in E$ with $\nu(e) = (v_1, \ldots, v_k)$ we set $|e| := k$. The vertices, edges and cuts will be called the ELEMENTS *of the graph. The elements of $Cut \cup \{\top\}$ are called* CONTEXTS. *Finally, as for every $x \in V \cup E \cup Cut$ we have exactly one context $c \in Cut \cup \{\top\}$ with $x \in area(c)$, we can write $c = area^{-1}(x)$ for every $x \in area(c)$, or even more simple and suggestive: $c = ctx(x)$.*

It is convenient to define a quasiorder $\leq$ on all elements of such a graph.

Definition 2 (Ordering on the Contexts, Enclosing Relation). *Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area)$ be a relational graph with cuts. We define a mapping $\beta : V \cup E \cup Cut \cup \{\top\} \to Cut \cup \{\top\}$ by $\beta(x) := x$ for $x \in Cut \cup \{\top\}$, and $\beta(x) := ctx(x)$ for $x \in V \cup E$. Next we set $x \leq y :\Longleftrightarrow \exists n \in \mathbb{N}_0 . \beta(x) \in area^n(\beta(y))$ We define $x < y :\Longleftrightarrow x \leq y \wedge y \not\leq x$ and $x \lneq y :\Longleftrightarrow x \leq y \wedge y \neq x$. For a context $c \in Cut \cup \{\top\}$, we set furthermore $\leq[c] := \{x \in V \cup E \cup Cut \cup \{\top\} \mid x \leq c\}$ and $\lneq[c] := \{x \in V \cup E \cup Cut \cup \{\top\} \mid x \lneq c\}$. Each element x of $\bigcup_{n \in \mathbb{N}} area^n(c)$ is said to be* ENCLOSED BY *c, and vice versa: c is said to* ENCLOSE *x. For each element of $area(c)$, we moreover say that it is* DIRECTLY ENCLOSED BY *c.*

The relation $\leq$ is indeed a quasiorder. Moreover, on the contexts, it is a tree. The proof for the following lemma can be found in [Dau03] and [Dau06b].

Lemma 1 ($\leq$ Induces a Tree on the Contexts). *For a relational graph with cuts $\mathfrak{G} := (V, E, \nu, \top, Cut, area)$, $\leq$ is a quasiorder. Furthermore, $\leq\big|_{Cut \cup \{\top\}}$ is an order on $Cut \cup \{\top\}$ which is a tree with $\top$ as greatest element.*

When defining the semantics, vertices which are deeper nested than some edge they are incident with cannot be evaluated. So this case has to be ruled out. For this reason, the next definition is needed.

Definition 3 (Dominating Nodes). *If $ctx(e) \leq ctx(v)$ ($\Leftrightarrow e \leq v$) for every $e \in E$ and $v \in V_e$, then $\mathfrak{G}$ is said to have* DOMINATING NODES.

Next, we will define EGIs to be relational graphs with cuts, where the edges are additionally labelled with names. If EGIs are used to formalize existential graphs, we would only need relation names. For the purpose of this paper, we will introduce an alphabet with names for constants, functions and relations.

Definition 4 (Alphabet with Constants, Functions and Relations). *An* ALPHABET *is a structure $(\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$ of* CONSTANT NAMES, FUNCTION NAMES *and* RELATION NAMES, *resp., together with an arity-function $ar : \mathcal{F} \cup \mathcal{R} \to \mathbb{N}$ which assigns to each function name and relation name its arity. To ease the notation, we set $ar(C) = 1$ for each $C \in \mathcal{C}$. We assume that the sets $\mathcal{C}, \mathcal{F}, \mathcal{R}$ are pairwise disjoint. The elements of $\mathcal{C} \,\dot\cup\, \mathcal{F} \,\dot\cup\, \mathcal{R}$ are the* NAMES *of the alphabet. Let $\doteq \in \mathcal{R}_2$ be a special name which is called* IDENTITY.

Later on, we will interpret an n-ary function F to be an n-ary relation which satisfies a specific property, namely: For each n objects $o_1, \ldots, o_{n-1}$ exists exactly one object o_n with $F(o_1, o_2, \ldots, o_{n-1}, o_n)$. So, functions can be understood as special relations. Please note that we adopt the arity of relations for functions. That is, an n-ary function assigns a value to $n-1$ arguments. This understanding of the arity of a function is not the common one, but it will ease the forthcoming notations. Analogously, even an object o can be understood as a special relation, namely the relation $\{(o)\}$. That is: objects correspond to unary relations which contain exactly one element (or to functions with zero arguments).

Now we are prepared to define existential graph instances (EGIs).

Definition 5 (Existential Graph Instance over $(\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$). *An* EXISTENTIAL GRAPH INSTANCE (EGI) OVER AN ALPHABET $\mathcal{A} = (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$ *is a structure* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ *where* $(V, E, \nu, \top, Cut, area)$ *is a relational graph with cuts and dom. nodes, and* $\kappa : E \to \mathcal{C} \dot\cup \mathcal{F} \dot\cup \mathcal{R}$ *is a mapping such that* $|e| = ar(\kappa(e))$ *for each* $e \in E$. *The elements of* E *with* $\kappa(e) = \doteq$ *are called* IDENTITY-EDGES. *The system of all EGIs over* $\mathcal{A}$ *will be denoted by* $\mathcal{EGI}^{\mathcal{A}}$.

As said in the introduction, formal existential graphs are in [Dau06b] defined as classes of EGIs, where only relation names occur. Those EGIs can in turn understood to be those CGwCs where only concept boxes of the form $\boxed{\top : *}$ appear. In their diagrammatic representation, we will draw the vertices, as usual in graph theory, as bold dots.

Next we define isomorphisms and partial isomorphisms between EGIs. The formal definition of a isomorphism is canonical. The rules of the calculus (like the rules of Peirce, i.e. erasure, insertion, double cut, iteration and deiteration, or the new rules presented in this paper for constants and functions) modify a graph within a given context. For this reason, we furthermore have a notion of two EGIs being isomorphic except a context.

Definition 6 ((Partial) Isomorphism). *For* $i = 1, 2$, *let two EGIs* $\mathfrak{G}_i := (V_i, E_i, \nu_i, \top_i, Cut_i, area_i, \kappa_i)$ *be given.*

An ISOMORPHISM $f = f_V \dot\cup f_E \dot\cup f_{Cut}$ *is composed of three bijective mappings* $f_V : V_1 \to V_2$, $f_E : E_1 \to E_2$ *and* $f_{Cut} : Cut_1 \cup \{\top_1\} \to Cut_2 \cup \{\top_2\}$ *which satisfy* $f_E(v_1, \ldots, v_n) = (f_V(v_1), \ldots, f_V(v_n))$ *for each* $e = (v_1, \ldots, v_n) \in E_1$, $f[area_1(c)] = area_2(f(c))$ *for each* $c \in Cut_1 \cup \{\top_1\}$ *(with* $f[area_1(c)] = \{f(k) \mid k \in area_1(c)\}$*), and* $\kappa_1(e) = \kappa_2(f_E(e))$ *for all* $e \in E_1$.

Now let furthermore two contexts $c_i \in Cut_i \cup \{\top_i\}$ *be given. For* $i = 1, 2$, *we set* $V_i' := \{v \in V_i \mid v \not\leq c_i\}$, $E_i' := \{e \in E_i \mid e \not\leq c_i\}$, *and* $Cut_i{}' := \{d \in Cut_i \cup \{\top_i\} \mid d \not< c_i\}$. *Let* $\mathfrak{G}_i'$ *be the restriction of* $\mathfrak{G}_i$ *to these sets, i.e., for* $area_i{}' := area_i\big|_{Cut_i{}'}$ *and* $\kappa_i{}' := \kappa_i\big|_{E_i{}'}$, *let* $\mathfrak{G}_i' := (V_i', E_i', \nu\big|_{E_i'}, \top_i, Cut_i', area_i', \kappa_i{}')$. *If* $f = f_{V_1'} \dot\cup f_{E_1'} \dot\cup f_{Cut_1'}$ *is an isomorphism between* $\mathfrak{G}_1'$ *and* $\mathfrak{G}_2'$ *with* $f_{Cut}(c_1) = c_2$, *then* f *is called* (PARTIAL) ISOMORPHISM FROM $\mathfrak{G}_1$ TO $\mathfrak{G}_2$ EXCEPT FOR c_1 AND c_2.

In this definition, for the restrictions $area_i{}'$ and $\kappa_i{}'$, we of course agree that the ranges of these functions are restricted to ${V_i}' \cup {E_i}' \cup {Cut_i}'$ as well. Moreover, note that this definition relies on the graph to have dominating nodes (otherwise it might happen that the structures $\mathfrak{G}_i'$ are no well-defined EGIs).

After defining the syntax for EGIs, we now turn to the semantics. First the models are defined in the usual manner known from formal logic.

Definition 7 (Relational Structures over $(\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$). *A* RELATIONAL STRUCTURE OVER AN ALPHABET $\mathcal{A} = (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$ *is a pair* $\mathcal{M} := (U, I)$ *consisting of a nonempty* UNIVERSE U *and a function* $I := I_\mathcal{C} \cup I_\mathcal{F} \cup I_\mathcal{R}$ *with*

1. $I_\mathcal{C} : \mathcal{C} \to U$,
2. $I_\mathcal{F} : \mathcal{F} \to \bigcup_{k\in\mathbb{N}} \mathfrak{P}(U^k)$ *is a mapping such that for each* $F \in \mathcal{F}$ *with* $ar(F) = k$, $I(F) \in U^k$ *is (total) function* $I(F) : U^{k-1} \to U$, *and*
3. $I_\mathcal{R} : \mathcal{R} \to \bigcup_{k\in\mathbb{N}} \mathfrak{P}(U^k)$ *is a mapping such that for each* $R \in \mathcal{F}$ *with* $ar(R) = k$, $I(R) \in U^k$ *is a relation. The name '$\doteq$' is mapped to the identity relation on* U.

When an EGI is evaluated in a relational structure (U, I), we have to assign objects of our universe of discourse U to its vertices. This is done by valuations.

Definition 8 (Valuations). *Let an EGI* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ *be given and let* (U, I) *be a relational structure over* $\mathcal{A}$. *Each mapping* $ref : V' \to U$ *with* $V' \subseteq V$ *is called a* PARTIAL VALUATION OF $\mathfrak{G}$. *If* $V' = V$, *then* ref *is called* (TOTAL) VALUATION OF $\mathfrak{G}$. *Let* $c \in Cut \cup \{\top\}$. *If* $V' \supseteq \{v \in V \mid v > c\}$ *and* $V' \cap \{v \in V \mid v \leq c\} = \emptyset$, *then* ref *is called* PARTIAL VALUATION FOR c. *If* $V' \supseteq \{v \in V \mid v \geq c\}$ *and* $V' \cap \{v \in V \mid v < c\} = \emptyset$, *then ref is called* EXTENDED PARTIAL VALUATION FOR c.

The semantics for EGIs is based on Peirce's endoporeutic method. He read and evaluated existential graphs from the outside, hence starting with the sheet of assertion, and proceeded inwardly. During this evaluation, he assigned successively values to the lines of identity. This idea is adopted in the next definition.

Definition 9 (Endoporeutic Evaluation of Graphs). *Let an EGI* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$ *be given and let* (U, I) *be a relational structure over* $\mathcal{A}$. *Inductively over the tree* $Cut \cup \{\top\}$, *we define* $(U, I) \models \mathfrak{G}[c, ref]$ *for each context* $c \in Cut \cup \{\top\}$ *and every partial valuation* $ref : V' \subseteq V \to U$ *for c:*

$(U, I) \models \mathfrak{G}[c, ref] :\Longleftrightarrow$

> ref *can be extended to an partial valuation* $\overline{ref} : V' \cup (V \cap area(c)) \to U$ *(i.e.,* $\overline{ref}$ *is an extended partial valuation for c with* $\overline{ref}(v) = ref(v)$ *for all* $v \in V'$*), such that the following conditions hold:*
>
> – $\overline{ref}(e) \in I(\kappa(e))$ *for each* $e \in E \cap area(c)$ *(edge condition))*
> – $(U, I) \not\models \mathfrak{G}[d, \overline{ref}]$ *for each* $d \in Cut \cap area(c)$ *(cut condition and iteration over* $Cut \cup \{\top\}$*))*

For $(U, I) \models \mathfrak{G}[\top, \emptyset]$ *we write* $(U, I) \models \mathfrak{G}$*. If* $\mathfrak{H}$ *is a set of EGIs and if* $\mathfrak{G}$ *is an EGI such that* $(U, I) \models \mathfrak{G}$ *for each model* (U, I) *that satisfies* $(U, I) \models \mathfrak{G}'$ *for each* $\mathfrak{G}' \in \mathfrak{H}$*, we write* $\mathfrak{H} \models \mathfrak{G}$*.*

Finally, we assume that we have a sound and complete calculus for EGIs where only relation names occur (i.e., over alphabets $(\emptyset, \emptyset, \mathcal{R}, ar)$). Moreover, we assume that this calculus is based on Peirce's rules for existential graphs (erasure, insertion, double cut, iteration and deiteration). As EGIs can be understood to be CGwCs over alphabets without names for constants or types, we can adopt the calculus of [Dau03] for this purpose. A similar calculus is provided in [Dau06b]. Both calculi contain Peirce's rules[1] and have additional rules which are needed to handle identity edges. Due to space limitations, no calculus is given here.

The rules of the common calculi for FO (Hilbert-style calculi, natural deduction, sequent calculi) allow only modifications of formulae at their top-level. In contrast to that, the rules of Peirce allow modifications of a graph inside arbitrarily deep contexts. For this reason, Peirce's rules are much more powerful, and their soundness proofs can turn out to be rather complex. For this reason, both in [Dau03] and [Dau06b], two lemmata are provided which ease the soundness proofs. The lemma which is needed in this paper is given below.

Theorem 1 (Main Thm. for Soundness, Equivalence Version). *Let EGIs* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa)$, $\mathfrak{G}' := (V', E', \nu', \top', Cut', area', \kappa')$ *be given and let* f *be an isomorphism between* $\mathfrak{G}$ *and* $\mathfrak{G}'$ *except for* $c \in Cut$ *and* $c' \in Cut'$*. Set* $Cut_c := \{d \in Cut \cup \{\top\} \mid d \not< c\}$*. Let* $\mathcal{M}$ *be a relational structure and let* $P(d)$ *be the following property for contexts* $d \in Cut_c$*: Every partial valuation* ref *for* d *satisfies* $\mathcal{M} \models \mathfrak{G}[d, ref] \iff \mathcal{M} \models \mathfrak{G}'[f(d), f(ref)]$*. Then, if* P *holds for* c*, then* P *holds for each* $d \in Cut_c$*. Particularly, If* P *holds for* c*, we have* $\mathcal{M} \models \mathfrak{G} \iff \mathcal{M} \models \mathfrak{G}'$*.*

3 General Logical Background

When considering constants and function names instead of relation names only, we have new entailments between graphs. For example, if C is a constant name, the empty sheet of assertion (semantically) entails the graph $\bullet\!\!-\!\!\!-\!\!\!- C$. Thus it must be possible to derive this graph from the empty sheet of assertion (which would not be possible if C was an 1-ary relation name). The new entailments must be reflected by the calculus, thus the calculus has to be extended in order to capture the specific properties of constants and functions. There are basically two approaches: Firstly, we can add axioms, secondly, we can add new rules to the calculus. Besides the empty sheet of assertion, Peirce's calculus for existential graphs has no axioms. To preserve this property, we will adopt he second approach. This section aims to describe the methodology how this shall be done.

[1] The iteration rule in [Dau06b] is more powerful than the iteration rule in [Dau03].

As already mentioned, constants and function names can be understood as relation names which are mapped to relations with specific properties. If we have an alphabet $\mathcal{A}' = (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$ with constants and function names, we can then consider the alphabet $\mathcal{A} := (\emptyset, \emptyset, \mathcal{C} \dot\cup \mathcal{F} \dot\cup \mathcal{R}, ar)$, where each name is now understood as relation name. In this understanding, each EGI over $\mathcal{A}'$ is an EGI over $\mathcal{A}$ as well. Moreover, if $\mathcal{M}' := (U, I')$ with $I' := I'_\mathcal{C} \cup I'_\mathcal{F} \cup I'_\mathcal{R}$ is relational structure over the alphabet $\mathcal{A}'$, then $\mathcal{M} := (U, I)$ with $I(F) := I'_\mathcal{F}(F)$ for each $F \in \mathcal{F}$, $I(R) := I'_\mathcal{R}(R)$ for each $R \in \mathcal{R}$, and $I(C) := \{(I'_\mathcal{C}(C)\}$ for each $C \in \mathcal{C}$ is the corresponding model over the alphabet $\mathcal{A}$. We implicitly identify $\mathcal{M}$ and $\mathcal{M}'$. Due to this convention, each model over $\mathcal{A}'$ is an model over $\mathcal{A}$ as well. But the models for $\mathcal{A}'$ form a subclass of the models for $\mathcal{A}$. That is, if we denote the models for $\mathcal{A}'$ with $\mathfrak{M}_2$ and the models for $\mathcal{A}$ with $\mathfrak{M}_1$, we have $\mathfrak{M}_2 \subsetneq \mathfrak{M}_1$.

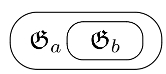
Thus we have to deal with two classes of models, which yield two entailment relations. If $\mathfrak{H}$ is a set of EGIs and if $\mathfrak{G}$ is an EGI such that $\mathcal{M} \models \mathfrak{G}$ for each relational structure $\mathcal{M} \in \mathfrak{M}_i$ with $\mathcal{M} \models \mathfrak{G}'$ for each $\mathfrak{G}' \in \mathfrak{H}$, we write $\mathfrak{H} \models_i \mathfrak{G}$.

In Sec. 2, we assumed to have a sound and complete calculus for EGIs where only relation names occur; that is, for EGIs which are evaluated in $\mathfrak{M}_1$. In the following, this calculus shall be denoted by $\vdash_1$. The soundness and completeness of $\vdash_1$ can be now stated as follows: If $\mathfrak{H} \cup \{\mathfrak{G}\}$ is a set of EGIs over $\mathcal{A}$, we have

$$\mathfrak{H} \vdash_1 \mathfrak{G} \quad \Longleftrightarrow \quad \mathfrak{H} \models_1 \mathfrak{G} \tag{1}$$

We seek a calculus $\vdash_2$ which extends $\vdash_1$ (that is, $\vdash_2$ has new rules, which will be denoted by $\vdash_2 \supseteq \vdash_1$) and which is sound and complete with respect to $\mathfrak{M}_2$.

The calculus $\vdash_1$, and hence $\vdash_2$ as well, encompasses the 5 basic-rules of Peirce. Thus for both calculi, the deduction theorem (see Lemma 6.5 of [Dau03] or Lemma. 8.7 of [Dau06b] holds, i.e., for $i = 1, 2$, we have

$$\mathfrak{G}_a \vdash_i \mathfrak{G}_b \quad \Longleftrightarrow \quad \vdash_i$$  (2)

We will extend $\vdash_1$ to $\vdash_2$ as follows: First of all, the new rules in $\vdash_2$ have to be sound. Then for a set of graphs $\mathfrak{H}$ and an EGI $\mathfrak{G}$ we have

$$\mathfrak{H} \vdash_2 \mathfrak{G} \quad \Longrightarrow \quad \mathfrak{H} \models_2 \mathfrak{G} \tag{3}$$

On the other hand, let us assume that for each $\mathcal{M} \in \mathfrak{M}_1 \backslash \mathfrak{M}_2$, there exists a graph $\mathfrak{G}_\mathcal{M}$ with

$$\vdash_2 \mathfrak{G}_\mathcal{M} \quad \text{and} \quad \mathcal{M} \not\models \mathfrak{G}_\mathcal{M} \tag{4}$$

If the last two assumptions (3) and (4) hold, we obtain that $\vdash_2$ is an adequate calculus, as the following theorem shows.

Theorem 2 (Completeness of $\vdash_2$). *A set $\mathfrak{H} \cup \{\mathfrak{G}\}$ of EGIs over an alphabet $\mathcal{A} := (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$ satisfies*

$$\mathfrak{H} \models_2 \mathfrak{G} \quad \Longrightarrow \quad \mathfrak{H} \vdash_2 \mathfrak{G}$$

Proof: Let $\mathfrak{H}_2 := \{\mathfrak{G}_\mathcal{M} \mid \mathcal{M} \in \mathfrak{M}_1 \backslash \mathfrak{M}_2\}$. From (3) we conclude: $\models_2 \mathfrak{G}_\mathcal{M}$ for all $\mathfrak{G}_\mathcal{M} \in \mathfrak{H}_2$. Now (4) yields:

$$\mathfrak{M}_2 = \{\mathcal{M} \in \mathfrak{M}_1 \mid \mathcal{M} \models \mathfrak{G} \text{ for all } \mathfrak{G} \in \mathfrak{H}_2\} \tag{5}$$

Now let $\mathfrak{H} \cup \{\mathfrak{G}\}$ be an arbitrary set of graphs. We get:

$$\begin{aligned}
\mathfrak{H} \models_2 \mathfrak{G} &\overset{\text{Def}}{\Longleftrightarrow} \text{f.a. } \mathcal{M} \in \mathfrak{M}_2 : \text{ if } \mathcal{M} \models \mathfrak{G}' \text{ for all } \mathfrak{G}' \in \mathfrak{H}, \text{ then } \mathcal{M} \models \mathfrak{G} \\
&\overset{(5)}{\Longleftrightarrow} \text{f.a. } \mathcal{M} \in \mathfrak{M}_1 : \text{ if } \mathcal{M} \models \mathfrak{G}' \text{ for all } \mathfrak{G} \in \mathfrak{H}_2 \cup \mathfrak{H}, \text{ then } \mathcal{M} \models \mathfrak{G} \\
&\Longleftrightarrow \mathfrak{H} \cup \mathfrak{H}_2 \models_1 \mathfrak{G} \\
&\overset{(1)}{\Longleftrightarrow} \mathfrak{H} \cup \mathfrak{H}_2 \vdash_1 \mathfrak{G} \\
&\Longleftrightarrow \text{ there are } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H} \text{ and } \mathfrak{G}'_1, \ldots, \mathfrak{G}'_m \in \mathfrak{H}_2 \text{ with} \\
&\qquad \mathfrak{G}_1 \quad \mathfrak{G}_2 \quad \ldots \quad \mathfrak{G}_n \quad \mathfrak{G}'_1 \quad \mathfrak{G}'_2 \quad \ldots \quad \mathfrak{G}'_m \quad \vdash_1 \mathfrak{G} \\
&\overset{(2)}{\Longleftrightarrow} \text{ there are } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H} \text{ and } \mathfrak{G}'_1, \ldots, \mathfrak{G}'_m \in \mathfrak{H}_2 \text{ with} \\
&\qquad \vdash_1 \left(\mathfrak{G}_1 \ \mathfrak{G}_2 \ldots \ \mathfrak{G}_n \ \mathfrak{G}'_1 \ \mathfrak{G}'_2 \ldots \ \mathfrak{G}'_m \ (\mathfrak{G}_b) \right) \\
&\overset{\vdash_2 \supseteq \vdash_1,\ (4)}{\Longrightarrow} \text{ there are } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H} \text{ and } \mathfrak{G}'_1, \ldots, \mathfrak{G}'_m \in \mathfrak{H}_2 \text{ with} \\
&\qquad \vdash_2 \mathfrak{G}'_1 \ldots \ \mathfrak{G}'_m \left(\mathfrak{G}_1 \ \mathfrak{G}_2 \ldots \ \mathfrak{G}_n \ \mathfrak{G}'_1 \ \mathfrak{G}'_2 \ldots \ \mathfrak{G}'_m \ (\mathfrak{G}_b) \right) \\
&\overset{\text{deit.}}{\Longleftrightarrow} \text{ there are } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H} \text{ and } \mathfrak{G}'_1, \ldots, \mathfrak{G}'_m \in \mathfrak{H}_2 \text{ with} \\
&\qquad \vdash_2 \mathfrak{G}'_1 \ldots \ \mathfrak{G}'_m \left(\mathfrak{G}_1 \ \mathfrak{G}_2 \ldots \ \mathfrak{G}_n \ (\mathfrak{G}_b) \right) \\
&\overset{\text{era.}}{\Longrightarrow} \text{ there are } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H} \text{ with } \vdash_2 \left(\mathfrak{G}_1 \ldots \ \mathfrak{G}_n \ (\mathfrak{G}_b) \right) \\
&\overset{(2)}{\Longleftrightarrow} \text{ there are } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H} \text{ with } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \vdash_2 \mathfrak{G} \\
&\overset{\text{Def.}}{\Longrightarrow} \mathfrak{H} \vdash_2 \mathfrak{G}
\end{aligned}$$

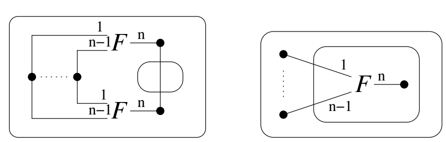
□

4 Extending the Calculus

In this section, the calculus is extended in order to capture the specific properties of constants and functions. We start the scrutiny with functions.

The following EGI holds in a model (U, I) exactly if F is interpreted as an n-ary (total) function $I(F) : U^{n-1} \to U$:

$\mathfrak{G}_F :=$



More precisely: The left subgraph is satisfied if F is interpreted as partial function (that is, to objects $o_1, \ldots, o_{n-1}$ exist at most one o_n with $I(F)(o_1, \ldots, o_n)$), the right subgraph is satisfied if for objects $o_1, \ldots, o_{n-1}$ exist at least one o_n with $I(F)(o_1, \ldots, o_n)$. In other words: The left subgraph guarantees the uniqueness, the right subgraph the existence of function values.

According to the last subsection, we have to find rules which are sound and which enable us to derive each graph $\mathfrak{G}_F$ with $F \in \mathcal{F}$. They are given below.

Definition 10 (New Rules for Function Names). *Let $F \in \mathcal{F}$ be an n-ary function name. Then all rules of the calculus, where F is treated like a relation name, may be applied. Moreover, the following additional transformations may be performed:*

- **Functional Property Rule (uniqueness of values)** *Let e, f be n-ary edges with $\nu(e) = (v_1, \ldots, v_{n-1}, v_e)$, $\nu(f) = (v_1, \ldots, v_{n-1}, v_f)$, $ctx(e) = ctx(v_e)$, $ctx(f) = ctx(v_f)$, and $\kappa(e) = \kappa(f) = F$. Let c be a context with $c \leq ctx(e)$ and $c \leq ctx(f)$. Then arbitrary identity-links id with $\nu(id) = (v_e, v_f)$ may be inserted into c or erased from c.*
- **Total Function Rule (existence of values)** *Let $v_1, \ldots, v_{n-1}$ be vertices, let c be a context with $c \leq ctx(v_1), \ldots, ctx(v_{n-1})$. Then we can add a vertex v_n and an edge e to c with $\nu(e) = (v_1, \ldots, v_n)$ and $\kappa(e) = F$. Vice versa, if v_n and e are a vertex and an edge in c with $\nu(e) = (v_1, \ldots, v_n)$ and $\kappa(e) = F$ such that v_n is not incident with any other edge, e and v_n may be erased.*

We have to show that these rules are sound are complete. We start with the soundness of the rules.

Lemma 2 (The Total Function Rule is Sound). *If $\mathfrak{G}$ and $\mathfrak{G}'$ are two EGIs over $\mathcal{A} := (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$, $\mathcal{M} := (U, I)$ is a relational structure with $\mathcal{M} \models \mathfrak{G}$ and $\mathfrak{G}'$ is derived from $\mathfrak{G}$ with the total function rule, then $\mathcal{M} \models \mathfrak{G}'$.*

Proof: Let $\mathfrak{G}'$ be obtained from $\mathfrak{G}$ by adding a vertex v_n and an edge e to c according to the total function rule. We want to apply Lemma 1 to c, so let ref be a valuation for the context c.

Let us first assume that we have $\mathcal{M} \models \mathfrak{G}[c, ref]$, i.e., there is an extension $\overline{ref}$ of ref to $V \cap area(c)$ with $\mathcal{M} \models \mathfrak{G}[c, \overline{ref}]$. Let $o := I(F)(ref(v_1, \ldots, ref(v_n))$. Then $\overline{ref}' := \overline{ref} \cup \{(v_n, o)\}$ is a extended partial valuation for c in $\mathfrak{G}'$ which satisfies $\mathcal{M} \models \mathfrak{G}[c, \overline{ref}']$, as the additional edge condition for e in the context c of $\mathfrak{G}'$ holds due to the definition of $\overline{ref}'$. Particularly, we obtain $\mathcal{M} \models \mathfrak{G}'[c, ref]$.

Now let $\mathcal{M} \models \mathfrak{G}'[c, ref]$, i.e., there is an extension $\overline{ref}'$ of ref to $V \cap area(c)$ with $\mathcal{M} \models \mathfrak{G}'[c, \overline{ref}']$. For $\overline{ref} := \overline{ref}' \backslash \{(v_n, \overline{ref}'(v_n))\}$ we have $\mathcal{M} \models \mathfrak{G}[c, \overline{ref}]$, thus $\mathcal{M} \models \mathfrak{G}[c, ref]$.

Now Lemma 1 yields the lemma. □

Lemma 3 (The Functional Property Rule is Sound). *If $\mathfrak{G}$ and $\mathfrak{G}'$ are two EGIs over $\mathcal{A} := (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$, $\mathcal{M} := (U, I)$ is a relational structure with $\mathcal{M} \models \mathfrak{G}$ and $\mathfrak{G}'$ is derived from $\mathfrak{G}$ with the functional property rule, then $\mathcal{M} \models \mathfrak{G}'$.*

Proof: Let $\mathfrak{G}'$ be obtained from $\mathfrak{G}'$ by inserting an identity-link id with $\nu(id) = (v_e, v_f)$ into c. We set $c_e := ctx(e)$ and $c_f := ctx(f)$. The EGIs $\mathfrak{G}$ and $\mathfrak{G}'$ are isomorphic except for the context c. First note that the contexts c_e and c_f must be comparable. W.l.o.g. we assume $c_e \geq c_f \geq c$.

We first consider the case $c_e = c_f = c$. We want to apply Lemma 1 to c, so let ref_c be a partial valuation for c. In $\mathfrak{G}'$ in the context c, we have added the edge id, thus for c, there is one more edge condition to check. So it suffices to prove

$$(U, I) \models \mathfrak{G}[c, ref_c] \quad \Longrightarrow \quad (U, I) \models \mathfrak{G}'[c, ref_c] \tag{6}$$

Let $(U, I) \models \mathfrak{G}[c, ref_c]$. That is, there is an extension $\overline{ref_c}$ of ref_c to $V \cap area(c)$ with $\mathfrak{G} \models \mathfrak{G}[c, \overline{ref_c}]$, i.e., $\overline{ref_c}$ satisfies all edge- and cut-conditions in c. Particularly, it satisfies the edge-conditions for e and f, that is:

$$(\overline{ref_c}(v_1), \ldots \overline{ref}(v_{n-1}), \overline{ref_c}(v_e)) \in I(\kappa(e)) \qquad \text{and}$$
$$(\overline{ref_c}(v_1), \ldots \overline{ref}(v_{n-1}), \overline{ref_c}(v_f)) \in I(\kappa(f))$$

i.e., $\overline{ref_c}(v_e) = I(F)\,(\overline{ref_c}(v_1), \ldots \overline{ref_c}(v_{n-1})) = \overline{ref_c}(v_f)$. From this we conclude that the additional edge condition for id in $\mathfrak{G}'$ is satisfied by $\overline{ref_c}$. We obtain $\mathfrak{G}' \models \mathfrak{G}[c, \overline{ref_c}]$, hence $\mathfrak{G}' \models \mathfrak{G}[c, ref_c]$, thus Eqn. (6) holds. Now Lemma 1 yields $\mathcal{M} \models \mathfrak{G} \Longleftrightarrow \mathcal{M} \models \mathfrak{G}'$.

Next we consider the case $c_e = c_f > c$. We want to apply Lemma 1 to c_e, so let ref_{c_e} be a partial valuation for c_e. To apply Lemma 1, it it suffices to prove

$$\mathfrak{G} \models \mathfrak{G}[c_e, \overline{ref_{c_e}}] \quad \Longleftrightarrow \quad \mathfrak{G}' \models \mathfrak{G}[c_e, \overline{ref_{c_e}}] \tag{7}$$

for each extension $\overline{ref_{c_e}}$ of ref_{c_e} to $area(c_e) \cap V$. So let $\overline{ref_{c_e}}$ be such an extension, If $\overline{ref_{c_e}}$ does not satisfy the edge-conditions for e and f, we have $\mathfrak{G} \not\models \mathfrak{G}[c, \overline{ref_{c_e}}]$ and $\mathfrak{G}' \not\models \mathfrak{G}[c, \overline{ref_{c_e}}]$, thus Eqn. (7) holds. So let $\overline{ref_{c_e}}$ satisfy the edge-conditions for e and f. Analogously to the case $c_e = c_f = c$ we obtain $\overline{ref_{c_e}}(v_e) = \overline{ref_{c_e}}(v_f)$. Moreover, for each extension ref_c of $\overline{ref_{c_e}}$ to a partial valuation of c, we obtain $\mathfrak{G} \models \mathfrak{G}[c, ref_c] \iff \mathfrak{G}' \models \mathfrak{G}[c, ref_c]$. This can be seen analogously to the case $c_e = c_f = c$, as $\mathfrak{G}$ and $\mathfrak{G}'$ differ only by adding the edge edge id in c, but for each extension of ref_c to $area(c) \cap V$, the edge-condition for id is due to $\overline{ref_{c_e}}(v_e) = \overline{ref_{c_e}}(v_f)$ fulfilled. Now it can easily be shown by induction that for each context d with $c_e > d \geq c$ and each extension ref_d of $\overline{ref_{c_e}}$ to $area(d) \cap V$, we have $\mathfrak{G} \models \mathfrak{G}[d, ref_d] \iff \mathfrak{G}' \models \mathfrak{G}[d, ref_d]$. This yields $\mathfrak{G} \models \mathfrak{G}[c_e, \overline{ref_{c_e}}] \iff \mathfrak{G}' \models \mathfrak{G}[c_e, \overline{ref_{c_e}}]$, i.e., Eqn. (7) holds again.

Next we consider the case $c_e > c_f > c$. The basic idea of the proof is analogous to the last cases, but we have two nested inductions. Again we want to apply Lemma 1 to c_e, so let ref_e be a partial valuation for c_e. Again we show that Eqn. (7) holds for each extension $\overline{ref_e}$ of ref_e to $area(c_e) \cap V$. Similarly to the last case, we assume that $\overline{ref_e}$ satisfies the edge-condition for e. It is sufficient to show that

$$\mathfrak{G} \models \mathfrak{G}[c_f, ref_f] \quad \Longleftrightarrow \quad \mathfrak{G}' \models \mathfrak{G}[c_f, ref_f] \tag{8}$$

holds for each each extension ref_f of $\overline{ref_e}$ to $area(c_f) \cap V$: Then similarly to the last case, an inductive argument yields that for each context d with $c_e > d \geq c_f$ and each extension ref_d of $\overline{ref_{c_e}}$ to $area(d) \cap V$, we have $\mathfrak{G} \models \mathfrak{G}[d, ref_d] \iff \mathfrak{G}' \models \mathfrak{G}[d, ref_d]$. This yields $\mathfrak{G} \models \mathfrak{G}[c_e, \overline{ref_e}] \iff \mathfrak{G}' \models \mathfrak{G}[c_e, \overline{ref_e}]$. That is, Eqn. (7) holds.

It remains to show that Eqn. (8) holds. Let us consider an extension ref_f of $\overline{ref_e}$ to $area(c_f) \cap V$. To prove Eqn. (8), it is sufficient to show that

$$\mathfrak{G} \models \mathfrak{G}[c_f, \overline{ref_f}] \quad \iff \quad \mathfrak{G}' \models \mathfrak{G}[c_f, \overline{ref_f}] \tag{9}$$

holds for each extension $\overline{ref_f}$ of ref_f to $area(c_f) \cap V$. Now we can perform the same inductive argument as in the last case. If $\overline{ref_f}$ does not satisfy the edge-condition for f, we are done. If $\overline{ref_f}$ satisfies the edge-condition, we have $\overline{ref_f}(v_e) = \overline{ref_f}(v_f)$. For each extension ref_c of $\overline{ref_f}$ to $area(c) \cap V$, we obtain $\mathfrak{G} \models \mathfrak{G}[c, ref_c] \iff \mathfrak{G}' \models \mathfrak{G}[c, ref_c]$. Now from the usual inductive argument we obtain that for each context d with $c_f > d \geq c$ and each extension ref_d of $\overline{ref_f}$ to $area(d) \cap V$, we have $\mathfrak{G} \models \mathfrak{G}[d, ref_d] \iff \mathfrak{G}' \models \mathfrak{G}[d, ref_d]$. From this we conclude that Eqn. (9), thus Eqn. (8), holds. This finishes the proof for the case $c_e > c_f > c$.

Finally, the cases $c_e > c_f = c$ and $c_f > c_e = c$ can be handled analogously. □

Next, the new rules for constants are introduced. As already been mentioned, it is well-known that functions f with zero arguments correspond to objects in the universe of discourse. For this reason, a distinction between constants and function names is, strictly speaking, not necessary. So the rules for constant names correspond to rules for 1-ary functions (i.e. functions f with $dom(f) = \emptyset$).

Definition 11 (New Rules for Constant Names). *Let $C \in \mathcal{C}$ be a constant name. Then all rules of the calculus, where F is treated like a relation name, may be applied. Moreover, the following additional transformations may be performed:*

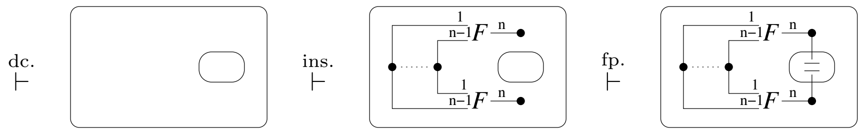
- **Constant Identity Rule** *Let e, f be two unary edges with $\nu(e) = (v_e)$, $\nu(f) = (v_f)$, $ctx(v_e) = ctx(e)$, $ctx(v_f) = ctx(f)$,and $\kappa(e) = \kappa(f) = C$. Let c be a context with $c \leq ctx(e)$ and $c \leq ctx(f)$. Then arbitrary identity-links id with $\nu(id) = (v_e, v_f)$ may be inserted into c or erased from c.*
- **Existence of Constants Rule** *In each context c, we may add a fresh vertex v and an fresh unary edge e with $\nu(e) = (v)$ and $\kappa(e) = C$. Vice versa, if v and e are a vertex and an edge in c with $\nu(e) = (v)$ and $\kappa(e) = F$ such that v is not incident with any other edge, e and v may be erased from c.*
  That is: Devices ●— C may be inserted into or erased from c.

As objects are handled like 1-ary functions, we immediately obtain the soundness of the rules from Lem. 2 and Lem. 3.

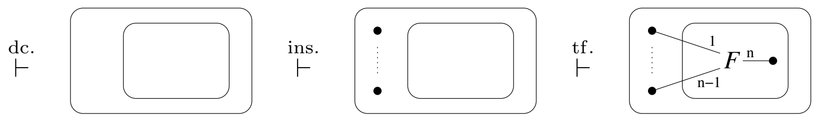
It remains to prove the completeness of the extended calculus. This is subject of the next theorem.

Theorem 3 (Extended Calculus is Complete). *Each set* $\mathfrak{H} \cup \{\mathfrak{G}\}$ *of EGIs over* $\mathcal{A} := (\mathcal{C}, \mathcal{F}, \mathcal{R}, ar)$ *satisfies* $\mathfrak{H} \models \mathfrak{G} \Rightarrow \mathfrak{H} \vdash \mathfrak{G}$*.*

Proof: Due to the remark before Def. 11 and Thm. 2, it is sufficient to show that for each $F \in \mathcal{F}$, the graph $\mathfrak{G}_F$ can be derived with the new rules. The functional property rule (abbreviated by fp) enables us to derive the left subgraph of $\mathfrak{G}_F$ as follows:



The right subgraph of $\mathfrak{G}_F$ can be derived with the total function rule (tf):



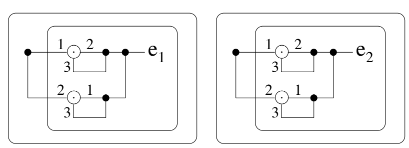
So we can derive $\mathfrak{G}_F$ as well, thus we are done. □

5 An Example for a Proof with Constants and Functions

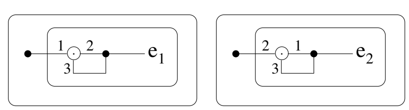
In this section, an example for a formal proof with EGIs is provided. We prove a trivial fact in group theory, namely the uniqueness of neutral elements. Assume that e_1 and e_2 are neutral elements, i.e. we have $\forall x. x \cdot e_1 = e_1 = e_1 \cdot x$ and $\forall x. x \cdot e_2 = e_2 = e_2 \cdot x$. From this we can conclude $e_1 = e_2$.

In the following, a formal proof with EGIs for this fact is provided. We assume that e_1, e_2 are employed as constant names and $\cdot$ as function name.

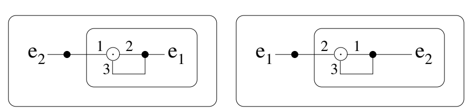
We start with the assumption that e_1, e_2 are neutral elements, i.e.



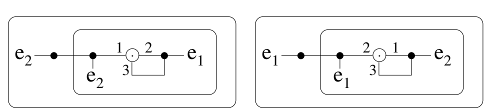
Erasure yields:



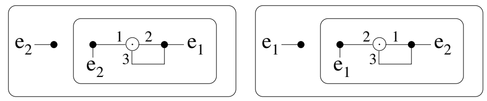
First, we insert e_1 and e_2 (i.e., edges which are labeled with e_1 and e_2) as follows:



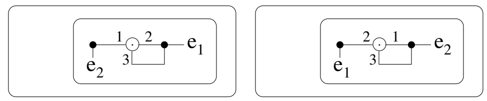
The edges are iterated:



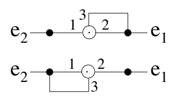
Now we can remove the identity edges with the constant identity rule.



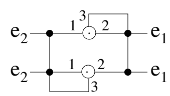
The next graph is derived with the existence of constants rule.



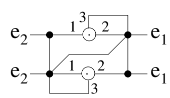
Next, we remove the double cuts and rearrange the graph.



We can insert identity edges with the constant identity rule.



The functional property rule now allows to add another identity edge.



The erasure rule finally yields:

As this graph expresses that e_1 and e_2 are identical, we are done.

6 Discussion and Outlook

Existential graphs should not be understood as a diagrammatic version of the specific form of FO where only relations are used. Instead, they can be modified to suit other purposes as well. In this paper, it has been shown how they have to be modified to cover constants and functions as well. Together with the general, formal elaboration of existential graphs in [Dau06b], we see that the systems conforms the needs of contemporary formal logic.

The approach presented in this paper is somewhat generic Al. Nonetheless, the set of the new rules depends on the syntactical implementation of constants and functions. In CGwCs, constant names are assigned to the vertices instead of the edges. Although the expressivity of the system remains the same, we have new syntactical possibilities to express a given statement. For this reason, further rules in the calculus are needed. A deep discussion on this can be found in [Dau06b].

Further research is in progress to show how existential graphs can be tailored to formalize other kinds of logics as well. For example, for Description Logics, such an approach is started in [DE06], where the syntax and semantics of an fragment of existential graphs is provided which corresponds to the Description Logic $\mathcal{ALCI}$, which is basically the smallest propositionally closed DL. Further research, which will probably take advantage of the gamma part of existential graphs, will be undertaken to provide an adequate calculus is well.

References

[Bur91] Robert W. Burch. *A Peircean Reduction Thesis: The Foundation of Topological Logic*. Texas Tech. University Press, Texas, Lubbock, 1991.

[CM92] M. Chein and M.-L. Mugnier. Conceptual graphs: Fundamental notions. *Revue d'Intelligence Artificiell*, 6:365–406, 1992.

[CM95] M. Chein and M.-L. Mugnier. Conceptual graphs are also graphs. Technical report, LIRMM, Université Montpellier II, 1995. Rapport de Recherche 95003.

[Dau02] Frithjof Dau. An embedding of existential graphs into concept graphs with negations. In Uta Priss, Dan Corbett, and Galia Angelova, editors, *ICCS*, volume 2393 of *LNAI*, pages 326–340, Borovets, Bulgaria, July, 15–19, 2002. Springer, Berlin – Heidelberg – New York.

[Dau03] Frithjof Dau. *The Logic System of Concept Graphs with Negations and its Relationship to Predicate Logic*, volume 2892 of *LNAI*. Springer, Berlin – Heidelberg – New York, November 2003.

[Dau06a] Frithjof Dau. Fixing shin's reading algorithm for peirce's existential graphs. In Dave Barker-Plummer, Richard Cox, and Nik Swoboda, editors, *Diagrams*, volume 4045 of *LNAI*, pages 88–92. Springer, Berlin – Heidelberg – New York, 2006.

[Dau06b] Frithjof Dau. Mathematical logic with diagrams, based on the existential graphs of peirce. Habilitation thesis. To be published. Available at: `http://www.dr-dau.net`, 2006.

[Dau06c] Frithjof Dau. The role of existential graphs in peirce's philosophy. In Peter Øhrstrøm, Henrik Schärfe, and Pascal Hitzler, editors, *ICCS*, pages 28–41. Aalborg University Press, 2006.

[Dau06d] Frithjof Dau. Some notes on proofs with alpha graphs. In Peter Øhrstrøm, Henrik Schärfe, and Pascal Hitzler, editors, *ICCS*, volume 4068 of *Lecture Notes in Computer Science*, pages 172–188. Springer, Berlin – Heidelberg – New York, 2006.

[DE06] Frithjof Dau and Peter Eklund. Towards a diagrammatic reasoning system for description logics. Submitted to the Journal of Visual Languages and Computing, Elsevier. Available at `www.kvocentral.org`., 2006.

[HB35] Weiss Hartshorne and Burks, editors. *Collected Papers of Charles Sanders Peirce*, Cambridge, Massachusetts, 1931–1935. Harvard University Press.

[PS00] Charles Sanders Peirce and John F. Sowa. Existential Graphs: MS 514 by Charles Sanders Peirce with commentary by John Sowa, 1908, 2000. Available at: `http://www.jfsowa.com/peirce/ms514.htm`.

[Rob73] Don D. Roberts. *The Existential Graphs of Charles S. Peirce*. Mouton, The Hague, Paris, 1973.

[Shi02] Sun-Joo Shin. *The Iconic Logic of Peirce's Graphs*. Bradford Book, Massachusetts, 2002.

[Sow84] John F. Sowa. *Conceptual structures: information processing in mind and machine*. Addison-Wesley, Reading, Mass., 1984.

[Sow92] John F. Sowa. Conceptual graphs summary. In T. E. Nagle, J. A. Nagle, L. L. Gerholz, and P. W. Eklund, editors, *Conceptual Structures: current research and practice*, pages 3–51. Ellis Horwood, 1992.

[Sow00] John F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole, Pacific Grove, CA, 2000.

[Zem64] Jay J Zeman. *The Graphical Logic of C. S. Peirce*. PhD thesis, University of Chicago, 1964. Available at: `http://www.clas.ufl.edu/users/jzeman/`.