Data Semantics and Linguistic Semantics

Reinhard Muskens

Tilburg University, Tilburg, The Netherlands
r.a.muskens@gmail.com

1 Introduction

Lexical semantics deals with the meaning of words, while phrasal semantics deals with the meaning of more complex constituents. In this note I will argue that the relation between the two can fruitfully be modeled with the help of a version of Frank Veltman's Data Semantics (Veltman [15, 16], Landman [9]).

As Veltman describes in [16, pp. 208–218], Data Semantics can (but need not) be based on an ontology of *(possible) facts*. Facts will have a twofold role in this case. On the one hand they are the things described by the atomic sentences of the logical language under consideration; on the other, certain collections of possible facts ('*data sets*') play a role analogous to the role of possible worlds in modal logic, in the sense that all sentences of the logical language are evaluated relative to collections of facts and the inclusion relation on these sets plays the role of an accessibility relation. Each data set corresponds to the evidence we conceivably could have obtained about our surroundings, just as possible worlds can be thought of as ways the world might have been (Lewis [10]).

The theory in [15, 16] is presented in the form of a semantics for a simple logical language (essentially the language of propositional modal logic) and there is no need to scale things up to a logic that can be used for doing natural language semantics. But from the perspective of the present note, such a scaling up is useful and I will make a start with it by transcribing a version of Data Semantics into a higher order logic and using the result for interpreting a tiny fragment of natural language. It will be shown how *frames*, a popular vehicle for the representation of lexical knowledge (see, e.g., Barsalou [1, 2], Löbner [11], Petersen & Osswald [13]) can be modeled as representations of certain complex facts supporting simple statements of the *John likes Mary* kind and how the transcribed machinery of Data Semantics can then be used to obtain translations of more complex statements in a fairly standard, Montague-like fashion.

From the viewpoint of logical semantics, the possible advantages of the undertaking are at least threefold. One possible gain is that the rather crude semantics usually given to open class words can now be replaced by a much subtler one, giving rise to the hope that the richer structure thus obtained can be put to good use in explaining features of phrasal composition (see also Petersen & Osswald [13]). Other advantages, which directly stem from Data Semantics, are a very intuitive treatment of the ontology of falsity and the incorporation into linguistic semantics of the data semantics treatment of *instability phenomena*. Falsity is explained as *incompatibility with the given, positive, facts* in data semantics. That the grass is blue is not just false because the grass is in the negative extension of *blue*; it is false because the grass is green and that fact is incompatible with the possible fact of the grass being blue. Instability is the phenomenon that a statement which is true on the basis of limited factual knowledge may become false once that knowledge has increased. A standard example example is *it may be raining*, which may be true on the basis of the facts available to me before, but not after, I have looked out of the window to ascertain that it is not raining.

Combining lexical and phrasal semantics into one formal theory is a move that seems inescapable if we are to have an adequate theory of natural meaning, and I do not view the two as

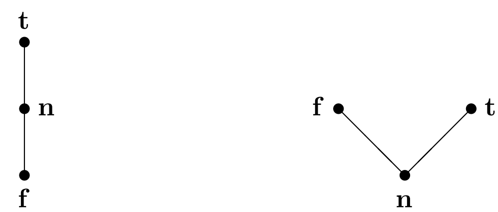

Figure 1: Hasse diagrams for the *logical* ordering of $\{\mathbf{t}, \mathbf{f}, \mathbf{n}\}$ (left) and the *approximation* ordering of this set (right).

overall competing. But inasfar as the two theories have made incompatible claims in the past some of these claims have to be given up when merging them, of course. In particular, I reject the idea (defended in Barsalou [2], who explicitly discusses frame representations of negation, disjunction, and universal quantification) that *all* natural language meaning can profitably be represented with the help of frames. Frames seem capable to model one layer of linguistic meaning well and many if not all open class words seem to refer to meanings representable with the help of frames. But it seems there is a second layer of natural language meaning that expresses ideas that are not representable in this way and is best treated with the help of the logical tools that phrasal semantics provides us with. While some phrases denote in the first layer, many do not. Data Semantics suggests a distinction between representations of facts and representations of propositions and offers a way to deal with their interrelations that seems highly relevant for connecting these two layers of linguistic meaning.

2 Partial Type Theory

Data Semantics provides a partial logic in the sense that it is possible for a proposition to be neither true nor false in a model on the basis of the evidence given. Since we want to transcribe Data Semantics with the help of a higher order logic, the latter should arguably also allow for this possibility. The logic we choose is the three-valued three-sorted functional type theory TY_2^3 described in Muskens [12]. Apart from the propositional type t, there will be a type $\mathfrak{f}$ of facts and a type e of entities. For all details concerning the logic I refer to [12]; here I only draw attention to the few properties that distinguish it from the two-sorted higher order logic described in Gallin [5] (which in its turn differs only marginally from the logic in Church [4]). The main difference between these logics and TY_2^3 is the latter's three-valuedness—sentences can be true ($\mathbf{t}$), false ($\mathbf{f}$), or neither true nor false ($\mathbf{n}$). In Figure 1 two orderings on these three values are depicted with the help of Hasse diagrams. The one on the left is called the *logical* ordering. Conjunction and disjunction will be interpreted as the meet and join in this lattice, so that a Strong Kleene evaluation scheme ensues. The ordering on the right is the *approximation* ordering. The meet operation in this ordering is sometimes called *consensus*. We will let $\otimes$ denote it.

Figure 2 gives a truth table for this operator, but also one for the one-place connective T. This connective checks whether its argument has the value $\mathbf{t}$ and returns $\mathbf{t}$ when it does, but $\mathbf{f}$ when it does not.

The identity symbol '=' likewise always returns either $\mathbf{t}$ or $\mathbf{f}$; $A = B$ will get the value $\mathbf{t}$ if the value of A indeed equals the value of B, but the value $\mathbf{f}$ otherwise.

$\otimes$	**t**	**f**	**n**
t	**t**	**n**	**n**
f	**n**	**f**	**n**
n	**n**	**n**	**n**

T	
t	**t**
f	**f**
n	**f**

Figure 2: Truth tables for the consensus operator $\otimes$ and the operator T (‘true and not false’).

This ends our sketch of the logic TY_2^3. The logic's propositional part is three-valued Strong Kleene, but it's rules for quantification and lambda conversion are the familiar ones.

3 Facts

The structure of the $\mathfrak{f}$ domains in our models will be constrained by axioms we directly borrow from [16]. Let 0 be a constant of type $\mathfrak{f}$ (denoting the *improper* fact) and let $\circ$, for which we use infix notation, be of type $\mathfrak{f}(\mathfrak{f}\mathfrak{f})$. (‘$f \circ g$’ is to be read as ‘the combination of f and g’.) Universal closures of the following will be axioms.

- $\exists f\, f \neq 0$
- $f \circ f = f$
- $f \circ g = g \circ f$
- $(f \circ g) \circ h = f \circ (g \circ h)$
- $0 \circ f = 0$

Here we have used f, g and h as variables of type $\mathfrak{f}$.

We say that f and g are *incompatible* if $f \circ g = 0$ and that f *incorporates* g, or $f \leq g$, also written $g \geq f$, if $f \circ g = f$. Given the axioms above it is immediate that incorporation is a partial ordering. We will let the type $(\mathfrak{f}t)t$ constant $\mathcal{F}$ be short for the following term (where F is a variable of type $\mathfrak{f}t$).

$$\lambda F.\, F \neq (\lambda f.\top) \wedge \forall fg.(Ff \wedge Fg) = F(f \circ g)$$

Given this definition, $\mathcal{F}$ singles out the subsets of the domain of facts that are (proper) *filters*. Note that, due to the fact that = always gives bivalent results, and bivalence is closed under the usual logical operations in TY_2^3, all axioms are bivalent, as are the notions of incompatibility and incorporation and statements of the form $\mathcal{F}F$ (i.e. ‘F is a filter’).

4 Information Models

In Data Semantics an *information model* is a triple $\langle S, \preceq, \mathcal{V}\rangle$ such that S is nonempty (a set of *information states*), $\preceq$ is a partial ordering on S, and $\mathcal{V}$ is a function with domain S such that, for each $s \in S$, $\mathcal{V}_s$ is a partial function assigning at most one of **t** or **f** to each atomic sentence of the language under consideration, with $\mathcal{V}_s \subset \mathcal{V}_{s'}$ if $s \preceq s'$.

Information models can be used to assign truth and falsity to the constructs of the logical language under consideration. Here is the relevant definition.

Definition 1. *Let $\langle S, \preceq, \mathcal{V}\rangle$ be an information model and let $s \in S$.*

– *if φ is atomic then*
  $s \models \varphi$ *iff* $\mathcal{V}_s(\varphi) = \mathbf{t}$*;*
  $s \dashv \varphi$ *iff* $\mathcal{V}_s(\varphi) = \mathbf{f}$

– $s \models \neg\varphi$ *iff* $s \dashv \varphi$*;*
  $s \dashv \neg\varphi$ *iff* $s \models \varphi$*;*

– $s \models \varphi \wedge \psi$ *iff* $s \models \varphi$ *and* $s \models \psi$*;*
  $s \dashv \varphi \wedge \psi$ *iff* $s \dashv \varphi$ *or* $s \dashv \psi$*;*

– $s \models \varphi \vee \psi$ *iff* $s \models \varphi$ *or* $s \models \psi$*;*
  $s \dashv \varphi \vee \psi$ *iff* $s \dashv \varphi$ *and* $s \dashv \psi$*;*

– $s \models \text{may } \varphi$ *iff* $s' \models \varphi$ *for some* $s' \succeq s$*;*
  $s \dashv \text{may } \varphi$ *iff* $s' \models \varphi$ *for no* $s' \succeq s$*;*

– $s \models \text{must } \varphi$ *iff* $s' \dashv \varphi$ *for no* $s' \succeq s$*;*
  $s \dashv \text{must } \varphi$ *iff* $s' \dashv \varphi$ *for some* $s' \succeq s$*;*

– $s \models \varphi \rightarrow \psi$ *iff* $s' \models \varphi$ *and* $s' \dashv \psi$ *for no* $s' \succeq s$*;*
  $s \dashv \varphi \rightarrow \psi$ *iff* $s' \models \varphi$ *and* $s' \dashv \psi$ *for some* $s' \succeq s$*.*

5 What will be our Information States?

We will want to base our information states on facts in some way, but there is more than one way to do it. In [16], Veltman considers models $\mathcal{M} = \langle \mathfrak{F}, \circ, 0, \mathcal{I}\rangle$ in which $\langle \mathfrak{F}, \circ, 0\rangle$ is a structure satisfying the axioms given in setion 3 (a *data lattice*) and $\mathcal{I}$ is a function sending atomic formulas to elements of $\mathfrak{F}$, i.e. to facts. A *(possible) data set* $\mathcal{D}$ on $\langle \mathfrak{F}, \circ, 0\rangle$ is a subset of $\mathfrak{F}$ with the property that finite combinations of its elements are compatible, i.e. such that $f_1 \circ \ldots \circ f_n \neq 0$, for any $f_1, \ldots, f_n \in \mathcal{D}$. Data sets essentially are subsets of filters in $\langle \mathfrak{F}, \circ, 0\rangle$. Veltman then considers the truth and falsity conditions of formulas in models $\mathcal{M} = \langle \mathfrak{F}, \circ, 0, \mathcal{I}\rangle$ relative to data sets in $\langle \mathfrak{F}, \circ, 0\rangle$, using clauses such as the following.

– $\mathcal{D} \models_{\mathcal{M}} \varphi$ iff $\mathcal{I}(\varphi) \in \mathcal{D}$, if φ is atomic;
  $\mathcal{D} \dashv_{\mathcal{M}} \varphi$ iff $f \circ \mathcal{I}(\varphi) = 0$, for some $f \in \mathcal{D}$, if φ is atomic;

– $\mathcal{D} \models_{\mathcal{M}} \varphi \rightarrow \psi$ iff $\mathcal{D}' \models_{\mathcal{M}} \varphi$ and $\mathcal{D}' \dashv_{\mathcal{M}} \psi$ for no data set $\mathcal{D}' \supset \mathcal{D}$;
  $\mathcal{D} \dashv_{\mathcal{M}} \varphi \rightarrow \psi$ iff $\mathcal{D}' \models_{\mathcal{M}} \varphi$ and $\mathcal{D}' \dashv_{\mathcal{M}} \psi$ for some data set $\mathcal{D}' \supset \mathcal{D}$.

The information models of the previous section are thus fleshed out in a way that identifies the set of information states with the set of data sets on a given data lattice, with $\preceq$ instatiated by the subset relation on this set. It seems, however, that there are at least two other possibilities.

- Information states could be the *filters* on a given data lattice, or, simpler,
- they could be the *facts* of a given data lattice, with the ordering $\preceq$ given by the relation $\geq$ considered in section 3.

I will choose the last and simplest option here. But before this is worked out, let us have a very short look at the option of letting information states be filters.

6 Truth Relative to Filters

If the truth and falsity of statements are relativised to filters of facts, they will be of type $(\mathfrak{f}t)t$—given such a filter (of type $\mathfrak{f}t$, they will return a truth value. We will have an operator ϑ (for 'that') that will take a fact and turn it into an atomic statement of this kind. Its definition is as follows.

$$\vartheta \quad = \quad \lambda f \lambda F.(\mathcal{F}F \wedge Ff) \otimes (\mathcal{F}F \rightarrow \forall f'(Ff' \rightarrow f \circ f' \neq 0))$$

If f is a fact, then ϑf, which we will often write simply as $[f]$, will denote the partial set of F such that $(\mathcal{F}F \wedge Ff) \otimes (\mathcal{F}F \rightarrow \forall f'(Ff' \rightarrow f \circ f' \neq 0))$. This latter statement will be true if and only if $\mathcal{F}F \wedge Ff$ holds, i.e. iff F is a filter and $f \in F$; it will be false if and only if F is a filter and there is an f' such that Ff' and $f \circ f' = 0$. This clearly transcribes the atomic clause of Veltman's definition based on data models.

We can continue with operators expressing negation, conjunction, and disjunction (here p and q are of type $(\mathfrak{f}t)t$).

$$\begin{aligned} \textsf{not} &= \lambda p \lambda F.\neg pF \\ \textsf{and} &= \lambda pq \lambda F.pF \wedge qF \\ \textsf{or} &= \lambda pq \lambda F.pF \vee qF \end{aligned}$$

This gives a Strong Kleene semantics to these operators, which also matches Veltman's definition. The operators may, must, and if can be given the following definitions.

$$\begin{aligned} \textsf{may} &= \lambda p \lambda F \exists G.\mathcal{F}G \wedge F \subset G \wedge \mathsf{T}pG \\ \textsf{must} &= \lambda p \lambda F \neg \exists G.\mathcal{F}G \wedge F \subset G \wedge \mathsf{T}\neg pG \\ \textsf{if} &= \lambda pq \lambda F \neg \exists G.(\mathcal{F}G \wedge F \subset G \wedge \mathsf{T}pG \wedge \mathsf{T}\neg qG) \end{aligned}$$

These operators take central stage in Veltman [15, 16], as they lead to the instability phenomena studied there. Here they play a less prominent role, as our main theme is the combination of a lexical semantics based on representations of facts with a phrasal semantics in the logical tradition.

7 Truth Relative to Facts

While importing the ideas of data semantics into type logic seems feasible if information states are interpreted to be filters of facts or even (closer to the ideas in [16]) data sets, a semantics directly based on *facts* seems to work as well, and is perhaps a bit more straightforward. The idea now is that propositions are of type $\mathfrak{f}t$ and are thus evaluated with respect to (complex) facts. The operator ϑ, which turns facts into such propositions, will now be of type $\mathfrak{f}\mathfrak{f}t$ and can have the following definition.

$$\vartheta \quad = \quad \lambda g \lambda f.f \leq g \otimes f \circ g \neq 0$$

Now ϑg, or $[g]$, will denote a partial set of facts with 0 in its gap. Other facts f will be in the positive denotation of $[g]$ iff $f \leq g$ and in its negative denotation iff $f \circ g = 0$. The operators not, and, and or now are naturally treated as follows (with p and q now of type $\mathfrak{f}t$.

$$\begin{aligned} \mathsf{not} &= \lambda p \lambda f.\neg pf \\ \mathsf{and} &= \lambda pq \lambda f.(pf \wedge qf) \\ \mathsf{or} &= \lambda pq \lambda f.(pf \vee qf) \end{aligned}$$

The modals, on the other hand, can be treated in the following way.

$$\begin{aligned} \mathsf{may} &= \lambda p \lambda f \exists g.(g \neq 0 \wedge g \leq f \wedge \mathsf{T}pg) \\ \mathsf{must} &= \lambda p \lambda f \neg \exists g.(g \neq 0 \wedge g \leq f \wedge \mathsf{T}\neg pg) \\ \mathsf{if} &= \lambda pq \lambda f \neg \exists g.(g \neq 0 \wedge g \leq f \wedge \mathsf{T}pg \wedge \mathsf{T}\neg qg) \end{aligned}$$

8 Facts and Frames

Frames in the Barsalou/Löbner tradition are representations of concepts, "recursively composed out of the attributes of the object to be represented, and the values of these attributes" (Löbner [11]). Readers familiar with work in computational linguistics will recognise them as (a semantic variant of) *feature structures*. Figure 3 gives some examples of frames taken from Petersen & Osswald [13]).

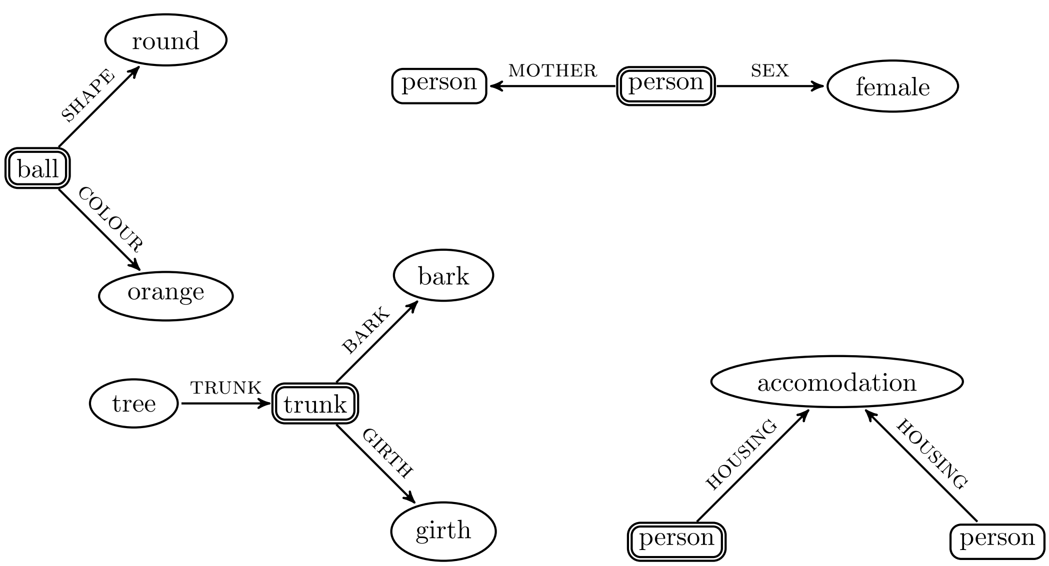

Figure 3: Frames for *basketball*, *mother*, *trunk*, and *flatemate* (from Petersen & Osswald [13]).

The idea is that the attributes in these labelled graphs are *functional*—a ball can have only one colour or shape, a person only one mother, etc. Whether this is a realistic constraint to put on conceptual representations (what to do with the concept of a *parent*?) is a matter that need not concern us here.

There are many ways to model feature structures with the help of logic. Kaspar & Rounds [8] give one that is specially dedicated to the topic; Blackburn [3] uses modal logic (which, after all, is the logic *par excellence* to talk about labeled graphs); Smolka [14] and Johnson [6, 7] use predicate logic. Here I will formalise frames with the help of complex (representations of) facts.

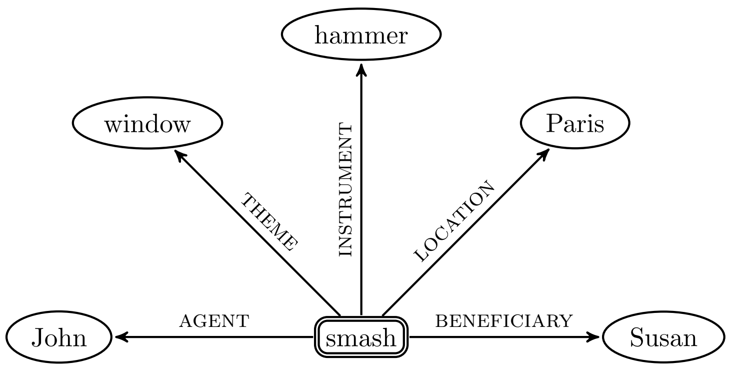

Figure 4: A frame for *In Paris, John smashed the window with a hammer for Susan.*

Consider the frame in Figure 4. The following, somewhat longish, term is a first approximation to its translation.

$$\begin{aligned}&\mathit{smash}\,x \circ \textsc{ag}\,xy_1 \circ \mathit{John}\,y_1 \circ \textsc{th}\,xy_2 \circ \mathit{window}\,y_2 \circ \textsc{instr}\,xy_3 \circ \mathit{hammer}\,y_3 \circ \textsc{loc}\,xy_4 \circ \mathit{Paris}\,y_4\\ &\qquad \circ \textsc{ben}\,xy_5 \circ \mathit{Susan}\,y_5\end{aligned}$$

Here *smash* is of type $e\mathfrak{f}$, AG is of type $ee\mathfrak{f}$, etc. The type e variables used here will be bound 'higher up' in the representation, as will become clear in the next section.

The functionality of attributes can be enforced with meaning postulates such as the following.

$$\forall xyz.y \neq z \rightarrow \textsc{th}\,xy \circ \textsc{th}\,xz = 0$$

So the frames with the help of which we categorise reality are representations of complex facts. Content words will also typically be associated with such representations. The next section will make a rudimentary beginning with the implementation of this idea.

9 A Very Small Fragment

Let us set up a small toy fragment in order to illustrate how our approach mixes lexical and phrasal semantics based on an ontology of facts. The following are a set of meaning postulates connecting non-logical constants with certain terms.

$$\begin{aligned}
\textsf{talk} &= \lambda x \lambda f \exists e.[\mathit{talk}\, e \circ \textsc{ag}\, ex]f \\
\textsf{man} &= \lambda x \lambda f \exists y.[\mathit{person}\, x \circ \mathit{sex}\, xy \circ \mathit{male}\, y]f \\
\textsf{apple} &= \lambda x \lambda f.[\mathit{apple}\, x]f \\
\textsf{eat} &= \lambda yx \lambda f \exists e.[\mathit{eat}\, e \circ \textsc{ag}\, ex \circ \textsc{th}\, ey]f \\
\textsf{every} &= \lambda P' P \lambda f \forall x.(P'xf \to Pxf) \\
\textsf{a} &= \lambda P' P \lambda f \exists x.(P'xf \wedge Pxf) \\
\textsf{the} &= \lambda P' P \lambda f \exists x.(\forall y(P'yf \leftrightarrow x = y) \wedge Pxf)
\end{aligned}$$

Here P is of type $e\mathfrak{f}t$, while *talk* is if type $e\mathfrak{f}$, *sex* is if type $ee\mathfrak{f}$, etc. The following are equivalent.

(1) a. $(\textsf{a man})\lambda x.(\textsf{a apple})\lambda y.\textsf{eat}\, yx$

b. $\lambda f \exists x(\exists z[\mathit{person}\, x \circ \mathit{sex}\, xz \circ \mathit{male}\, z]f \wedge \exists y[\mathit{apple}\, y]f \wedge \exists e.[\mathit{eat}\, e \circ \textsc{ag}\, ex \circ \textsc{th}\, ey]f)$

c. $\lambda f \exists xzye.[\mathit{person}\, x \circ \mathit{sex}\, xz \circ \mathit{male}\, z]f \wedge [\mathit{apple}\, y]f \wedge [\mathit{eat}\, e \circ \textsc{ag}\, ex \circ \textsc{th}\, ey]f$

As are the following.

(2) a. $(\textsf{every man})\lambda x.(\textsf{a apple})\lambda y.\textsf{eat}\, yx$

b. $\lambda f \forall x(\exists z[\mathit{person}\, x \circ \mathit{sex}\, xz \circ \mathit{male}\, z]f \to \exists ye([\mathit{apple}\, y]f \wedge [\mathit{eat}\, e \circ \textsc{ag}\, ex \circ \textsc{th}\, ey]f))$

Here we see the 'two-layer' approach to semantics at work. One layer is formed by representations of facts, while a second is the locus of the usual logical operators. In a very simple fragment like this the first layer is embedded in the second one, but not vice versa. But when the fragment is scaled up more intermingling may be necessary. If, for example, we want to model that John believes that pigs have wings, we may need to do it with the help of a proposition ('pigs have wings') that contains facts but is itself also contained in a fact that helps constitute a proposition.

10 Conclusion

In this note I have laid out some ideas on how to combine lexical and phrasal semantics with the help of Data Semantics. Data Semantics seems to be the right tool for the job because it sits well with the idea of, on the one hand, an ontology of positive facts supporting certain categorisations of the realities presented to us, and, on the other, the full gamut of logical operations present in natural language.

References

[1] L. Barsalou. Frames, Concepts, and Conceptual Fields. In A. Lehrer and E. F. Kittay, editors, *Frames, Fields, and Contrasts*, pages 21–74. Lawrence Erlbaum Associates, 1992.

[2] L. Barsalou. Perceptual Symbol Systems. *Behavioral and Brain Sciences*, 22:577–660, 1999.

[3] P. Blackburn. Modal Logic and Attribute-Value Structures. In M. de Rijke, editor, *Diamonds and Defaults*. Kluwer, Dordrecht, 1993.

[4] A. Church. A Formulation of the Simple Theory of Types. *Journal of Symbolic Logic*, 5:56–68, 1940.

[5] D. Gallin. *Intensional and Higher-Order Modal Logic.* North-Holland, Amsterdam, 1975.

[6] M. Johnson. *Attribute-Value Logic and the Theory of Grammar.* CSLI, Stanford, CA, 1988.

[7] M. Johnson. Logic and Feature Structures. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, Sydney, Australia, 1991.

[8] R. Kasper and W. Rounds. A Logical Semantics for Feature Structures. In *Proc. 24th meeting of the ACL*, pages 101–115, 1986.

[9] A. Landman. *Towards a Theory of Information.* PhD thesis, University of Amsterdam, 1986.

[10] D. Lewis. *On the Plurality of Worlds.* Blackwell, Oxford, 1986.

[11] S. Löbner. Functional Concepts and Frames. Available at `http://semanticsarchive.net/Archive/jI1NGEwO/`.

[12] R.A. Muskens. *Meaning and Partiality.* CSLI, Stanford, 1995.

[13] W. Petersen and T. Osswald. Concept Composition in Frames—Focusing on Genitive Constructions. In Th. Gamerschlag, D. Gerland, R. Osswald, and W. Petersen, editors, *Frames and Concept Types: Applications in Language and Philosophy*, volume 94 of *Studies in Linguistics and Philosophy*. Springer, Dordrecht, 2013.

[14] G. Smolka. A Feature Logic with Subsorts, 1988. Technical Report. IBM Deutschland.

[15] F. Veltman. Data Semantics. In J. Groenendijk, T.M.V. Janssen, and M. Stokhof, editors, *Truth, Interpretation, and Information; Selected Papers from the Third Amsterdam Colloquium*, pages 43–63. Foris, Dordrecht, 1984.

[16] F. Veltman. *Logics for Conditionals.* PhD thesis, University of Amsterdam, 1985.