Variables in Concept Graphs

Frithjof Dau

Technische Universität Darmstadt, Fachbereich Mathematik
Schloßgartenstr. 7, D-64289 Darmstadt, dau@mathematik.tu-darmstadt.de

Abstract A main feature of many logics used in computer science is a means to express quantification. Usually, syntactical devices like variables and quantifiers are used for this purpose. In contrast to that, in conceptual graphs, a single syntactical item, the generic marker '$*$' is used. Nonetheless, sometimes conceptual graphs with variables have to be considered. If the generic marker is replaced by variables, it has to be investigated how this syntactical difference is reflected by the semantics and transformation rules for conceptual graphs. In this paper, this task is carried out for the system of concept graph with cuts (CGwCs). Two different classes of CGwCs with variables is introduced, and for both, a semantics and an adequate calculus for CGwCs is provided.

1 Introduction

A main feature of many logics used in computer science, like the different versions of description logic, first oder logic (FOL), conceptual graphs or the resource description framework (RDF), is a means to express quantification.

In any linear symbolic notion of logic (like the common notions for FOL or description logic), variables and quantifiers are the syntactical entities which are used for this purpose. Even in RDF, which can be understood as a diagrammatic reasoning system, so-called *blanks* are used to express existential quantification, which are very much used like variables.

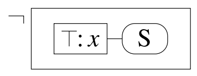
The use of variables has some consequences for the handling of formulas. A variable may occur several times in a formula. In order to grasp the meaning of a formula, one has to keep track of these different occurrences, particularly, whether they are in the same scope of an (existential or universal) quantifier. Particularly, we have to distinguish between variables and their occurrences in a formula. Moreover, one needs some means to rename variables, which has to be captured either by a convention like the so-called *alpha conversion* of formulas, or by some rules in the calculus.

For conceptual graphs, the situation is different. There exists only one syntactical element which is used to express existential quantification: the generic marker '$*$'. In a conceptual graph, different occurrences of '$*$' as referent in different concept boxes refer to (not necessarily) different entities. As only one sign for quantification is used, one does not have to keep track of generic markers. Moreover, in the handling of conceptual graphs, there is no need to distinguish between the sign and its occurrences, and a convention like the alpha-conversion of symbolic logic is not needed. This makes conceptual graphs easier to read and handle by humans, which is one of the mains goals of conceptual graphs. Using '$*$' instead of variables is a strong means to achieve this goal.

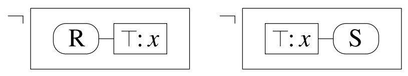
Nonetheless, sometimes it is reasonable to use variables in conceptual graphs as well. For example, finding a *linear* notation for conceptual graphs (the linear form) is much easier if generic markers are replaced by variables.[1] The use of variables instead of generic markers in any fragment of conceptual graphs has to be reflected by the semantics, as well by any calculus. Approaches for some specific forms of concept graphs with variables are investigated be Klinger in [7, 8] and Wille in [14, 15]. But, to the best of my knowledge, a comprehensive discussion between the differences of conceptual graphs with variables and conceptual graphs with generic markers has not been provided yet.

For simple conceptual graphs, which do not provide nestings or any means of negation, the use of variables instead of generic markers does not lead to major . The definition of their syntax and semantics is straightforward. Moreover, a calculus based on projections can easily provided. For example, the projections of [2] for conceptual graphs or the projection-like mappings for RDF (see [6, 1]) could be adopted for simple conceptual graphs, or the diagrammatic calculus of a work in preparation ([4]) could be used. But we run into problems if we add a means to express negation to conceptual graphs.

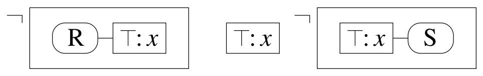
Recall that in conceptual graphs, a negation box $\neg\boxed{\quad\mathfrak{G}\quad}$ is used to negate the enclosed subgraph $\mathfrak{G}$. Let us first consider some simple examples to see some problems we have to cope with. Assume that we allow that arbitrary concept boxes are labeled with the same variable. Consider the following two graphs:



and



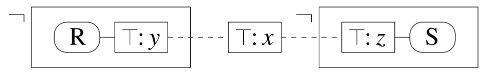
The intuitive meaning of the left graph is clear: It is 'it is not true that there exists an x with $S(x)$', i.e., no object has the property R. But what about the right graph? As we have no special sign like the quantifiers of symbolic logic for indicating the scope of variables, it is usually assumed that all variables in a conceptual graph denote the same object, thus the scope of a variable (i.e., the context where the existential quantification takes place) is the innermost context which contains all boxes which are labeled with x. For our right graph, this is the sheet of assertion. Due to this problem, Sowa considers only conceptual graphs where each coreference set has a *defining label.* The graph we consider is semantically equivalent to the graph below which has defining labels.



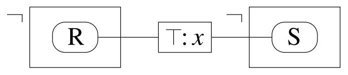
[1] It is well known that conceptual graphs are based on Peirce's diagrammatical *existential graphs*, where *lines of identity* are used to express existential quantification and identity. But even Peirce replaced in some places the lines of identity by variables, which he called *selectives* (but he strongly recommended to avoid them).

So, if we add a means to express negation, we have to be careful in the definition of the syntax and semantics of conceptual graphs with variables.

It is possible to consider conceptual graphs with variables where each variable occurs almost once. Or graph is equivalent to each of the two graphs below which satisfy this restriction.



and



Of course, the semantical equivalence of all the graphs we considered has to reflected by any calculus as well, which shows that in any calculus, we need rules which meet the specific properties of variables. For example, we need rules which allow to rearrange boxes which are labeled with the same variable, and we have to discuss how a renaming of variables has to he handled. A calculus like this goes beyond a simple one-to-one-translation of the calculus for CGwCs with generic markers.

In [3], the system of Concept Graphs with Cuts (CGwCs) is comprehensively investigated. CGwCs can be understood as a mathematical elaboration (in terms of mathematical graph theory) of conceptual graphs with negation boxes, where the negation boxes are replaced by a syntactical elements called *cuts*, and the coreference links are replaced by identity edges. Recall that cuts are graphically represented as bold ovals. In [3], a contextual semantics and a sound and complete calculus, based on Peirce's calculus for existential graphs, for CGwCs is provided.

In this paper, the differences between generic markers and variables are discussed. The scrutiny will be carried out on CGwCs, i.e. it will be shown how the results of [3] for CGwCs with generic markers can be transferred to CGwCs with variables.

In the next section, the basic definitions for CGwCs with variables are provided. In the third section, we start our investigation with a discussion of CGwCs with variables where each variable occurs almost once. These graphs will be called *purified* CGwCs with variables. This is no loss of expressivity, as each CGwCs with variables is semantically equivalent to a purified one. Purified CGwCs stand in one-to-one-correspondence to CGwCs with generic markers. Based on this correspondence, the adequate calculus for CGwCs with generic markers is translated into an adequate calculus for purified CGwCs with variables. This will be done in the next section of this paper.

In the fourth section, we extend the class of purified CGwCs with variables to CGwCs with variables where variables may occur more than once. The calculus of the preceeding section is extended in order to obtain an adequate calculus for this bigger class of CGwCs with variables. Finally, a short discussion of the results is provided.

2 Basic Definitions and Semantics

We start with the definition of the underlying alphabet for CGwCs.

Definition 1 (Alphabet).

1. *Let* $Var := \{x_1, x_2, \ldots\}$ *be a countably infinite set. The elements of* Var *are called* variables*. Let* $*$ *be a further sign, which is called the* generic marker*.*
2. *An* alphabet *is a triple* $\mathcal{A} := (\mathcal{G}, \mathcal{C}, \mathcal{R})$ *where* $\mathcal{G}$ *is a finite set of* object names*,* $(\mathcal{C}, \leq_{\mathcal{C}})$ *is a finite ordered set of* concept names *with a greatest element* $\top$*, and* $(\mathcal{R}, \leq_{\mathcal{R}})$ *is a family of finite ordered sets* $(\mathcal{R}_k, \leq_{\mathcal{R}_k})$*,* $k = 1, \ldots, n$ *of* relation names*. Let* $id \in \mathcal{R}_2$ *be a special name which is called* identity*.*

Next, we define different versions of CGwCs with different means to express existential quantification. In [3], CGwCs with generic markers are provided as mathematical structures $(V, E, \nu, \top, Cut, area, \kappa, \rho)$, where $\rho : V \to \mathcal{G} \cup \{*\}$ is a mapping which assigns to each vertex $v \in V$ its reference $\rho(v)$, being an object name or the generic marker. Now we consider a modified version of these graphs with a label mapping $\rho : V \to \mathcal{G} \cup \mathrm{Var}$. Let $\mathfrak{G}$ be such a graph. Let $\mathrm{Var}^{\mathfrak{G}} := \{\alpha \in \mathrm{Var} \mid \exists v \in V : \rho(v) = \alpha\}$ be the set of the variables which occur in $\mathfrak{G}$. In the introduction, we already discussed that the *scope* of a variable α is the innermost context which contains all vertices labeled with x. Mathematically, for $\alpha \in \mathrm{Var}^{\mathfrak{G}}$, we set $\mathit{scope}(\alpha) := \bigvee\{c \in Cut \cup \{\top\} \mid \exists v \in area(c) : \rho(v) = \alpha\}$. Finally, similar to the definition of V^* and $V^{\mathcal{G}}$ in [3], we set $V^{\mathrm{Var}} := \{v \in V \mid \rho(v) \in \mathrm{Var}\}$, and the vertices in V^{Var} are called *variable vertices* or *variable boxes*.

Similar to Sowa's defining labels of coreference sets, we define *dominating variable boxes* to be variable boxes $v \in V$ which satisfy $\mathit{cut}(v) = \mathit{scope}(\rho(v))$. For such a v, each $w \in V$ with $\rho(w) = \rho(v)$ is said to be *dominated by* v. If we otherwise have $\rho(w) \neq \rho(v)$ for each $w \in V$ with $w \neq v$, the dominating box v will be called *singular variable box*.

Now we can define the two classes of CGwCs with variables which will be discussed in this paper. A graph $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa, \rho)$ with $\rho[V] \subseteq \mathcal{G} \cup \mathrm{Var}$ is called *concept graph with cuts (CGwC) and variables over* $\mathcal{A}$, if for each $\alpha \in \mathrm{Var}^{\mathfrak{G}}$ there exists a dominating variable box v with $\rho(v) = \alpha$. If we have moreover $v_1 = v_2$ for all $v_1, v_2 \in V$ with $\rho(v_1) = \rho(v_2) \in \mathrm{Var}$, then $\mathfrak{G}$ is called *purified concept graph with cuts (CGwC) and variables over* $\mathcal{A}$.

As in [3], we use power context families as model structures for concept graphs. The models are defined as follows:

Definition 2 (Models). *A* power context family $\vec{\mathbb{K}} := (\mathbb{K}_i)_{k=0,\ldots,n}$ *is a family of contexts* $\mathbb{K}_k := (G_k, M_k, I_k)$ *that satisfies* $G_k \subseteq (G_0)^k$ *for each* $k = 1, \ldots, n$*.*

For an alphabet $\mathcal{A} := (\mathcal{G}, \mathcal{C}, \mathcal{R})$ *and a power context family* $\vec{\mathbb{K}}$*, we call the union* $\lambda := \lambda_{\mathcal{G}} \dot{\cup} \lambda_{\mathcal{C}} \dot{\cup} \lambda_{\mathcal{R}}$ *of the mappings* $\lambda_{\mathcal{G}}: \mathcal{G} \to G_0$*,* $\lambda_{\mathcal{C}}: \mathcal{C} \to \underline{\mathfrak{B}}(\mathbb{K}_0)$ *and* $\lambda_{\mathcal{R}}: \mathcal{R} \to \mathfrak{R}_{\vec{\mathbb{K}}}$ *a* $\vec{\mathbb{K}}$-interpretation *of* $\mathcal{A}$ *if* $\lambda_{\mathcal{C}}$ *and* $\lambda_{\mathcal{R}}$ *are order-preserving,* $\lambda_{\mathcal{C}}(\top) = \top$*,* $\lambda_{\mathcal{R}}(\mathcal{R}_k) \subseteq \underline{\mathfrak{B}}(\mathbb{K}_k)$ *for all* $k = 1, \ldots, n$*, and* $(g_1, g_2) \in Ext(\lambda_{\mathcal{R}}(id)) \Leftrightarrow g_1 = g_2$ *hold for all* $g_1, g_2 \in \mathcal{G}$*. The pair* $(\vec{\mathbb{K}}, \lambda)$ *is called* $\mathcal{A}$-structure *or* $\mathcal{A}$-model*.*

Similar to [3], we have to define valuations for CGwCs with variables, and based on valuation, we can define how CGwCs with variables are evaluated in models. The next two definitions have in [3] counterparts for CGwCs with generic markers, which are slightly modified to encompass the fact that we allow different variable vertices which are labeled with the same variable.

Definition 3 (Partial and Total Valuations for CGwCs with Variables). *Let* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa, \rho)$ *be a CGwC with variables and let* $\mathcal{M}$ *be a* $\mathcal{A}$*-structure. A mapping* $ref : V' \to G_0$ *with* $V^{\mathcal{G}} \subseteq V' \subseteq V$, $ref(v) = \lambda_{\mathcal{G}}(\rho(v))$ *for all* $v \in V^{\mathcal{G}}$ *, and* $ref(v_1) = ref(v_1)$ *for all* $v \in V^{Var}$ *is called a* partial valuation of $\mathfrak{G}$*. If we moreover have* $V' \supseteq \{v \in V^{Var} \mid scope(\rho(v)) > c\}$ *and* $V' \cap \{v \in V^{Var} \mid scope(\rho(v)) \leq c\} = \emptyset$*, we say that* ref *is a partial valuation* for the context c*. If* $V' = V$ *holds, then* ref *is called* (total) valuation of $\mathfrak{G}$*.*

Definition 4 (Endoporeutic Evaluation of Graphs).
Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa, \rho)$ *be a CGwC and variables and let* $\mathcal{M}$ *be a* $\mathcal{A}$*-structure. Inductively over the tree* $Cut \cup \{\top\}$*, we define* $(\vec{\mathbb{K}}, \lambda) \models \mathfrak{G}[c, ref]$ *for each context* $c \in Cut \cup \{\top\}$ *and every partial valuation* $ref : V' \subseteq V \to G_0$ *for* c*. We set* $(\vec{\mathbb{K}}, \lambda) \models \mathfrak{G}[c, ref] :\Longleftrightarrow$

> ref *can be extended to a partial valuation* $\widetilde{ref} : \widetilde{V'} \to G_0$ *with* $\widetilde{V'} := V' \cup \{v \in V^{Var} \mid scope(\rho(v)) = c\}$ *which satisfies:*
>
> - $\widetilde{ref}(v) \in Ext(\lambda_{\mathcal{C}}(\kappa(v)))$ *for each* $v \in V \cap area(c)$ *(vertex condition)*
> - $\widetilde{ref}(e) \in Ext(\lambda_{\mathcal{R}}(\kappa(e)))$ *for each* $e \in E \cap area(c)$ *(edge condition)*
> - $(\vec{\mathbb{K}}, \lambda) \not\models \mathfrak{G}[d, \widetilde{ref}]$ *for each* $d \in Cut \cap area(c)$ *(cut condition)*

For $(\vec{\mathbb{K}}, \lambda) \models \mathfrak{G}[\top, \emptyset]$ *we write* $(\vec{\mathbb{K}}, \lambda) \models \mathfrak{G}$*. If we have two concept graphs* $\mathfrak{G}_1$*,* $\mathfrak{G}_2$ *such that* $(\vec{\mathbb{K}}, \lambda) \models \mathfrak{G}_2$ *for each contextual structure* $(\vec{\mathbb{K}}, \lambda)$ *with* $(\vec{\mathbb{K}}, \lambda) \models \mathfrak{G}_1$*, we write* $\mathfrak{G}_1 \models \mathfrak{G}_2$*.*

We will consider CGwCs with variables only up to isomorphism and *renaming of the variables*. This idea is the well known *alpha-conversion* of formulas in linear and symbolic formalizations of FOL. Graphs which are identical up to different variable names will be called *equivalent*. This term is adopted from the Resource Description Framework, RDF).

Definition 5 (Equivalence of Graphs). *Let* $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa, \rho)$*,* $\mathfrak{G}' := (V', E', \nu', \top', Cut', area', \kappa', \rho')$ *be two CGwCs with variables. We will say that* $\mathfrak{G}$ *and* $\mathfrak{G}'$ *are* equivalent*, if* $\mathfrak{G}'$ *is isomorphic to a CGwC with variables* $\mathfrak{G}'' := (V, E, \nu, \top, Cut, area, \kappa, \rho'')$ *such that there is a bijective mapping* $f : Var \to Var$ *which satisfies* $\rho''(v) = \rho(v)$ *for each* $v \in V$ *with* $\rho(v) \in \mathcal{G}$ *and* $\rho(v'') = f(\rho(v))$ *for each* $v \in V$ *with* $\rho(v) \in Var$ *.*

Obviously, equivalent graphs have same meaning, i.e. if $\mathcal{M}$ is a model and $\mathfrak{G}, \mathfrak{G}'$ are two equivalent CGwCs with variables over $\mathcal{A}$, we have $\mathcal{M} \models \mathfrak{G} \Longleftrightarrow \mathcal{M} \models \mathfrak{G}'$.

In the forthcoming calculus, we could employ a rule which allows to transform a CGwC with variables into an equivalent graph. In this work, we use a more convenient approach: To ease the handling of CGwCs with variables, we agree that CGwCs with variables are considered only up to equivalence.

3 Purified CGwCs with Variables

In the following sections, we will deal with different kinds of CGwCs. Before we proceed, a simple notational convention shall be introduced: In the following, we will sometimes use upper indices to denote which kind of CGwCs we use. An upper index ^{g} denotes CGwCs with generic markers, and the upper indices ^{v} and ^{pv} denotes CGwCs with variables and purified CGwCs with variables, respectively. Moreover, we will use upper indices in brackets to denote mappings between these different classes of CGwCs.

We start with the canonical translation from purified CGwCs with variables to CGwCs with generic markers, which is given by replacing each variable by a generic marker. Vice versa, if a CGwCs with generic markers is given, we can replace each generic marker by a fresh variable. Of course, the assignment of variables to generic vertices is not uniquely given, but this poses no problem, as we consider CGwCs with variables only up to equivalence.

Definition 6 (Translations $^{(g)}$ and $^{(pv)}$). *Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa, \rho)$ be a purified CGwC with variables. Then let $\mathfrak{G}^{(g)} := (V, E, \nu, \top, Cut, area, \kappa, \rho^{(g)})$ the CGwC with generic markers with $\rho^{(g)}(v) = \rho(v)$, if $v \in \mathcal{G}$, and $\rho^{(g)}(v) = *$, if $v \in V^{Var}$. Vice versa, let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa, \rho)$ be a CGwC with generic markers, and let $f : V^* \to Var$ be an injective mapping from the set of generic nodes into the set of variables. Then let $\mathfrak{G}^{(pv)}$ be (the equivalence class of) the purified CGwC with variables $\mathfrak{G}^{pv} := (V, E, \nu, \top, Cut, area, \kappa, \rho^{(pv)})$ with $\rho^{(pv)}(v) = \rho(v)$, if $v \in \mathcal{G}$, and $\rho^{(pv)}(v) = f(v)$, if $v \in V^*$*

Obviously, the the mappings $^{(pv)}$ and $^{(g)}$ are mutually inverse bijections between purified CGwCs with variables and CGwCs with generic markers. Moreover, entailment is respected by $^{(pv)}$ and $^{(g)}$, i.e. if $\mathcal{M}$ be is model and if $\mathfrak{G}_v$ is a variable-purified CGwCs and $\mathfrak{G}_g$ is a CGwC with generic markers, we have

$$\mathcal{M} \models \mathfrak{G}_v \iff \mathcal{M} \models \mathfrak{G}_v^{(g)} \quad \text{and} \quad \mathcal{M} \models \mathfrak{G}_g \iff \mathcal{M} \models \mathfrak{G}_g^{(pv)} \tag{1}$$

These results justify the use of the term 'translation' for $^{(pv)}$ and $^{(g)}$.

Now we have to carry over the adequate calculus for CGwCs with generic markers to purified CGwCs with variables. The idea is straightforward: We will translate each rule for CGwCs with generic markers to a corresponding rule for purified CGwCs with variables. Let r be a rule of the calculus for generic CGwCs. We will write $\mathfrak{G}_a \vdash_r^g \mathfrak{G}_b$, if $\mathfrak{G}_a, \mathfrak{G}_b$ are two CGwCs with generic markers such that $\mathfrak{G}_b$ can be derived from $\mathfrak{G}_a$ by an application of the rule r. Now we will 'translate' each rule $\vdash_r^g$ to a rule $\vdash_r^{pv}$ for purified CGwCs with variables, i.e., the calculus $\vdash^{pv}$ will satisfy that

$$\mathfrak{G}_a \vdash_r^g \mathfrak{G}_b \iff \mathfrak{G}_a^{(pv)} \vdash_r^{pv} \mathfrak{G}_b^{(pv)} \tag{2}$$

holds for all CGwCs with generic markers $\mathfrak{G}_a$ and $\mathfrak{G}_b$. Note that vice versa, as $^{(pv)}$ and $^{(g)}$ are mutually inverse bijections, from (2) we obtain $\mathfrak{G}_a \vdash_r^{pv} \mathfrak{G}_b \iff \mathfrak{G}_a^{(g)} \vdash_r^g \mathfrak{G}_b^{(g)}$ for all CGwCs with variables $\mathfrak{G}_a$ and $\mathfrak{G}_b$ as well.

If the calculus $\vdash^{pv}$ is designed this way, it is complete. In order to see this, let $\mathfrak{G}_a$, $\mathfrak{G}_b$ be two purified CGwCs with variables. Then we have:

$$\mathfrak{G}_a \models \mathfrak{G}_b \overset{(1)}{\Leftrightarrow} \mathfrak{G}_a^{(g)} \models \mathfrak{G}_b^{(g)} \Leftrightarrow \mathfrak{G}_a^{(g)} \vdash^{g} \mathfrak{G}_b^{(g)} \overset{(2)}{\Leftrightarrow} \mathfrak{G}_a^{(g)(pv)} \vdash^{pv} \mathfrak{G}_b^{(g)(pv)} \Leftrightarrow \mathfrak{G}_a \vdash^{pv} \mathfrak{G}_b$$

Now we can provide the translations of the calculus for CGwCs with generic markers to purified CGwCs with variables such which satisfies Eqn. (2). The differences to the calculus for CGwCs with generic markers are indicated by writing the changed phrases in a different text style.

Definition 7 (Calculus for Purified CGwCs with Variables).
The calculus for purified CGwCs with variables over the alphabet $\mathcal{A} := (\mathcal{G}, \mathcal{C}, \mathcal{R})$ consists of the following rules:

- **erasure:** *In positive contexts, any directly enclosed edge, isolated vertex, and closed subgraph may be erased.*
- **insertion:** *In negative contexts, any directly enclosed edge, isolated vertex, and closed subgraph* whose variable vertices are labeled with fresh variables *may be inserted.*
- **iteration:** *Let $\mathfrak{G}_0 := (V_0, E_0, \nu_0, \top_0, Cut_0, area_0, \kappa_0, \rho_0)$ be a (not necessarily closed) subgraph of $\mathfrak{G}$ and let $c \leq cut(\mathfrak{G}_0)$ be a context such that $c \notin Cut_0$. Then a copy of $\mathfrak{G}_0$,* where each vertex $v = \boxed{P : \alpha}$ is replaced by $v = \boxed{P : \alpha'}$ for a fresh variable α', *may be inserted into c. For every vertex $v \in V_0^*$ with $cut(v) = cut(\mathfrak{G}_0)$, an identity-link from v to its copy may be inserted.*
- **deiteration:** *If $\mathfrak{G}_0$ is a subgraph of $\mathfrak{G}$ which could have been inserted by rule of iteration, then it may be erased.*
- **double cuts:** *Double cuts (two cuts c_1, c_2 with $area(c_1) = \{c_2\}$) may be inserted or erased.*
- **generalization:** *For evenly enclosed vertices and edges, their concept names resp. their relation names may be generalized.* Moreover, for evenly enclosed vertices which carry an object name as reference, the object name may be replaced by a fresh variable α.
- **specialization:** *For oddly enclosed vertices and edges, their concept names resp. their relation names may be specialized.* Moreover, for oddly enclosed vertices which carry a variable as reference, the variable may be replaced by an object name.
- **exchanging references:** *Let $e \in E^{id}$ be an identity link with $\rho(e\big|_1) = g_1$, $\rho(e\big|_2) = g_2$, $g_1, g_2 \in \mathcal{G} \cup Var$ and $cut(e) = cut(e\big|_1) = cut(e\big|_2)$. Then the references of v_1 and v_2 may be exchanged, i.e., the following may be done: We can set $\rho(e\big|_1) = g_2$ and $\rho(e\big|_2) = g_1$.*[2]

[2] Note that we allow to exchange two variable references as well, but applying the rule this way to a purified CGwCs with variables yields simply an equivalent graph, and equivalent graphs are already considered to be identical. Nonetheless, in the next section we will use this rule for not necessarily purified CGwCs with variables as well, and for these graphs, the rule has indeed an effect.

- **merging two vertices:** *Let* $e \in E^{id}$ *be an identity link with* $\nu(e) = (v_1, v_2)$ *such that*
  - $cut(v_1) \geq cut(e) = cut(v_2)$,
  - $\rho(v_1) = \rho(v_2) \in \mathcal{G}$ or $\rho(v_1), \rho(v_2) \in \mathrm{Var}$, *and*
  - $\kappa(v_2) = \top$

  hold. Then v_1 *may be merged into* v_2*, i.e.,* v_1 *and* e *are erased and, for every edge* $e \in E$, $e\big|_i = v_1$ *is replaced by* $e\big|_i = v_2$.
- **splitting a vertex:** *Let* $g \in \mathcal{G} \cup Var$*. Let* $v = \boxed{P : g}$ *be a vertex in the context* c_0 *and incident with relation edges* $R_1, \ldots, R_n$*, placed in contexts* $c_1, \ldots, c_n$*, respectively. Let* c *be a context such that* $c_1, \ldots, c_n \leq c \leq c_0$*. Then the following may be done: In* c*, a new vertex* $v' = \boxed{\top : g'}$, where $g' = g$, if $g \in \mathcal{G}$, or g' is a fresh variable, if $g \in \mathrm{Var}$, *and a new identity-link between* v *and* v' *is inserted. On* $R_1, \ldots, R_n$*, arbitrary occurrences of* v *are substituted by* v'.
- $\top$**-erasure:** *For* $g \in \mathcal{G} \cup Var$*, an isolated vertex* $\boxed{\top : g}$ *may be erased from arbitrary contexts.*
- $\top$**-insertion:** *For* $g \in \mathcal{G} \cup Var$*, an isolated vertex* $\boxed{\top : g}$ *may be inserted in arbitrary contexts.* Particularly, if $g \in \mathrm{Var}$, g has to be a fresh variable in order to obtain a well-formed purified CGwC with variables.
- **identity-erasure:** *Let* $g \in \mathcal{G}$*, let* $v_1 = \boxed{P_1 : g}$ *and* $v_2 = \boxed{P_2 : g}$ *be two vertices. Then any identity-link between* v_1 *and* v_2 *may be erased.*
- **identity-insertion:** *Let* $g \in \mathcal{G}$*, let* $v_1 = \boxed{P_1 : g}$, $v_2 = \boxed{P_2 : g}$ *be two vertices in contexts* c_1*,* c_2*, resp. and let* $c \leq c_1, c_2$ *be a context. Then an identity-link between* v_1 *and* v_2 *may be inserted into* c*.*

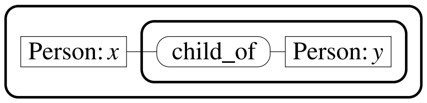
4 CGwCs with Variables

In this section, we will extend the calculus $\vdash^{pv}$ of Sec. 3 to the system of (not necessarily purified) CGwCs with variables.

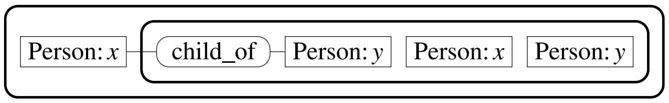
The calculus of Sec. 3 so far is defined only for purified CGwCs with variables (particularly, it can only be applied to these graphs). Moreover, we had to design the rules of $\vdash^{pv}$ to make sure that an application of any rule to a purified CGwCs with variables yields a purified CGwCs with variables again. For example, the rule 'insertion' of $\vdash^{pv}$ can only be applied to a purified CGwC with variables, and we only allowed to insert subgraphs where all variable boxes are labeled with a *fresh* variable. As we now consider non-purified CGwCs with variables as well, this is now an unnecessary restriction.

It is reasonable not to add more rules to $\vdash^{pv}$ in order to obtain a sound and complete calculus $\vdash^{v}$ for CGwCs with variables. Instead, we will extend each rule of $\vdash^{pv}$ to rule for CGwCs with variables, and if the rule had some restrictions which were needed to ensure that an application of the rule yields a purified CGwC with variables, these restrictions are now dismissed. On the other hand, when we extend a rule $\vdash^{pv}_{r}$ to a rule $\vdash^{v}_{r}$ for not necessarily purified CGwCs with variables, we have to take care that when $\vdash^{v}_{r}$ is applied, no scopes of variables are allowed to change. We exemplify this necessity with two examples.

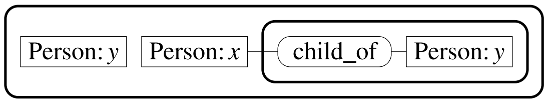
We start with an example of extending the rule '$\top$-insertion'. Consider the following valid graph with the meaning 'every person is the child of a person'.



For CGwCs with variables, it is self-suggesting that we now allow to insert $\top$-boxes into arbitrary contexts. Let us first consider an insertion of $\top$-boxes such that no scopes of variables are changed, for example like this:

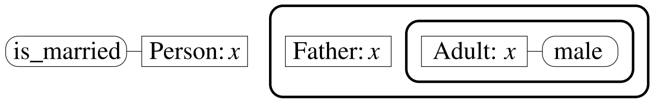


This is indeed a valid derivation. But if we insert a new box $\boxed{\top : x}$ which changes the scope of the variable x, like in the next graph,

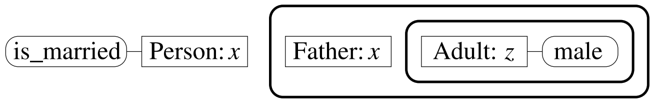


we obtain an invalid graph with the meaning 'every person is the child of every person'. Thus, for CGwCs with variables, we can insert a new box $\boxed{\top : \alpha}$ with $\alpha \in \text{Var}$ only if this insertion does not change the scope of α.

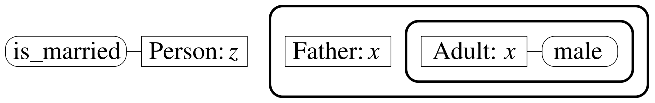
In the rule 'generalization', we will not only allow to generalize an object name to a fresh variable, but we will allow to generalize a variable to a fresh variable as well. Again, we have to take care that no scope of a variable changes when this rule is applied. To see this, consider the following valid graph with the meaning 'there exists a married person, and if this person is a father, it is a male adult' (we assume that only adults are allowed to marry).



We can generalize the variable x of the innermost concept box to the fresh variable z. Then we obtain the following graph:



The meaning of this graph is 'there exists a married person, and if this person is a father, there exists a is a male adult'. Obviously, this derivation is valid. But if we generalize the variable x of the outermost concept box to the fresh variable z, we obtain the following graph where the scope of x has changed:



The meaning of this graph is 'there exists a married person, and every father is a male adult', which is, as there are fathers who are still a minor, not true.

The calculus $\vdash^{pv}$ had been designed to make sure that an application of any rule to a purified CGwCs with variables yields a purified CGwCs with variables again. This restriction can be dismissed now, but recall that we only consider CGwCs with variables which have dominating variable boxes. This has to be taken into account when the rules of $\vdash^{v}$ are introduced. To summarize: If we reformulate a rule $\vdash^{pv_r}$ to a rule $\vdash^{v}_{r}$, we have to make sure that no scope of any variable is changed, and applying a rule yields always a CGwC with variables having dominating variable boxes.

Now we are prepared to provide the calculus for CGwCs with variables. Most of the rules are direct extensions of the rules for purified CGwCs with variables, but there are two significant changes. First of all, in the generalization rule, we allow to replace the variable of a variable vertex by a fresh variable. Secondly, in the rule 'splitting a vertex', if the reference of the vertex is a variable, we now allow that the new copy is labeled with the same variable as well. The specialization rule and the rule 'merging two vertices' are extended in a way that they allows the reverse transformation in oddly enclosed contexts.

Definition 8 (Calculus for CGwCs with Variables).

The calculus for variable-purified CGwCs over the alphabet $\mathcal{A} := (\mathcal{G}, \mathcal{C}, \mathcal{R})$ consists of the following rules:

- **erasure:** *In positive contexts, any directly enclosed edge, isolated vertex, and closed subgraph $\mathfrak{G}'$,* where each variable box of $\mathfrak{G}'$ is dominated by a variable box which does not belong to $\mathfrak{G}'$*, may be erased.*
- **insertion:** *In negative contexts, any directly enclosed edge, isolated vertex, and closed subgraph $\mathfrak{G}'$,* where each variable box of $\mathfrak{G}'$ is dominated by a variable box which does not belong to $\mathfrak{G}'$*, may be inserted.*
- **iteration:** *Let $\mathfrak{G}_0 := (V_0, E_0, \nu_0, \top_0, Cut_0, area_0, \kappa_0, \rho_0)$ be a (not necessarily closed) subgraph of $\mathfrak{G}$ and let $c \leq cut(\mathfrak{G}_0)$ be a context such that $c \notin Cut_0$. Then a copy of $\mathfrak{G}_0$,* where each vertex $v = \boxed{P : \alpha}$ is replaced by $v = \boxed{P : \alpha'}$ for a fresh variable α'*, may be inserted into c. For every vertex $v \in V_0^*$ with $cut(v) = cut(\mathfrak{G}_0)$, an identity-link from v to its copy may be inserted.*
- **deiteration:** *If $\mathfrak{G}_0$ is a subgraph of $\mathfrak{G}$ which could have been inserted by rule of iteration, then it may be erased.*
- **double cuts:** *Double cuts (two cuts c_1, c_2 with $area(c_1) = \{c_2\}$) may be inserted or erased.*
- **generalization:** *For evenly enclosed vertices and edges, their concept names resp. their relation names may be generalized.* Moreover, for each evenly enclosed vertex, its reference may be replaced by a fresh variable α.
- **specialization:** *For vertices v and edges e in the area of an odd cut c, their concept names resp. their relation names may be specialized.* Moreover, if v is a singular variable vertex, this variable may be replaced by an object name or another variable α, provided we have $scope(\alpha) \geq c$ in $\mathfrak{G}$.

– **exchanging references:** *Let $e \in E^{id}$ be an identity link with $\rho(e\big|_1) = g_1$, $\rho(e\big|_2) = g_2$, $g_1, g_2 \in \mathcal{G} \cup Var$ and $cut(e) = cut(e\big|_1) = cut(e\big|_2)$. Then the references of v_1 and v_2 may be exchanged, i.e., the following may be done: We can set $\rho(e\big|_1) = g_2$ and $\rho(e\big|_2) = g_1$.*[3]
– **merging two vertices:** *Let $e \in E^{id}$ be an identity link with $\nu(e) = (v_1, v_2)$ such that*
  - *$cut(v_1) \geq cut(e) = cut(v_2)$,*
  - *$\rho(v_1) = \rho(v_2) \in \mathcal{G}$*, or $\rho(v_1), \rho(v_2) \in$ Var such that $\rho(v_1) = \rho(v_2)$ or v_2 is a singular variable vertex, *and*
  - *$\kappa(v_2) = \top$*

  hold. Then v_1 may be merged into v_2, i.e., v_1 and e are erased and, for every edge $e \in E$, $e\big|_i = v_1$ is replaced by $e\big|_i = v_2$.
– **splitting a vertex:** *Let $g \in \mathcal{G} \cup Var$. Let $v = \boxed{P : g}$ be a vertex in the context c_0 and incident with relation edges $R_1, \ldots, R_n$, placed in contexts $c_1, \ldots, c_n$, respectively. Let c be a context such that $c_1, \ldots, c_n \leq c \leq c_0$. Then the following may be done: In c, a new vertex $v' = \boxed{\top : g'}$*, where $g' = g$, if $g \in \mathcal{G}$, and $g' = g$ or g' is a fresh variable, if $g \in$ Var, *and a new identity-link between v and v' is inserted. On $R_1, \ldots, R_n$, arbitrary occurrences of v are substituted by v'.*
– **$\top$-erasure:** *For $g \in \mathcal{G}$, an isolated vertex $\boxed{\top : g}$ may be erased from arbitrary contexts. For $\alpha \in Var$, an dominated isolated vertex $\boxed{\top : \alpha}$ may be erased from arbitrary contexts.*
– **$\top$-insertion:** *Let c be context. For $g \in \mathcal{G} \cup Var$, an isolated vertex $\boxed{\top : g}$ may be inserted into $area(c)$.* For $\alpha \in$ Var with *$scope(\alpha) \geq c$*, an isolated vertex $\boxed{\top : \alpha}$ may be inserted into *$area(c)$*.
– **identity-erasure:** *Let $g \in \mathcal{G} \cup Var$, let $v_1 = \boxed{P_1 : g}$ and $v_2 = \boxed{P_2 : g}$ be two vertices. Then any identity-link between v_1 and v_2 may be erased.*
– **identity-insertion:** *Let $g \in \mathcal{G} \cup Var$, let $v_1 = \boxed{P_1 : g}$, $v_2 = \boxed{P_2 : g}$ be two vertices in contexts c_1, c_2, resp. and let $c \leq c_1, c_2$ be a context. Then an identity-link between v_1 and v_2 may be inserted into c.*

in contrast to purified CGwCs with variables, as we extended the class of well-formed graphs, the soundness of these rules is not immediately clear. Nonetheless, to each rule of the calculus for CGwCs with variables corresponds a rule for CGwCs with generic markers. For each rule of the calculus for CGwCs with generic markers, a soundness-proof is provided in [3]. The underlying ideas for the rules are in both systems identical, and a closer observation of the proofs of [3] shows that they can be rewritten for the system of CGwCs with variables. Thus, the following lemma is given without a proof.

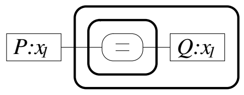
Lemma 1 (Soundness of $\vdash^v$). *The rules of $\vdash^v$ are sound, i.e., for two CGwCs with variables $\mathfrak{G}^v_a, \mathfrak{G}^v_b$, we have $\mathfrak{G}^v_a \vdash^v \mathfrak{G}^v_b \Longrightarrow \mathfrak{G}^v_a \models \mathfrak{G}^v_b$.*

[3] Note that we allow to exchange two variable references as well, which has now, in contrast to purified CGwCs with variables has an effect.

Now we have to show that the rules of $\vdash^v$ are complete. We start with a lemma where we show that each CGwCs with variables can be transformed to a purified CGwCs with variables, and vice versa, with the rules of $\vdash^v$.

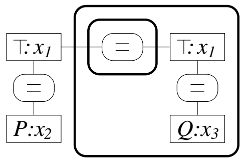
Lemma 2. *For each CGwC with variables $\mathfrak{G}^v$ exists an syntactically equivalent purified CGwC with variables $\mathfrak{G}^{pv}$, i.e., we have $\mathfrak{G}^v \vdash^v \mathfrak{G}^{pv}$ and $\mathfrak{G}^{pv} \vdash^v \mathfrak{G}^v$.*

Proof: We will transform $\mathfrak{G}^v$ into a purified CGwC with variables $\mathfrak{G}^{pv}$. The procedure of the proof shall be exemplified with the graph on the right.

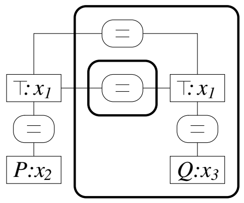


For each variable vertex v, we do the following: First, v is split such that the copy v' of v is labeled with a fresh variable, and *all* occurrences of v on an edge are replaced by v'. Then, the references of v and v' are exchanged.

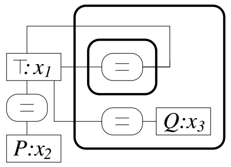
After this, each variable box which is not a singular variable box is labeled with the concept name $\top$.



Then, for each variable α, we choose an dominating variable box v_α insert an identity-link between v_α and all remaining vertices labeled with α.



Finally, for each variable α, each variable box $w \neq v_\alpha$ labeled with α is merged into v_α.



The resulting graph $\mathfrak{G}^{pv}$ is purified. As each step in the proof can be carried out in both directions, it is provably equivalent to $\mathfrak{G}^v$, thus we are done. □

Each rule $\vdash^v_r$ of $\vdash^v$ is an extension of the rule $\vdash^{pv}_r$. Thus, $\vdash^v$ is a complete calculus for purified CGwCs. Together with the last lemma, we immediately obtain the completeness of $\vdash^v$, i.e., we get:

Corollary 1 (Completeness of $\vdash^v$). *The rules of $\vdash^v$ are complete, i.e., for two CGwCs with variables $\mathfrak{G}_a, \mathfrak{G}_b$, we have $\mathfrak{G}_a \models \mathfrak{G}_b \Longrightarrow \mathfrak{G}_a \vdash^v \mathfrak{G}_b$.*

5 Conclusion

In this paper, we investigated how the results of [3] for CGwCs with generic markers can be transferred to CGwCs with variables. At a first glance, the difference between these two systems is a minor syntactical difference. But a closer observation shows that one has to take care of several technical details, mostly in the formalization of the transformation rules for CGwCs with variables. Particularly, we had to take care that an application of a transformation rule to a

CGwCs yields a well-formed graph again, and we had to ensure that no application of an transformation rule changes the scope of a variable. These restrictions result in transformation rules CGwCs with variables which are technically more complex than their counterparts for CGwCs with generic markers.

This shows on the one hand that it is necessary to investigate CGwCs with variables on its own. On the other hand, this work shows that one main goal of conceptual graphs, namely that humans can better handle them than any symbolic notation for logic, is better achieved if generic markers instead of variables are used for existential quantification. For this reason, this conclusion advocates the use of generic markers.

References

[1] J. F. Baget: *Homomorphismes d'hypergraphes pour la subsumption en RDF/RDFS.* RSTI L'objet, LMO 04, 2004, p. 203–216.

[2] M.-L. Mugnier: *Concept Types and Coreference in Simple Conceptual Graphs.* In: Pfeiffer, H. D.; Wolff, K. E. (Eds): Conceptual Structures at Work, LNAI, Vol. 3127, Springer-Verlag, Berlin – Heidelberg – New York, 2004, p 303–318.

[3] F. Dau: *The Logic System of Concept Graphs with Negation (And Its Relationship to Predicate Logic).* LNAI, Vol. 2892, Springer, Berlin–Heidelberg–New York, 2003.

[4] F. Dau: *RDF as Graph-Based Diagrammatic Reasoning System: Syntax, Semantics, Calculus* Submitted to the 2nd European Semantic Web Conference.

[5] F. Dau: *Rhetorical Structures in Diagrammatic Reasoning Systems.* Submitted to the symposium on Visual Languages and Human Centric Computing 05.

[6] P. Hayes: *RDF Semantics: W3C Recommendation.* 10 February 2004. http://www.w3.org/TR/rdf-mt/

[7] J. Klinger: *Semiconcept Graphs with Variables.* In: U. Priss, D. Corbett, and G. Angelova (Eds.): Conceptual Structures: Integration and Interfaces. LNAI 2393. Springer-Verlag, Heidelberg-Berlin, 2002.

[8] J. Klinger: *The Logic System of Protoconcept Graphs.* PhD-thesis. To appear.

[9] F. Manola, E. Miller: *RDF Primer.* http://www.w3.org/TR/rdf-primer/

[10] C. S. Peirce: Collected Papers. Harvard University Press, Cambridge, Massachusetts, 1931–1935.

[11] J. F. Sowa: *Conceptual Structures: Information Processing in Mind and Machine.* The System Programming Series. Adison-Wesley, Reading 1984.

[12] J. F. Sowa: *Conceptual Graphs Summary.* in: T. E. Nagle, J. A. Nagle, L. L. Gerholz, P. W. Eklund (Eds.): Conceptual Structures: current research and practice, Ellis Horwood, 1992, 3–51.

[13] J. F. Sowa: Logic: Graphical and Algebraic, Manuscript, Croton-on-Hudson 1997.

[14] R. Wille: *Existential Concept Graphs of Power Context Families.* In: U. Priss, D. Corbett and G. Angelova (Eds.): Conceptual Structures: Integration and Interfaces. LNAI 2393, Springer Verlag, Berlin–New York, 2002.

[15] R. Wille: *Implicational Concept Graphs.* In: H. D.. Pfeiffer, K. E. Wolff (Eds.): Conceptual Struchtures at Work. LNAI 3127, Springer Verlag, Berlin–New York, 2004.

6 Appendix: The Calculi for Both Systems

In this appendix, a short overview on the handling of variables in for the calculi $\vdash^{pv}$ for purified CGwCs with variables and $\vdash^{v}$ for CGwCs with variables is provided.

	purified CGwCs with variables	CGwCs with variables
erasure insertion	Each closed subgraph (which can be an isolated vertex) may be erased. Arbitrary closed subgraphs (which can be isolated vertices), if they contain only fresh variables, may be inserted.	Only subgraphs $\mathfrak{G}'$ (which can be a single isolated vertex, where all variable boxes of $\mathfrak{G}'$ are dominated by other variable boxes which do not belong to $\mathfrak{G}'$, subgraph may be erased or inserted.
iteration deiteration	If a subgraph $\mathfrak{G}'$ is iterated, then in the copy of $\mathfrak{G}'$, each variable has to be replaced by a fresh variable.	If a subgraph $\mathfrak{G}'$ is iterated, then in the copy of $\mathfrak{G}'$, each variable has to be replaced by a fresh variable. Only subgraphs with singular variable boxes can be deiterated.
double cuts	no problems	no problems
general. special.	For the rule 'generalization', object names can be replaced by fresh variables. (Note that generalizing variable boxes to variable boxes with a fresh variable has no effect). Vice versa, for the rule 'specialization', variables can be specialized to object names.	For the rule 'generalization', object names can be generalized to fresh variables. If a variable box is dominated, its variable can be generalized to a fresh variable. Vice versa, for the rule 'specialization', variables in singular variable boxes can be specialized to object names. Moreover, variables in singular variable boxes can be replaced by a variable α, if the box is -after the transformation- dominated by another variable box.
exchanging references	No restrictions, but due to the equivalence of graphs, exchanging the references of two variable boxes has no effect.	No restrictions. Note that exchanging the references of two variable boxes now has an effect.
split/merge vertices	If a variable box is split, the new box has to be labelled with fresh variable.	If a variable box is split, the new box has to be labelled with same or fresh variable.
$\top$-erasure $\top$-insertion	An isolated $\top$-box can be erased. An isolated $\top$-box which is labelled with an object name or a fresh variable can be inserted.	An isolated $\top$-box, if it is labelled with an object name, or if is a variable box which is dominated by another variable box, of if it is a a singular variable box, can be erased or inserted.
id-era. id-ins.	id-erasure/id-insertion is possible only between two object boxes with the same references.	id-erasure/id-insertion is possible between two object boxes or two variable boxes with the same references.