Functorial Question Answering

Giovanni de Felice[†], Konstantinos Meichanetzidis[†⋆], Alexis Toumi[†]

† Department of Computer Science, University of Oxford. ⋆ Cambridge Quantum Computing Ltd.

July 8, 2019

We study the relational variant of the categorical compositional distributional (DisCoCat) models of Coecke et al. [1], where we replace vector spaces and linear maps by sets and relations. We show that RelCoCat models factorise through Cartesian bicategories, as a corollary we get logspace reductions from semantics and entailment to evaluation and containment of conjunctive queries respectively. Finally, we define question answering as an $\texttt{NP}-\texttt{complete}$ problem.

Introduction

In this paper, we give a semantics to pregroup grammars in *regular logic*: the fragment of first-order logic generated by relational symbols, equality $(=)$, truth $(\top)$, conjunction $(\wedge)$ and existential quantification $(\exists)$. Regular logic plays a foundational role in the theory of relational databases, where it corresponds to *conjunctive queries*. Chandra and Merlin [2] showed that conjunctive query evaluation and containment are logspace equivalent to graph homomorphism: they are $\texttt{NP}-\texttt{complete}$. Bonchi et al. [3] reformulated this in terms of the free Cartesian bicategory $\mathbf{CB}(\Sigma)$: arrows are conjunctive queries, structure-preserving functors $K : \mathbf{CB}(\Sigma) \to \mathbf{Rel}$ are precisely relational databases.

We define *concrete* RelCoCat (categorical compositional relational) models as strong monoidal functors $F : \mathbf{G} \to \mathbf{Rel}$ from the rigid monoidal category $\mathbf{G}$ generated by a pregroup grammar. We show that RelCoCat models factorise as $F = K \circ L$ for $K : \mathbf{CB}(\Sigma) \to \mathbf{Rel}$ a relational database and a functor $L : \mathbf{G} \to \mathbf{CB}(\Sigma)$, which we call an *abstract* RelCoCat model. As a corollary, the computational problems for RelCoCat $\texttt{Semantics}$ and $\texttt{Entailment}$ reduce to conjunctive query evaluation and containment respectively. Building on previous work [4], we then show that the $\texttt{QuestionAnswering}$ problem is $\texttt{NP}-\texttt{complete}$.

Logical semantics for pregroups has been developped in a line of work by Preller [5, 6, 7], however the corresponding reasoning problems were undecidable. Entailment in distributional models has been considered in [8], but sentences could only be compared if they have the same grammatical structure.

1 Lambek Pregroups and Free Rigid Categories

Pregroup grammar is an algebraic model of natural language grammar introduced by Lambek [9], it is weakly equivalent to context-free grammars [10]. In this section, we give a definition of the `Parsing` problem in terms of the homsets of the free *rigid monoidal category* generated by a pregroup dictionary.

Given a natural number $n \in \mathbb{N}$, we abuse notation and let $n = \{ i \in \mathbb{N} \mid i < n \}$. Given sets X and Y, $X + Y$ and $X \times Y$ denote the disjoint sum and the Cartesian product respectively. Let $List(X) = \bigcup_{n \in \mathbb{N}} X^n$ be the free monoid with unit $\epsilon \in X^0$ the empty list and product denoted by concatenation.

Definition 1.1. A pregroup is a strict monoidal category with unit ϵ and product denoted by concatenation, which is thin — i.e. with at most one arrow (denoted $\leq$) between any two objects — and where each object t has left and right adjoints denoted ${}^\star t$ and $t^\star$, i.e.

- $t({}^\star t) \quad \leq \quad \epsilon \quad \leq \quad ({}^\star t)t$ (left adjunction)
- $(t^\star)t \quad \leq \quad \epsilon \quad \leq \quad t(t^\star)$ (right adjunction)

A pregroup grammar is a tuple $G = (V, B, \Delta)$ where V is a finite set called the *vocabulary*, B is a poset of *basic types* and $\Delta \subseteq V \times P(B)$ is a finite set of pairs called the *dictionary*, where $P(B)$ is the free pregroup generated by B as defined in [9]. We write $\Delta(u) = \{ t(0) \dots t(n-1) \mid t \in \prod_{i<n} \Delta(u(i)) \}$ for $u \in V^n$.

Definition 1.2. `Grammaticality`

Input: $G = (V, B, \Delta), \quad u \in List(V), \quad s \in P(B)$
Output: $\exists\, t \in \Delta(u) \ \cdot \ t \leq s$

Example 1.3. In what follows we will take the basic types $B = \{ s, q, d, n, i, o \}$ for sentence, question, determinant, noun, subject and object respectively, with $n \leq i$ and $n \leq o$. The dictionary Δ assigns $({}^\star i)s(o^\star)$ to transitive verbs and $({}^\star d)n$ to common nouns. The word "who" is assigned both $({}^\star n)n(s^\star)i$ and $q(s^\star)i$. From the pregroup axioms it follows that:

$$q(s^\star)i \ \ ({}^\star i)s(o^\star) \ \ d \ \ ({}^\star d)n \ \ ({}^\star n)n(s^\star)i \ \ ({}^\star i)s(o^\star) \ \ n \quad \leq \quad q$$

i.e. $u =$ "Who influenced the philosopher who discovered calculus?" is grammatical.

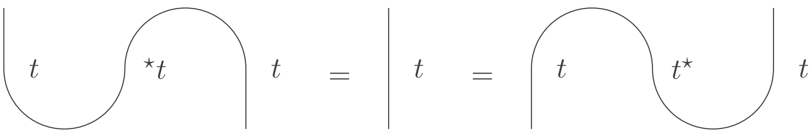
Lemma 1.4 (Switching lemma [9])**.** For all $t \leq s \in P(B)$ there is some $t' \in P(B)$ and a pair of reductions $t \leq t'$ and $t' \leq s$ with no expansions and no contractions respectively.

Corollary 1.5 ([10])**.** $\texttt{Grammaticality} \in \texttt{P}$

Proof. The proof goes by translating pregroup grammars to context-free grammars. □

As thin categories, pregroups cannot distinguish between distinct parsings of the same phrase, e.g. "men and (women who read)" and "(men and women) who read". This motivated Preller, Lambek [11] to introduce free compact 2-categories, capturing proof-relevance in pregroup grammars. We will use compact 2-categories with one 0-cell, first introduced in Joyal, Street [12] from which we use the *planar string diagram* notation. We refer the reader to Selinger's survey [13] where they are called *rigid monoidal categories*.

Definition 1.6. A (strict) monoidal category $(\mathbf{C}, \otimes, \epsilon)$ is rigid when each object $t \in Ob(\mathbf{C})$ has left and right adjoints ${}^{\star}t$ and $t^{\star}$ and two pairs of arrows $t \otimes {}^{\star}t \to \epsilon \to {}^{\star}t \otimes t$ and $t^{\star} \otimes t \to \epsilon \to t \otimes t^{\star}$ depicted by cups and caps, subject to the following snake equations:



We say a strong monoidal functor is rigid when it sends cups to cups and caps to caps.

Given a pregroup grammar $G = (V, B, \Delta)$, the dictionary $\Delta \subseteq V \times P(B)$ defines an *autonomous signature* with generating objects $V + B$ and arrows $\{w \to t\}_{(w,t) \in \Delta}$. We write $\mathbf{G}$ for the free rigid category that it generates, also called the *lexical category* in [6]. An arrow $r : u \to s$ for an utterance $u \in List(V)$ is a proof that u is a grammatical sentence, i.e. a dictionnary entry for each word followed by a diagram which encodes the reduction. Hence, the free rigid category $\mathbf{G}$ allows us to encode parsing as a function problem.

Lemma 1.7. Given a pregroup grammar $G = (V, B, \Delta)$, an utterance $u \in List(V)$ and a type $s \in P(B)$, we have $(G, u, s) \in \texttt{Grammaticality} \iff \exists\, r \in \mathbf{G}(u, s)$.

Proof. This follows from the switching lemma for compact 2-categories as proved in [11]. Arrows $r \in \mathbf{G}(u, s)$ are of the form $r : u \to t \to s$ for some type $t \in \Delta(u)$ with $t \leq s$. □

Definition 1.8. `Parsing`

Input: $G = (V, B, \Delta), \quad u \in List(V), \quad s \in P(B)$

Output: $r \in \mathbf{G}(u, s)$

Proposition 1.9 ([14])**.** `Parsing` is poly-time computable in the size of the basic types B, the dictionary $\Delta \subseteq V \times P(B)$ and the length of the inputs $(u, s) \in List(V) \times P(B)$.

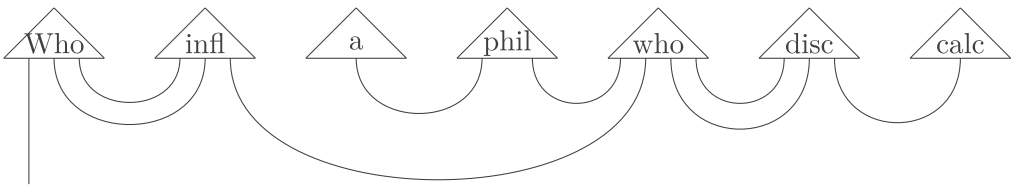
Proof. The parsing algorithm has time complexity n^3 in general, restricted cases of interest in linguistic applications may be parsed in linear time, see [15]. □

Note that our definition of the lexical category $\mathbf{G}$ differs slightly from [6] in that we take not only basic types $b \in B$ but also words $w \in V$ as generating objects. This allows us to capture both type assignment and reduction as a single arrow as well as to define the semantics of pregroup grammars as a functor, see definition 3.2 where we will also make use of the following lemma.

Lemma 1.10. For any pregroup grammar $G = (V, B, \Delta)$ there is an equivalent grammar $G' = (V, B', \Delta')$ such that $\Delta \subseteq \Delta'$ and B' is a discrete poset, i.e. $a \leq b \implies a = b$.

Proof. For each $(w, \; tat') \in \Delta$ with $a \leq b \in B$ we add $(w, \; tbt')$ as a dictionnary entry. This yields a dictionnary Δ' of size polynomial in $|B| \times |\Delta|$ and basic types B' given by the underlying set of the poset B such that G and $G' = (V, B', \Delta')$ are equivalent. □

Example 1.11. The following planar diagram $r : u \to q$ corresponds to the parsing of example 1.3, where we have we have omitted the types for readability. We keep our notation consistent with the literature by depicting dictionnary entries $w \to t$ as triangles labeled by $w \in V$ with output $t \in P(B)$.



2 Conjunctive Queries and Free Cartesian Bicategories

A *relational signature* is a set of symbols Σ equipped with a function $\mathtt{ar} : \Sigma \to \mathbb{N}$. Given a finite set U, we define the set of Σ-*models* $\mathcal{M}_\Sigma(U) = \{ K \subseteq \coprod_{R \in \Sigma} U^{\mathtt{ar}(R)} \}$, i.e. a Σ-model K gives an interpretation $K(R) \subseteq U^{\mathtt{ar}(R)}$ for every symbol $R \in \Sigma$. Let $\mathcal{M}_\Sigma$ be the set of all finite Σ-models and $U(K)$ the underlying universe of

$K \in \mathcal{M}_\Sigma$. Given two Σ-models K, K', a homomorphism $f : K \to K'$ is a function $f : U(K) \to U(K')$ such that $\forall\, R \in \Sigma\ \forall\, \vec{x} \in U^{\mathtt{ar}(R)} \ \cdot\ \vec{x} \in K(R) \implies f(\vec{x}) \in K'(R)$.

Definition 2.1. `Homomorphism`

> ***Input:*** $K, K' \in \mathcal{M}_\Sigma$
>
> ***Output:*** $f : K \to K'$

Proposition 2.2. [16] `Homomorphism` is $\mathtt{NP-complete}$.

Proof. Membership may be shown to follow from Fagin's theorem: homomorphisms are defined by an existential second-order logic formula. Hardness follows by reduction from graph homomorphism: take $\Sigma = \{\bullet\}$ and $\mathtt{ar}(\bullet) = 2$ then a Σ-model is a graph. □

Regular logic formulae are defined by the following context-free grammar:

$$\varphi \quad ::= \quad \top \ \mid\ x = x' \ \mid\ \varphi \wedge \varphi \ \mid\ \exists\, x \cdot \varphi \ \mid\ R(\vec{x})$$

where $x, x' \in \mathcal{X}$, $R \in \Sigma$ and $\vec{x} \in \mathcal{X}^{\mathtt{ar}(R)}$ for some countable set of variables $\mathcal{X}$. We denote the variables of φ by $\mathtt{var}(\varphi) \subseteq \mathcal{X}$, its free variables by $\mathtt{fv}(\varphi) \subseteq \mathtt{var}(\varphi)$ and its atomic formulae by $\mathtt{atoms}(\varphi) \subseteq \coprod_{R\in\Sigma} \mathrm{var}(\varphi)^{\mathtt{ar}(R)}$. *Conjunctive queries* $\varphi \in \mathcal{Q}_\Sigma$ are the prenex normal form $\varphi = \exists\, x_0 \cdots \exists\, x_k \cdot \varphi'$ of regular logic formulae, for the bound variables $\{x_0, \ldots, x_k\} = \mathtt{var}(\varphi) \setminus \mathtt{fv}(\varphi)$ and $\varphi' = \bigwedge \mathtt{atoms}(\varphi)$. Given a model $K \in \mathcal{M}_\Sigma$, let $\mathtt{eval}(\varphi, K) = \{ v \in U(K)^{\mathtt{fv}(\varphi)} \mid (K, v) \vDash \varphi \}$ where the satisfaction relation $(\vDash)$ is defined in the usual way.

Definition 2.3. `Evaluation`

> ***Input:*** $\varphi \in \mathcal{Q}_\Sigma, \quad K \in \mathcal{M}_\Sigma$
>
> ***Output:*** $\mathtt{eval}(\varphi, K) \subseteq U(K)^{\mathtt{fv}(\varphi)}$

Definition 2.4. `Containment`

> ***Input:*** $\varphi, \varphi' \in \mathcal{Q}_\Sigma$
>
> ***Output:*** $\varphi \subseteq \varphi' \ \equiv\ \forall\, K \in \mathcal{M}_\Sigma \ \cdot\ \mathtt{eval}(\varphi, K) \subseteq \mathtt{eval}(\varphi', K)$

Definition 2.5. Given a query $\varphi \in \mathcal{Q}_\Sigma$, the canonical model $CM(\varphi) \in \mathcal{M}_\Sigma$ is given by $U(CM(\varphi)) = \mathtt{var}(\varphi)$ and $CM(\varphi)(R) = \{ \vec{x} \in \mathtt{var}(\varphi)^{\mathtt{ar}(R)} \mid R(\vec{x}) \in \mathtt{atoms}(\varphi) \}$ for $R \in \Sigma$.

Theorem 2.6 (Chandra-Merlin [2])**.** `Evaluation` and `Containment` are logspace equivalent to `Homomorphism`, hence $\mathtt{NP-complete}$.

Proof. Given a query $\varphi \in \mathcal{Q}_\Sigma$ and a model $K \in \mathcal{M}_\Sigma$, query evaluation $\mathtt{eval}(\varphi, K)$ is given by the set of homomorphisms $CM(\varphi) \to K$. Given $\varphi, \varphi' \in \mathcal{M}_\Sigma$, we have $\varphi \subseteq \varphi'$ iff there is a homomorphism $f : CM(\varphi) \to CM(\varphi')$ such that $f(\mathtt{fv}(\varphi)) = \mathtt{fv}(\varphi')$. Given a model $K \in \mathcal{M}_\Sigma$, we construct $\varphi \in \mathcal{Q}_\Sigma$ with $\mathtt{fv}(\varphi) = \varnothing$, $\mathtt{var}(\varphi) = U(K)$ and $\mathtt{atoms}(\varphi) = K$. □

Bonchi, Seeber and Sobocinski [3] introduced graphical conjunctive queries (GCQ), a graphical calculus where query containment is captured by the axioms of the *free Cartesian bicategory* $\mathbf{CB}(\Sigma)$ generated by the relational signature Σ.

Definition 2.7 (Carboni-Walters [17]). A Cartesian bicategory is a symmetric monoidal category enriched in partial orders such that:

1. every object is equipped with a special commutative Frobenius algebra,
2. the monoid and comonoid structure of each Frobenius algebra are adjoint,
3. every arrow is a lax comonoid homomorphism.

A morphism of Cartesian bicategories is a strong monoidal functor which preserves the partial order, the monoid and the comonoid structure.

Theorem 2.8. ([3, prop. 9,10]) Let $\mathbf{CB}(\Sigma)$ be the free Cartesian bicategory generated by one object and arrows $\{ R : 0 \to \mathtt{ar}(R) \}_{R \in \Sigma}$, see [3, def. 21]. There is a two-way semantics-preserving translation $\Theta : \mathcal{Q}_\Sigma \to \mathbf{CB}(\Sigma)$, $\Lambda : \mathbf{CB}(\Sigma) \to \mathcal{Q}_\Sigma$, i.e. for all $\varphi, \varphi' \in \mathcal{Q}_\Sigma$ we have $\varphi \subseteq \varphi' \iff \Theta(\varphi) \leq \Theta(\varphi')$, and for all arrows $d, d' \in \mathbf{CB}(\Sigma)$, $d \leq d' \iff \Lambda(d) \subseteq \Lambda(d')$.

Proof. The translation is defined by induction from the syntax of regular logic formulae to that of GCQ diagrams and back. Note that given $\varphi \in \mathcal{Q}_\Sigma$ with $|\mathtt{fv}(\varphi)| = n$, we have $\Theta(\varphi) \in \mathbf{CB}(\Sigma)(0, n)$ and similarly we have $\mathtt{fv}(\Lambda(d)) = m + n$ for $d \in \mathbf{CB}(\Sigma)(m, n)$, i.e. open wires correspond to free variables. □

The category $\mathbf{Rel}$ of sets and relations with Cartesian product as tensor, singleton as unit, the diagonal and its transpose as monoid and comonoid, subset as partial order, is a Cartesian bicategory. Given a set U, the subcategory $\mathbf{Rel}|_U \hookrightarrow \mathbf{Rel}$ with natural numbers $m, n \in \mathbb{N}$ as objects and relations $R \subseteq U^{m+n}$ as arrows is also a Cartesian bicategory. It is furthermore a PROP, i.e. a symmetric monoidal category with addition of natural numbers as tensor on objects.

Proposition 2.9. ([3, prop. 23]) Models $K \in \mathcal{M}_\Sigma(U)$ are in bijective correspondence with identity-on-objects morphisms of Cartesian bicategories $K : \mathbf{CB}(\Sigma) \to \mathbf{Rel}|_U$.

Proof. By the universal property of the free Cartesian bicategory, an identity-on-objects morphism $K : \mathbf{CB}(\Sigma) \to \mathbf{Rel}|_U$ is uniquely determined by its image on generators $\{ K(R) \subseteq U^{\mathtt{ar}(R)} \}_{R \in \Sigma}$: this is precisely the data for a Σ-model. □

Corollary 2.10. Let $[\mathbf{CB}(\Sigma), \mathbf{Rel}]$ denote the set of morphisms of Cartesian bicategories, there are bijective correspondences:

$$\mathbf{CB}(\Sigma)(0,0) \overset{(1)}{\simeq} \{\varphi \in \mathcal{Q}_\Sigma \mid \mathtt{fv}(\varphi) = \varnothing\} \overset{(2)}{\simeq} \mathcal{M}_\Sigma \overset{(3)}{\simeq} [\mathbf{CB}(\Sigma), \mathbf{Rel}]$$

Proof. (1) follows from theorem 2.8, (2) from theorem 2.6 and (3) is proposition 2.9. $\square$

3 RelCoCat Semantics and Natural Language Entailment

A *concrete* RelCoCat model is a rigid monoidal functor $F : \mathbf{G} \to \mathbf{Rel}$ for $\mathbf{G}$ the rigid monoidal category generated by a pregroup grammar $G = (V, B, \Delta)$. We require that the image for words $w \in V$ be the singleton $F(w) = 1$, hence the image for a dictionnary entry $(w, t) \in \Delta$ is given by a subset $F(w \to t) \subseteq F(t)$. We also assume the image of F lies in $\mathbf{Rel}|_U$ for some finite universe U, which may be taken to be the union of the universe for each basic type $U = \bigcup_{b \in B} F(b)$.

Lemma 3.1. A RelCoCat model $F : \mathbf{G} \to \mathbf{Rel}|_U$ is uniquely determined by its image on basic types and on dictionnary entries, i.e. by a function $\mathtt{ar} : B \to \mathbb{N}$ and a subset $F(w \to t) \subseteq U^{\mathtt{ar}(t)}$ for each $(w, t) \in \Delta$. Thus, F induces a model $K \in \mathcal{M}_\Delta(U)$ over the dictionnary seen as a signature where entries $(w, t) \in \Delta$ are symbols of arity $F(t) \in \mathbb{N}$.

Proof. This follows from lemma 1.10 and the universal property of the free rigid category: the functor $F : \mathbf{G} \to \mathbf{Rel}|_U$ is uniquely defined by its image on generators, i.e. on the basic types $b \in B$ and the dictionnary entries $(w, t) \in \Delta$ seen as generating arrows $w \to t$. $\square$

Hence, a concrete RelCoCat model $F : \mathbf{G} \to \mathbf{Rel}$ is fully specified by a finite set of triples $\{\, w : t :: F(w \to t)\,\}_{(w,t)\in\Delta}$, called a *pregroup lexicon* in [6]. This allows us to define `Semantics` as a function problem with RelCoCat models as input.

Definition 3.2. `Semantics`

Input: $r \in \mathbf{G}(u, s), \quad F : \mathbf{G} \to \mathbf{Rel}|_U$
Output: $F(r) \subseteq U^{F(s)}$

Lemma 3.3. Every concrete RelCoCat model $F : \mathbf{G} \to \mathbf{Rel}|_U$ factorises as $F = K \circ L$ for a relational model $K \in \mathcal{M}_\Delta(U)$ and a rigid monoidal functor $L : \mathbf{G} \to \mathbf{CB}(\Delta)$.

Proof. By propositions 2.9 and 3.1, $F : \mathbf{G} \to \mathbf{Rel}|_U$ induces a morphism of Cartesian bicategories $K : \mathbf{CB}(\Delta) \to \mathbf{Rel}|_U$. The functor $L : \mathbf{G} \to \mathbf{CB}(\Delta)$ sends each dictionnary entry to itself as a relational symbol, by construction we have $K \circ L = F$. $\square$

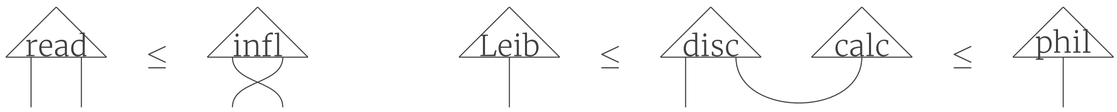
Proposition 3.4. There is a logspace reduction from `Semantics` to conjunctive query `Evaluation`, hence `Semantics` $\in$ NP.

Proof. The factorisation $K \circ L = F$ of lemma 3.3 and the translation Λ of theorem 2.8 are in logspace, they give a query $\varphi = \Lambda(L(r)) \in \mathcal{Q}_\Delta$ such that $\texttt{eval}(\varphi, K) = F(r)$. $\square$

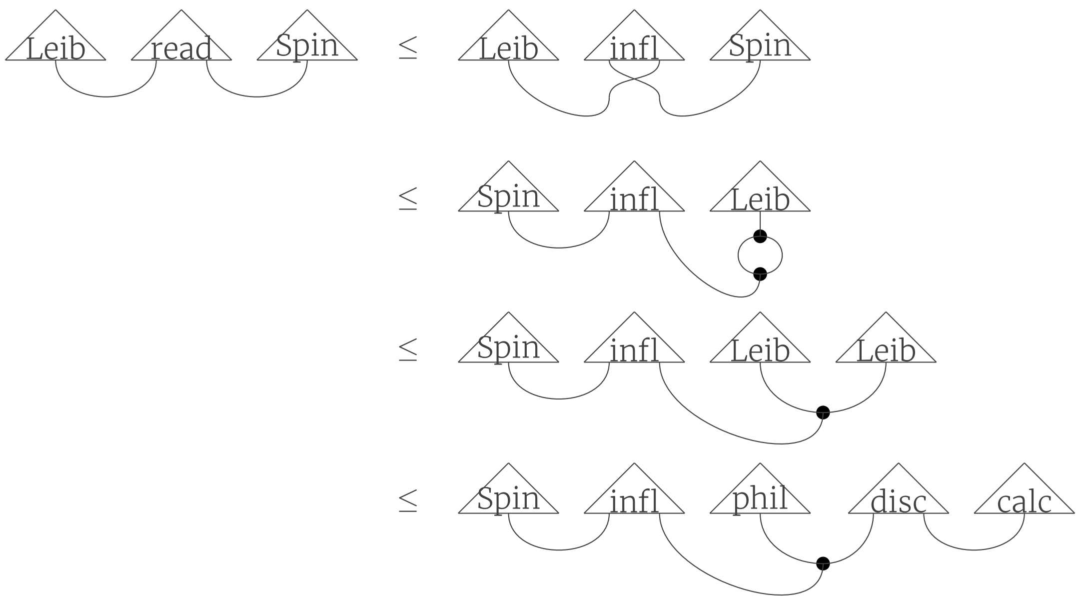
We conjecture that the constraint language induced by a pregroup grammar meets the tractability condition for the CSP dichotomy theorem [18].

Conjecture 3.5. Fix $G = (V, B, \Delta)$, `Semantics` is poly-time computable in the size of $(u, s) \in List(V) \times P(B)$ and in the size of the universe U.

We can also consider *abstract* RelCoCat models, i.e. a rigid monoidal functors $L : \mathbf{G} \to \mathbf{C}$ for a finitely presented Cartesian bicategory $\mathbf{C}$. For example, take $\mathbf{C}$ to be generated by the signature $\Sigma = \{\text{Leib}, \text{Spin}, \text{phil}, \dots\}$ as 1-arrows with codomain given by the function $\texttt{ar} : \Sigma \to \mathbb{N}$ and the following set of 2-arrows:



The 2-arrows of $\mathbf{C}$ encode *existential rules* of the form $\forall\, x_0 \cdots \forall\, x_k \cdot \varphi \to \varphi'$ for two conjunctive queries φ, φ' with $\texttt{fv}(\varphi) = \texttt{fv}(\varphi') = \{x_0, \dots, x_k\}$, also called tuple-generating dependencies in database theory, see [19] for a survey. The composition of 2-arrows in $\mathbf{C}$ then allow us to compute entailment, e.g.:



where the second and third inequations follow from the axioms of definition 2.7 (adjointness and lax comonoidality), the first and last from the generators.

Definition 3.6. `Entailment`

Input: $r \in \mathbf{G}(u, s), \quad r' \in \mathbf{G}(u', s), \quad L : \mathbf{G} \to \mathbf{C}$

Output: $L(r) \leq L(r')$

Proposition 3.7. `Entailment` is undecidable for finitely presented Cartesian bicategories. When $\mathbf{C}$ is freely generated, the problem reduces to conjunctive query `Containment`.

Proof. Entailment of conjunctive queries under existential rules is undecidable, see [20]. When $\mathbf{C} = \mathbf{CB}(\Sigma)$ is freely generated by a relational signature Σ, i.e. with no existential rules, theorem 2.8 yields a logspace reduction to `Containment`: `Entailment` $\in$ NP. □

Abstract RelCoCat models in free Cartesian bicategories make `Entailment` a decidable problem, they also allow us to reformulate lemma 3.3 as follows: every concrete model $F : \mathbf{G} \to \mathbf{Rel}$ factorises as $F = K \circ L$ for an abstract model $L : \mathbf{G} \to \mathbf{CB}(\Sigma)$ and a relational model $K : \mathbf{CB}(\Sigma) \to \mathbf{Rel}$. In the next section, we use this fact to define `QuestionAnswering` as a computational problem.

4 Question Answering as an NP-complete Problem

We consider the following computational problem: given a natural language corpus and a question, does the corpus contain an answer? We show how to translate a corpus into a relational database so that question answering reduces to query evaluation. We fix an abstract RelCoCat model $L : \mathbf{G} \to \mathbf{CB}(\Sigma)$ with $L(s) = 0$, i.e. grammatical sentences are mapped to closed formulae. We assume that $L(q) = L(a)$ for q and a the question and answer types respectively, i.e. both are mapped to queries with the same number of free variable.

Lexical items such as "influence" and "Leibniz" are mapped to their own symbol in the relational signature Σ, whereas functional words such as relative pronouns are sent to the Frobenius algebra of $\mathbf{CB}(\Sigma)$, see [21].

We define a corpus as a set of sentences $u \in List(V)$ with parsing $r : u \to s$ in $\mathbf{G}$, i.e. a subset $C \subseteq \coprod_{u \in List(V)} \mathbf{G}(u, s)$. If we apply L independently to each sentence, the resulting queries have disjoint sets of variables. In order to obtain the desired database, we need to map variables to some designated entities: a standard natural language processing task called *entity linking* (EL).

Let $\varphi_C = \bigwedge_{r \in C} \Lambda(L(r))$ be the conjunction of each sentence in the corpus, where Λ is the translation from diagrams to conjunctive queries of theorem 2.8. We define an entity linking for C as a function $\mu : \mathtt{var}(\varphi_C) \to E$ for some finite set E of entities. Thus, we get the following algorithm for translating the corpus C with entity linking $\mu : \mathtt{var}(\varphi_C) \to E$ into a model:

1. translate each parsed sentence $r \in C$ into a conjunctive query $\Lambda(L(r)) \in \mathcal{Q}_\Delta$,
2. compute their conjunction $\varphi_C = \bigwedge_{r \in C} \Lambda(L(r))$ and the substitution $\mu(\varphi_C)$,
3. construct the corresponding canonical model $K = CM(\mu(\varphi_C)) \in \mathcal{M}_\Sigma(E)$.

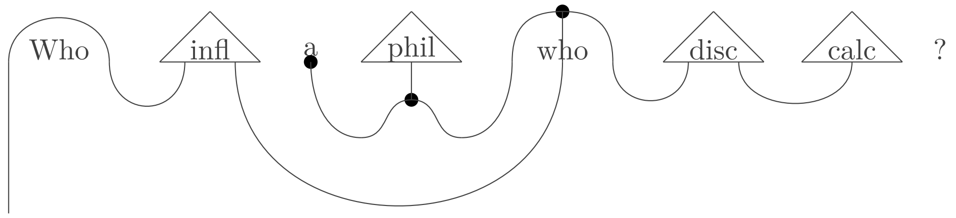
Definition 4.1. `QuestionAnswering`

$$\begin{array}{ll} \textbf{\textit{Input:}} & C \subseteq \coprod_{u \in List(V)} \mathbf{G}(u, s), \quad \mu : \mathtt{var}(\varphi_C) \to E, \quad \varphi \in \mathcal{Q}_\Sigma \\ \textbf{\textit{Output:}} & \mathtt{eval}(\varphi, K) \subseteq E^{\mathtt{fv}(\varphi)} \quad \text{where } K = CM(\mu(\varphi_C)) \end{array}$$

Theorem 4.2. `QuestionAnswering` is $\mathtt{NP} - \mathtt{complete}$.

Proof. Membership follows immediately by reduction to **`Evaluation`**. Hardness follows by reduction from graph homomorphism, we only give a sketch of proof and refer to [4] where EL is called matching. Any graph can be encoded in a corpus given by a set of subject-verb-object sentences, where EL maps nouns to their corresponding node. □

Example 4.3. We take $L : \mathbf{G} \to \mathbf{CB}(\Sigma)$ to map the question word "Who" to the compact-closed structure, the determinant "a" to the unit and the common noun "philosopher" to the symbol *phil* $\in \Sigma$ composed with the comonoid. We can now find the nouns that answer the question $r \in \mathbf{G}(u, q)$ of example 1.11 as the evaluation of the following query.

$L(r) \quad =$ 

$$\Lambda(L(r)) \quad = \quad \exists\, x_1 \, \exists\, x_2 \;\cdot\; \mathit{infl}(x_0, x_1) \;\wedge\; \mathit{phil}(x_1) \;\wedge\; \mathit{disc}(x_1, x_2) \;\wedge\; \mathit{calc}(x_2)$$

If "Spinoza influenced the philosopher Leibniz" and "Leibniz discovered calculus" are in the corpus C, we have $L(\mathit{Spinoza} \to n) \in \mathtt{QuestionAnswering}(C, \mu, \Lambda(L(r)))$.

We conclude with related work and potential directions for future work:

- text summarisation through conjunctive query minimisation [22],
- semantics of "How many?" questions and counting problems [23],
- many-sorted RelCoCat models with graphical regular logic [24],

- from Boolean semantics to generalised relations in arbitrairy topoi [25],
- from regular logic to description logics in bicategories of relations [26],
- comonadic semantics for bounded short-term memory [27],
- quantum speedup for question answering via Grover's search [28].

Acknowledgments

The authors would like to thank Bob Coecke, Dan Marsden, Antonin Delpeuch, Vincent Wang, Jacob Leygonie, Dusko Pavlovic, Rui Soares Barbosa and Samson Abramsky for inspiring discussions in the process of writing this article, as well as anonymous Reviewers of ACT2019 for constructive comments that improved the presentation of this work. K.M. is supported by the EPSRC NQIT Hub and Cambridge Quantum Computing Ltd. A.T. acknowledges Simon Harrison for financial DPhil support.

References

[1] Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh. A Compositional Distributional Model of Meaning. In *Proceedings of the Second Symposium on Quantum Interaction (QI-2008)*, pages 133–140, 2008.

[2] Ashok K. Chandra and Philip M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *Proceedings of the Ninth Annual ACM Symposium on Theory of Computing - STOC '77*, pages 77–90, Boulder, Colorado, United States, 1977. ACM Press.

[3] Filippo Bonchi, Jens Seeber, and Pawel Sobocinski. Graphical Conjunctive Queries. *arXiv:1804.07626 [cs]*, April 2018.

[4] Bob Coecke, Giovanni de Felice, Dan Marsden, and Alexis Toumi. Towards Compositional Distributional Discourse Analysis. *Electronic Proceedings in Theoretical Computer Science*, 283:1–12, November 2018.

[5] Anne Preller and Mehrnoosh Sadrzadeh. Semantic Vector Models and Functional Models for Pregroup Grammars. *Journal of Logic, Language and Information*, 20(4):419–443, 2011.

[6] Anne Preller. Natural language semantics in biproduct dagger categories. *J. Applied Logic*, 12(1):88–108, 2014.

[7] Anne Preller. From Logical to Distributional Models. *Electronic Proceedings in Theoretical Computer Science*, 171:113–131, December 2014.

[8] Mehrnoosh Sadrzadeh, Dimitri Kartsaklis, and Esma Balkır. Sentence entailment in compositional distributional semantics. *Annals of Mathematics and Artificial Intelligence*, 82(4):189–218, April 2018.

[9] Joachim Lambek. Type Grammar Revisited. In Alain Lecomte, François Lamarche, and Guy Perrier, editors, *Logical Aspects of Computational Linguistics*, pages 1–27, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[10] Wojciech Buszkowski and Katarzyna Moroz. Pregroup Grammars and Context-free Grammars. *Computational Algebraic Approaches to Natural Language, Polimetrica*, 121, 2008.

[11] Anne Preller and Joachim Lambek. Free compact 2-categories. *Mathematical Structures in Computer Science*, 17(2):309–340, 2007.

[12] André Joyal and Ross Street. Planar diagrams and tensor algebra. 1988.

[13] P. Selinger. A Survey of Graphical Languages for Monoidal Categories. *New Structures for Physics*, pages 289–355, 2010.

[14] R Oehrle. A parsing algorithm for pregroup grammars. *Proceedings of Categorial Grammars 2004*, pages 59–75, January 2004.

[15] Anne Preller. Linear Processing with Pregroups. *Studia Logica: An International Journal for Symbolic Logic*, 87(2/3):171–197, 2007.

[16] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[17] A. Carboni and R. F. C. Walters. Cartesian bicategories I. *Journal of Pure and Applied Algebra*, 49(1):11–32, November 1987.

[18] Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. *arXiv:1703.03021 [cs]*, March 2017.

[19] Michaël Thomazo. *Conjunctive Query Answering Under Existential Rules - Decidability, Complexity, and Algorithms*. PhD thesis, October 2013.

[20] J. F. Baget and M. L. Mugnier. Extensions of Simple Conceptual Graphs: The Complexity of Rules and Constraints. *Journal of Artificial Intelligence Research*, 16:425–465, June 2002.

[21] Mehrnoosh Sadrzadeh, Stephen Clark, and Bob Coecke. The Frobenius anatomy of word meanings I: Subject and object relative pronouns. *Journal of Logic and Computation*, 23:1293–1317, 2013.

[22] Chandra Chekuri and Anand Rajaraman. Conjunctive query containment revisited. *Theoretical Computer Science*, page 19, 2000.

[23] Giorgio Stefanoni, Boris Motik, and Egor V. Kostylev. Estimating the Cardinality of Conjunctive Queries over RDF Data Using Graph Summarisation. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 1043–1052, 2018.

[24] Brendan Fong and David I. Spivak. Graphical Regular Logic. *arXiv:1812.05765 [cs, math]*, December 2018.

[25] Bob Coecke, Fabrizio Genovese, Martha Lewis, Dan Marsden, and Alex Toumi. Generalized relations in linguistics & cognition. *Theoretical Computer Science*, 2018.

[26] Evan Patterson. Knowledge Representation in Bicategories of Relations. *arXiv:1706.00526 [cs, math]*, June 2017.

[27] Samson Abramsky and Nihil Shah. Relating Structure and Power: Comonadic Semantics for Computational Resources. *arXiv:1806.09031 [cs]*, June 2018.

[28] William Zeng and Bob Coecke. Quantum Algorithms for Compositional Natural Language Processing. *Electronic Proceedings in Theoretical Computer Science*, 221:67–75, August 2016.