1. Introduction

Definition 1.1. We define information to be anything that can be expressed purely in strings of symbols, given a convention. Dually, a convention is that which provides context so that certain strings of symbols can convey information.

It would be nice to have a grammar with which to specify a convention, but I would suspect that this cannot be done justice.

1.1. Informatics. Informatics is the study of information and its expression. An expression $E$ is always created by an individual $A$ having a convention $C_A$ and with the intent to transmit information to an individual or group $B$, known by its convention $C_B$.

What is communicated in the number 1, given the standard mathematical convention? What is expressed in the symbols $(x, y) \in A \times A$ where $A = \langle +, -, 0 \rangle$ is an abelian group and $x + 2y = 0$ given this convention? If the author is $A$ and the reader is $B$, that which has been communicated in Expression (1) can serve as an example of information and hence allow the reader to witness an example of the author’s convention.

1.2. Purpose of this paper. In this paper, I will attempt to communicate about information using category theory and about category theory using examples from informatics. More concretely, I will first define categories and functors. I will then give many examples of them using well-known ideas from information science, such as databases and ontologies.

The mathematician may hope to be reminded of why we study databases, and the informatician may hope to be reminded why we might spend time learning category theory. The first is easier. Since information can be communicated in symbols, it can be made tangible and stored in computers. In order to study it, we should examine that which we have currently have stored and in particular the symbolic forms in which it is currently stored.

The author’s preferred storage devices are currently categorical databases. While these have not been implemented on computers, there exists a convention within which they can be communicated in symbols – the authors’.

The database practitioner and working ontologist might be interested to learn category theory for the same reason people tend to appreciate what little math they do know: Mathematics, and category theory in particular, is currently humanity’s best language for conveying information rigorously and precisely.

Thanks to Tristan Nguyen and the ONR for the grant: N000140910466.
2. What is category theory about

Category theory is perhaps an undefined term, but it certainly involves the language of “categories and functors.” Roughly, a category is a (possibly infinite) database schema, and a functor is data on that schema. This connection will be made precise in Definitions ?? and ??.

Definition 2.1. A graph is a tuple $\mathcal{G} := \langle \text{Ob}_G, \text{Arr}_G, s_G, t_G \rangle$, where $\text{Ob}_G$ and $\text{Arr}_G$ are sets, and $s_G : \text{Arr}_G \to \text{Ob}_G$ and $t_G : \text{Arr}_G \to \text{Ob}_G$ are total functions. We express this by the diagram of sets

\[
\begin{array}{ccc}
\text{Ob}_G & \xleftarrow{s_G} & \text{Arr}_G \\
& \neq & \\
\text{Ob}_G & \xrightarrow{t_G} & \text{Arr}_G
\end{array}
\]

The elements of $\text{Ob}_G$ are called the objects of $\mathcal{G}$, and the elements of $\text{Arr}_G$ are called the arrows of $\mathcal{G}$. The function $s_G$ is called the source and the function $t_G$ is called the target, and for an arrow $f \in \text{Arr}_G$, we refer to $s(f)$ as the source of $f$ and $t(f)$ as the target of $f$.

The following constitutes the definition of a related concept, where instead of using the conventions of category theory, we do our best to emulate database convention.

Definition 2.2. A named graph is that which can be expressed within the database schema,

(2)

\[
\begin{array}{ccc}
\text{Node} & \xrightarrow{\text{source}} & \text{Arrow} \\
\text{id} & \xrightarrow{} & \text{id} \\
\text{name} & \xrightarrow{} & \text{name}
\end{array}
\]

Definition 2.3. A named reflexive graph is that which can be expressed within the database schema,

(3)

\[
\begin{array}{ccc}
\text{Node} & \xrightarrow{\text{primary}} & \text{Arrow} \\
\text{id} & \xrightarrow{} & \text{id} \\
\text{primary arrow} & \xrightarrow{} & \text{source node} \\
\text{name} & \xrightarrow{} & \text{target node} \\
\end{array}
\]

under the constraint that

\[N.\text{primary.source} = N.\text{primary.target} = N\]

for all nodes $N = \langle N.\text{id}, N.\text{primary}, N.\text{name} \rangle$.

Definition 2.4. A reflexive graph is a tuple $\mathcal{G} := \langle \text{Ob}_G, \text{Arr}_G, p_G, s_G, t_G \rangle$ where $\text{Ob}_G$ and $\text{Arr}_G$ are sets, and $p_G : \text{Ob}_G \to \text{Arr}_G$, $s_G : \text{Arr}_G \to \text{Ob}_G$, and $t_G : \text{Arr}_G \to \text{Ob}_G$ are total functions such that

\[s_G \circ p_G(X) = t_G \circ p_G(X) = X\]
for any element $X \in \text{Ob}_G$. We express this by the diagram of sets

$$\begin{array}{c}
\text{Ob}_G \\
\downarrow s_G \\
\text{Arr}_G \\
\uparrow t_G \\
\hline
\end{array}$$

**Definition 2.5.** A *categorical database schema* is that which can be expressed within the following database schema.

(4) Table

<table>
<thead>
<tr>
<th>id</th>
<th>primary key column</th>
<th>name</th>
</tr>
</thead>
</table>

Column

| id | source table | foreign referent table | name |

Clearly, categorical databases and named reflexive graphs have isomorphic schemas. Thus we can make the following mathematical definition:

**Definition 2.6.** A *database schema* $\mathcal{D}$ consists of a reflexive graph $G$ and a functor $\delta : G \to \text{Sets}$. We call $G$ the *schema* of $\mathcal{D}$ and we call $\delta$ the *data* of $\mathcal{D}$.

**Example 2.7.** Here is an example of a database schema, expressed as in Definition 2.5.

**Definition 2.8.** A *categorical database* is that which can be expressed within the following database schema.

(5) Table

| id | primary_key_column | name |

Column

| id | source_table | value_table | name |

ROW

| id | source_table | primary_key_cell |

CELL

| id | source_column | source_row | value_row |

subject to the following constraints for all $T \in \text{TABLE}$, $R \in \text{ROW}$, $C \in \text{CELL}$:

(6) $T.\text{primary_key_column}.\text{source_table} = T$

(7) $T.\text{primary_key_column}.\text{target_table} = T$

(8) $R.\text{primary_key_cell}.\text{source_row} = R$

(9) $R.\text{primary_key_cell}.\text{current_value}.\text{value_row} = R$

(10) $C.\text{source_column}.\text{source_table} = C.\text{source_row}.\text{source_table}$

(11) $C.\text{value_row}.\text{source_table} = C.\text{source_column}.\text{value_table}$
and that

\[ \pi_{\text{id}, \text{source column}, \text{source row}}(\text{CELL}) = \text{COLUMN} \times_{\text{source table} = \text{source table}} \text{ROW} \]

3. **Examples from Informatics**

In this section we define other things from informatics in terms of categories.

3.1. **RDF**.

3.2. **Context free grammars**.