Quantification in Frame Semantics with Binders and Nominals of Hybrid Logic

Laura Kallmeyer, Rainer Osswald, Sylvain Pogodalla

Quantification in Frame Semantics with Binders and Nominals of Hybrid Logic*

Laura Kallmeyer[1], *Rainer Osswald*[1], *and Sylvain Pogodalla*[1,2]
[1] Heinrich Heine Universität, Düsseldorf, Germany
{laura.kallmeyer,rainer.osswald}
@phil.uni-duesseldorf.de
[2] INRIA, Villers-lès-Nancy, F-54600, France
Université de Lorraine, LORIA, UMR 7503, Vandœuvre-lès-Nancy, F-54500, France
CNRS, LORIA, UMR 7503, Vandœuvre-lès-Nancy, F-54500, France
sylvain.pogodalla@inria.fr

ABSTRACT

This paper aims at integrating logical operators into frame-based semantics. Frames are semantic graphs that allow to capture lexical meaning in a fine-grained way but that do not come with a natural way to integrate logical operators such as quantifiers. The approach we propose starts from the observation that modal logic is a powerful tool for describing relational structures, hence frames. We use its hybrid logic extension in order to incorporate quantification and thereby allow for inference and reasoning. We integrate our approach to a type theoretic compositional semantics, formulated within Abstract Categorial Grammars. We also show how the key ingredients of hybrid logic, nominals and binders, can be used to model semantic coercion, such as the one induced by the *begin* predicate. In order to illustrate the effectiveness of the proposed syntax-semantics interface, all the examples can be run and tested with the Abstract Categorial Grammar development toolkit.

Keywords: Frame Semantics, Quantification, Hybrid Logic, Abstract Categorial Grammar

*This work was supported by the INRIA sabbatical program and by the CRC 991 "The Structure of Representations in Language, Cognition, and Science" funded by the German Research Foundation (DFG).

1 FRAMES AND LEXICAL SEMANTICS

Frames emerged as a representation format of conceptual and lexical knowledge (Fillmore 1977; Barsalou 1992; Löbner 2014). They are commonly presented as semantic graphs with labelled nodes and edges, such as the one in Figure 1, where nodes correspond to entities (individuals, events, …) and edges correspond to (functional or non-functional) relations between these entities. In Figure 1 all relations except *part-of* are meant to be functional.

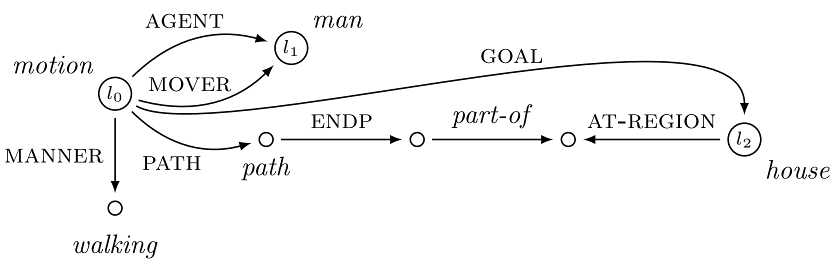

Figure 1: Frame for the meaning of *the man walked to the house* (adapted from Kallmeyer and Osswald 2013)

Structuring the knowledge as frames offers a fine-grained and systematic decomposition of meaning. This conception of frames is however not to be confused with the somewhat simpler FrameNet frames, although the former can help to capture the structural relations of the latter (see Osswald and Van Valin 2014).

Frames can be formalized as extended typed feature structures (Petersen 2007; Kallmeyer and Osswald 2013) and specified as models of a suitable logical language, the *labelled attribute-value description language* (LAVD language in short). Such a language allows for the composition of lexical frames on the sentential level by means of an explicit syntax-semantics interface (Kallmeyer and Osswald 2013).

1.1 *Logical Representation of Feature Structures*

The syntax-semantics of (Kallmeyer and Osswald 2013) relies on a formal representation of semantic frames as *base-labelled feature structure with types and relations*. This definition extends the standard definition of feature structures in two respects: In addition to features, proper relations between nodes can be expressed. Moreover, it is not required that every node is accessible from a single root node via a feature path; instead, it is required that every node is accessible from one of the base-labelled nodes. Semantic frames defined in this way can be seen as finite first-order structures which conform to a signature consisting of a set $\mathsf{Label} \cup \mathsf{Type}$ of unary relation symbols and a set $\mathsf{Feat} \cup \mathsf{Rel}$ of binary relation symbols subject to the constraints that the members of Label denote singletons, the members of Feat denote *functional* relations, and that the above accessibility condition holds. In the example frame of Figure 1, symbols inside nodes (l_0, l_1, …) indicate base labels, symbols attached to nodes (*man*, *motion*, …) belong to Type, members of Feat are marked by small caps (AGENT, ENDP, …), and *part-of* is the only member of Rel occurring in this frame.

But the logical framework of (Kallmeyer and Osswald 2013) does not provide means for explicit quantification. As a consequence, the referential entities of the domain of discourse are implicitly treated as definite, which is reflected by the *naming* of nodes l_0, l_1, etc.

Such relational structures can also easily be turned into Kripke structures. Thus, semantic frames, or feature structures, provide a natural application domain for modal languages and, in particular, for hybrid extensions because of the need to cope with node labels and feature path re-entrancies (Blackburn 1993).

1.2 *Semantic Frames and Hybrid Logic*

As Blackburn (1993) points out, attribute-value structures can be described using the logical language of *Hybrid Logic* (HL, cf. Areces and ten Cate 2007), an extension of the language of modal logic, well-suited to the description of graph structures like the one of Figure 1. HL introduces *nominals*, i.e., node names, that allow the logical formulas to refer to specific nodes of the graph. The nominal l_0 for instance refers to the *motion* node in Figure 1. It is then possible, for example, to specify that the AGENT and the MOVER edges from the node l_0 should meet on the same node in Figure 1. This additional expressiveness of HL over modal logic allows one to express node sharing in attribute value structures (Blackburn 1993). HL is an established logical formalism which has been extensively studied, in particular with respect to the addition of *variables* for nodes, and the associated *binders*, that can appear in the logical formulas. Its relation to attribute-value structures and its expressiveness make it a natural candidate to relate quantified expressions and frame semantics.

With respect to (Kallmeyer and Osswald 2013), the approach we propose here does not consider frames as "genuine semantic representations". The one-to-one equivalence between the logical formulas of the LAVD language of (Kallmeyer and Osswald 2013) and the frames as graph (or relational) structures relies on the existence of minimal models for such formulas. While HL with nominals, but without variables and binders are very close to the LAVD language, it is not obvious what the notion of minimal model of the latter becomes when using quantification. Thus, we have a more traditional view where the sense of an expression is an hybrid logical formula and its reference is computed against models. The latter are the frames we wish to consider. But, contrary to what happens with minimal models, they are then not fully specified by the logical formulas which serve as frame descriptions.

1.3 *Related Work*

Hybrid logic with nominals but without quantification over states was already used to describe semantic dependency graphs in (Baldridge and Kruijff 2002). Natural language quantification is there encoded using RESTR and BODY relations. However, it remains not clear how to compute relations between such representations (e.g., how to check that *John kisses Mary* holds in case *every man kisses Mary* holds). An additional step of interpretation of the graphs seems to be required.

A similar approach is proposed in (Kallmeyer and Richter 2014) for quantification in frame semantics. In this approach, "quantifier frames" also introduce RESTR and BODY attributes that point to nodes (typically representing an entity and an event, respectively). But they do not directly encode the truth conditions that would be associated with a model-theoretic interpretation. Bridging the gap between the quantifier frame and the model-theoretic interpretation requires the additional extraction of a predicate-logical formula, that, in turn, can be model-theoretically interpreted in order to compute the truth value of the expression.

In the approach we presently propose, and contrary to (Kallmeyer and Richter 2014), there is no quantifier frame as such. The quantifiers are part of the formulas constraining the frames that can make an expression true, and no additional interpretation is required. The logical operators and the frames (as models) are kept separate, following the approach suggested by (Muskens 2013).

2 HYBRID LOGIC AND SEMANTIC FRAMES

2.1 Hybrid Logic

We use the notations of (Areces and ten Cate 2007).

Definition 1 (Formulas). Let $\mathsf{Rel} = \mathsf{Func} \cup \mathsf{PropRel}$ be a set of functional and non-functional relational symbols, Prop a set of propositional variables, Nom a set of nominals (node names), and Svar a set of state variables. Let $\mathsf{Stat} = \mathsf{Nom} \cup \mathsf{Svar}$.

The language of formulas Forms is defined as:

$$\mathsf{Forms} ::= \top \mid p \mid s \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \langle R \rangle \phi \mid \mathbf{\exists}\phi \mid @_s\phi \mid \downarrow x.\phi \mid \exists x.\phi$$

where $p \in \mathsf{Prop}$, $s \in \mathsf{Stat}$, $R \in \mathsf{Rel}$, $x \in \mathsf{Svar}$, and $\phi, \phi_1, \phi_2 \in \mathsf{Forms}$.

Moreover, we define:

- $\mathbf{\forall}\phi \equiv \neg\mathbf{\exists}\neg\phi$
- $[R]\phi \equiv \neg\langle R \rangle\neg\phi$
- $\phi \Rightarrow \psi \equiv \neg\phi \vee \psi$

We call $\mathbf{\forall}$ and $[R]$ universal operators, and $\mathbf{\exists}$ and $\langle R \rangle$ existential operators. The elements of Func will be written in small caps.

Definition 2 (Model). A *model* $\mathscr{M}$ is a triple $\langle M, (R^{\mathscr{M}})_{R \in \mathsf{Rel}}, V \rangle$ such that M is a non-empty set, each $R^{\mathscr{M}}$ is a binary relation on M, and the valuation $V : \mathsf{Prop} \cup \mathsf{Nom} \longrightarrow \wp(M)$ is such that if $i \in \mathsf{Nom}$ then $V(i)$ is a singleton. An assignment g is a mapping $g : \mathsf{Svar} \longrightarrow M$. For an assignment g, g^x_m is an assignment that differs from g at most on x and $g^x_m(x) = m$. For $s \in \mathsf{Stat}$, we also define $[s]^{\mathscr{M},g}$ to be the only m such that $V(s) = \{m\}$ if $s \in \mathsf{Nom}$ and $[s]^{\mathscr{M},g} = g(s)$ if $s \in \mathsf{Svar}$.

Definition 3 (Satisfaction Relation). Let $\mathscr{M}$ be a model, $w \in M$, and

g an assignment for $\mathcal{M}$. The *satisfaction relation* is defined as follows:

$$
\begin{array}{lll}
\mathcal{M}, g, w \models \top & & \\
\mathcal{M}, g, w \models s & \text{iff} & w = [s]^{\mathcal{M},g} \text{ for } s \in \mathsf{Stat} \\
\mathcal{M}, g, w \models \neg\phi & \text{iff} & \mathcal{M}, g, w \not\models \phi \\
\mathcal{M}, g, w \models \phi_1 \wedge \phi_2 & \text{iff} & \mathcal{M}, g, w \models \phi_1 \text{ and } \mathcal{M}, g, w \models \phi_2 \\
\mathcal{M}, g, w \models \langle R \rangle \phi & \text{iff} & \text{there is a } w' \in M \text{ such that} \\
 & & R^{\mathcal{M}}(w, w') \text{ and } \mathcal{M}, g, w' \models \phi \\
\mathcal{M}, g, w \models p & \text{iff} & w \in V(p) \text{ for } p \in \mathsf{Prop} \\
\mathcal{M}, g, w \models @_s\phi & \text{iff} & \mathcal{M}, g, [s]^{\mathcal{M},g} \models \phi \text{ for } s \in \mathsf{Stat} \\
\mathcal{M}, g, w \models \downarrow x.\phi & \text{iff} & \mathcal{M}, g^x_w, w \models \phi \\
\mathcal{M}, g, w \models \exists x.\phi & \text{iff} & \text{there is a } w' \in M \text{ such that } \mathcal{M}, g^x_{w'}, w \models \phi \\
\mathcal{M}, g, w \models \mathbf{E}\phi & \text{iff} & \text{there is a } w' \in M \text{ such that } \mathcal{M}, g, w' \models \phi
\end{array}
$$

We can then check that $\mathcal{M}, g, w \models \mathbf{A}\phi$ iff $\forall w'\ \mathcal{M}, g, w' \models \phi$. $\mathbf{A}$ is the universal modality. $\mathbf{A}\phi$ states that the property ϕ should hold at each node of the model.

Definition 4 (Satisfaction and Validity)**.** A formula ϕ is:

- *satisfiable* if there is a model $\mathcal{M}$, and an assignment g on $\mathcal{M}$, and a state $w \in M$ such that $\mathcal{M}, g, w \models \phi$
- *globally true* in a model $\mathcal{M}$ under an assignment g if it is satisfiable at all states of the model, i.e., $\mathcal{M}, g, w \models \phi$ for all $w \in M$. We write $\mathcal{M}, g \models \phi$
- *valid* if for all models $\mathcal{M}$ and assignments g, $\mathcal{M}, g \models \phi$.

We can reformulate the frame of Figure 1 (Section 1) within this framework. Prop corresponds to Type, Nom corresponds to Label, and Rel subsums Feat. Note that the functionality of the members of Feat must be enforced separately by axioms. Then, the semantic frame of Figure 1 is a model that satisfies the formula (1) at the element named by l_0. This formula also highlights the crucial role of nominals in this setting.

$$
\begin{aligned}
&l_0 \wedge \textit{motion} \wedge \langle \textsc{agent} \rangle (l_1 \wedge \textit{man}) \wedge \langle \textsc{mover} \rangle l_1 \wedge \langle \textsc{goal} \rangle (l_2 \wedge \textit{house}) \wedge \\
&\quad \langle \textsc{manner} \rangle \textit{walking} \wedge (\exists v\ w. \langle \textsc{path} \rangle (\textit{path} \wedge \langle \textsc{endp} \rangle v) \wedge \\
&\qquad @_{l_2}(\langle \textsc{at-region} \rangle w) \wedge @_v(\langle \textit{part-of} \rangle w)) \quad (1)
\end{aligned}
$$

2.2 *Expressive Power*

According to the satisfaction relation definition, $\downarrow$ and $\exists$ bind node variables without changing the current evaluation node. In addition to $\mathbf{\exists}$, Blackburn and Seligman (1995) introduce another quantifier Σ for which the satisfaction relation also changes the evaluation node:[1]

$$\mathscr{M}, g, w \vDash \Sigma x.\phi \text{ iff } \exists w' \ \mathscr{M}, g^x_{w'}, w' \vDash \phi$$

This defines two independant families of operators: $\downarrow$ and $\exists$, and $\mathbf{\exists}$ and Σ.[2] However, using any two operators of both families (for instance $\downarrow$ and $\mathbf{\exists}$, the "weakest" ones) is expressively equivalent to using the most expressive fragment of the hybrid languages (the full hybrid language).

It is usual to refer to the hybrid languages $\mathscr{H}(\theta_1, \ldots, \theta_n)$ as the extension of the modal language with nominals and the operators $\theta_1, \ldots, \theta_n \in \{\downarrow, @, \mathbf{\exists}, \exists\}$. It is worth noting that even using the simplest binder $\downarrow$ already causes the satisfiability problem for $\mathscr{H}(\downarrow)$ to be undecidable (Areces *et al.* 1999) where the satisfiability problem corresponds to answering the question whether given a formula ϕ, there is a model $\mathscr{M}$, an assignment g and a node w such that $\mathscr{M}, g, w \vDash \phi$.

Nevertheless, there are syntactic restrictions on formulas that make the satisfiability problem decidable. In particular, formulas of the full hybrid language that do not contain the pattern "universal operator scoping over a $\downarrow$ operator scoping over a universal operator" have a decidable satisfiability problem (ten Cate and Franceschet 2005). Such formulas are used in (Kallmeyer *et al.* 2015).

But the formulas we use in the present paper do show this pattern. On the other hand, they do not use the pattern "existential operator scoping over a $\downarrow$ operator scoping over an existential operator". For such formulas, the *validity problem* is shown to be decidable (ten Cate and Franceschet 2005). Although the validity problem for first-order logic is undecidable, this result by itself does not really improve on first-order logic representations. A more promising approach would be to consider semantic restrictions of the underlying class of models.

[1] Blackburn and Seligman (1995) call $\mathbf{\exists}$ the *somewhere* operator, and write it $\Diamond$, and $\mathbf{\forall}$ is the *universal* modality, written $\Box$.

[2] Note that $\downarrow$ can be defined in terms of $\exists$ by $\downarrow x.\phi \equiv \exists x.x \wedge \phi$ and that $\mathbf{\exists}$ can be defined in terms of Σ by $\mathbf{\exists}\phi \equiv \Sigma z.\phi$ with z not occurring in ϕ.

For instance, (Schneider 2007) describes some classes where decidability results hold. As we do not take advantage so far of the Frame Semantics hypothesis that considers attributes to be functional, the class of models with such a semantic restriction is a natural candidate for studying the satisfiability problem. In any case, for every hybrid language, testing a given formula against a given finite model is decidable (Franceschet and de Rijke 2006).

2.3 *Frame Semantics with Quantification*

Since the models we are considering are *semantic frames* instead of arbitrary first-order models, we first present some models in which we consider the sentences (2a), (3a), and (4a). When the model is the frame of Figure 2, we expect (2a) to be true. There indeed is a *kissing* event with AGENT and THEME attributes linking to persons named (represented by the NAME attribute) *John* and *Mary* respectively. Accordingly, we wish to represent the semantics of (2a) by the hybrid logic formula (2b).

On the other hand, (3a) is expected to be false as there is a person named *Paul* who is AGENT of a single *kissing* event whose THEME is a person named *Sue*. The frame of Figure 2 indeed falsify the formula (3b) because we can find a node (namely, i_0) at which *man* holds,[3] but there is no *kissing* node from which we can both reach i_0 through an AGENT relation and, through a THEME relation, a node at which $person \wedge \langle \text{NAME} \rangle Mary$ also holds.

With the object wide scope reading, we also expect (4a) to be false in the frame of Figure 2 because while the person named *Paul* and *Peter* both are AGENT of *kissing* events, these events do not have the same THEME. However, with the subject wide scope reading, (4a) is expected to be true in this frame.

(2) *a.* *John kisses Mary*

b. $\exists(kissing \wedge \langle AGENT \rangle (person \wedge \langle NAME \rangle John) \wedge$
$\langle THEME \rangle (person \wedge \langle NAME \rangle Mary))$

[3] Actually, on Figure 2, only *person* holds at i_0. We can have *man* hold as well with the additional postulate that $(person \wedge \langle \text{NAME} \rangle Paul) \Rightarrow man$, and similarly of each node with a NAME attribute.

(3) a. *Every man kisses Mary*

b. $\forall(\downarrow i.man \Rightarrow \exists(kissing \wedge \langle \text{AGENT} \rangle i \wedge$
$\langle \text{THEME} \rangle (person \wedge \langle \text{NAME} \rangle Mary)))$

(4) a. *Every man kisses some woman*

b. $\forall(\downarrow i.man \Rightarrow \exists(\downarrow i'.woman \wedge$
$\exists(kissing \wedge \langle \text{AGENT} \rangle i \wedge \langle \text{THEME} \rangle i')))$

c. $\exists(\downarrow i.woman \wedge \forall(\downarrow i'.man \Rightarrow$
$\exists(kissing \wedge \langle \text{AGENT} \rangle i' \wedge \langle \text{THEME} \rangle i)))$

(5a) shows how state storing with the $\downarrow$ operator correctly interacts with the @ operator in order to describe node sharing. This sentence is expected to be true (both readings) in the model given by the frame of Figure 3. The frame semantics analysis of bounded motions verbs in (Kallmeyer and Osswald 2013) requires the *motion* to have a GOAL attribute. It is moreover required that the node reached is the same as the one of the entity provided by the *PP*. We express this requirement in the HL formulas (5b) and (5c):

1. by binding to the variable i' a *house* node ,
2. by binding to the variable g a node that is accessible from the *motion* node via the $\langle \text{GOAL} \rangle$ relation,
3. and by stating that i' and g should be the same node, i.e, $g \wedge i'$ should hold.

(5) a. *Every man walked to some house*

b. $\forall(\downarrow i.man \Rightarrow (\exists(\downarrow i'.house \wedge$
$(\exists a\ g.\exists(motion \wedge \langle \text{AGENT} \rangle a \wedge \langle \text{MOVER} \rangle a \wedge \langle \text{GOAL} \rangle g \wedge$
$\langle \text{PATH} \rangle path \wedge \langle \text{MANNER} \rangle walking \wedge @_a i \wedge$
$(\exists r\ v\ w.event \wedge \langle \text{PATH} \rangle (path \wedge \langle \text{ENDP} \rangle v) \wedge$
$@_r(\langle \text{AT-REGION} \rangle w) \wedge @_v(\langle \text{part-of} \rangle w) \wedge @_r(g \wedge i')))))))$

c. $\exists(\downarrow i.house \wedge (\forall(\downarrow i'.man \Rightarrow$
$(\exists a\ g.\exists(motion \wedge \langle \text{AGENT} \rangle a \wedge \langle \text{MOVER} \rangle a \wedge \langle \text{GOAL} \rangle g \wedge$
$\langle \text{PATH} \rangle path \wedge \langle \text{MANNER} \rangle walking \wedge @_a i' \wedge$
$(\exists r\ v\ w.event \wedge \langle \text{PATH} \rangle (path \wedge \langle \text{ENDP} \rangle v) \wedge$
$@_r(\langle \text{AT-REGION} \rangle w) \wedge @_v(\langle \text{part-of} \rangle w) \wedge @_r(g \wedge i)))))))$

Our goal is to compositonally associate each expression in natural language to a HL formula. This logical formula is to be checked against the possible models, and the sentence is true w.r.t. a model $\mathscr{M}$ in case this model satisfies the logical formula. More precisely, given a

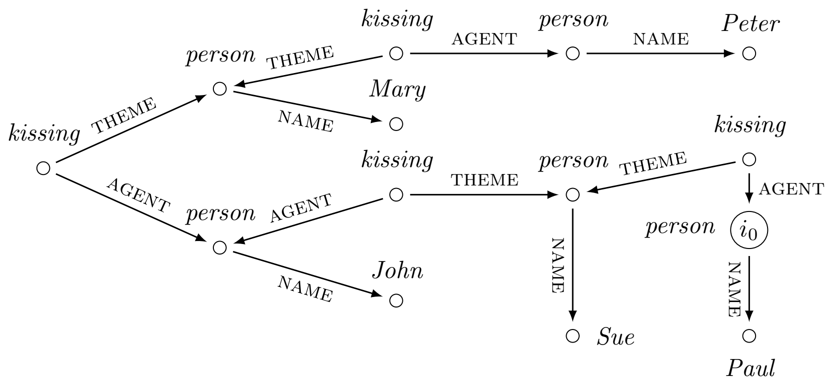

Figure 2: Quantification

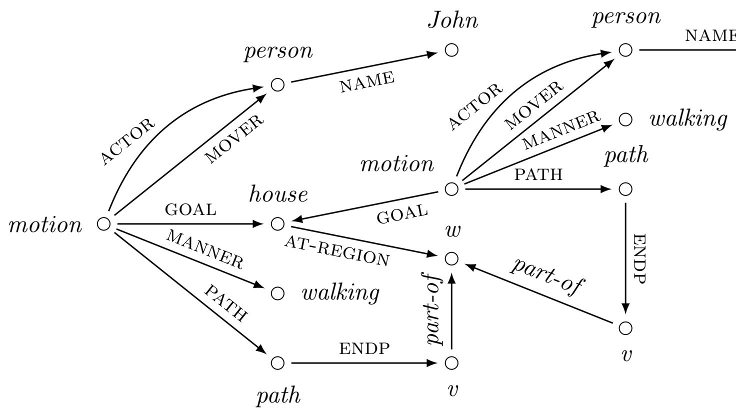

Figure 3: Quantification and node sharing

sentence s and its semantic representation $[\![s]\!]$, we say that s *is true* iff for all assignments g, $\mathscr{M}, g \vDash [\![s]\!]$ (i.e., $[\![s]\!]$ is globally true in $\mathscr{M}$ under any assignment).

3 SYNTAX-SEMANTICS INTERFACE WITH ABSTRACT CATEGORIAL GRAMMARS

In order to exemplify our approach to quantification in frame semantics, we rely on the framework of Abstract Categorial Grammars (ACG) (de Groote 2001). ACGs derive from type-theoretic grammars in the tradition of Lambek (1958), Curry (1961), and Montague (1974). Rather than a grammatical formalism on their own, they provide a framework in which several grammatical formalisms may be encoded (de Groote and Pogodalla 2004). Since our focus is on the semantic modeling of quantification in frame semantics and its compositional account, we provide a Montague-grammar based syntactic modeling that is sufficient four our purpose. Integration of the modeling of scope ambiguity in a TAG encoding (de Groote 2002) for instance would require an embedding into an underspecidied representation language (Bos 1995; Pogodalla 2004; Kallmeyer and Romero 2008) that plays no role in the final interpretation of the logical formula to be interpreted.

3.1 *Abstract Categorial Grammars*

The definition of an ACG is based on a small set of mathematical primitives from type-theory, λ-calculus, and linear logic. These primitives combine via simple composition rules, offering ACGs a good flexibility. In particular, ACGs generate languages of linear λ-terms, which generalizes both string and tree languages. As key feature, ACG provides the user with a direct control over the parse structures of the grammar, the *abstract language*. Such structures are later on interpreted by a morphism, the *lexicon*, to get the concrete *object language*. We call *vocabulary* the *higher-order signature* defining the atomic elements (atomic types and typed constants).

For sake of self-containedness, we remind here the basic definitions of ACGs. We use the standard notations of the typed λ-calculus.

Definition 5 (Types). Let A be a set of atomic types. The set $\mathscr{T}(A)$ of *implicative types* built upon A is defined with the following grammar:

$$\mathscr{T}(A) ::= A|\mathscr{T}(A) \multimap \mathscr{T}(A)|\mathscr{T}(A) \to \mathscr{T}(A)$$

Definition 6 (Higher-Order Signatures). A *higher-order signature* Σ is a triple $\Sigma = \langle A, C, \tau \rangle$ where:

- A is a finite set of atomic types;
- C is a finite set of constants;
- $\tau : C \to \mathscr{T}(A)$ is a function assigning types to constants.

Definition 7 (λ-Terms). Let X be an infinite countable set of λ-variables. The set $\Lambda(\Sigma)$ of λ-terms built upon a higher-order signature $\Sigma = \langle A, C, \tau \rangle$ is inductively defined as follows:

- if $c \in C$ then $c \in \Lambda(\Sigma)$;

- if $x \in X$ then $x \in \Lambda(\Sigma)$;
- if $x \in X$ and $t \in \Lambda(\Sigma)$ and x occurs free in t exactly once, then $\lambda^{o} x.t \in \Lambda(\Sigma)$;
- if $x \in X$ and $t \in \Lambda(\Sigma)$, then $\lambda x.t \in \Lambda(\Sigma)$;
- if $t, u \in \Lambda(\Sigma)$ and the set of free variables of u and t are disjoint then $(t\,u) \in \Lambda(\Sigma)$.

Note there is a linear λ-abstraction (denoted by λ^{o}) and a (usual) intuitionistic λ-abstraction (denoted by λ). There also are the usual notion of α, β, and η conversions (Barendregt 1984).

Definition 8 (Typing Judgment)**.** Given a higher-order signature Σ, the typing rules are given with an inference system whose judgments are of the form: $\Gamma; \Delta \vdash_{\Sigma} t : \alpha$ where:

- Γ is a finite set of non-linear variable typing declaration;
- Δ is a finite set of linear variable typing declaration.

Both Γ and Δ may be empty. If both of them are empty, we usually write $t : \alpha$ (t is of type α) instead of $\vdash_{\Sigma} t : \alpha$. Moreover, we drop the Σ subscript when the context permits. Table 1 gives the typing rules.

Table 1: Typing rules for deriving typing judgments

$$\frac{}{\Gamma; \vdash_{\Sigma} c : \tau(c)}\ \text{(const.)}$$

$$\frac{}{\Gamma; x : \alpha \vdash_{\Sigma} x : \alpha}\ \text{(lin. var.)} \qquad \frac{}{\Gamma, x : \alpha; \vdash_{\Sigma} x : \alpha}\ \text{(var.)}$$

$$\frac{\Gamma; \Delta, x : \alpha \vdash_{\Sigma} t : \beta}{\Gamma; \Delta \vdash_{\Sigma} \lambda^{o} x.t : \alpha \multimap \beta}\ \text{(l. abs.)} \qquad \frac{\Gamma; \Delta_1 \vdash_{\Sigma} t : \alpha \multimap \beta \qquad \Gamma; \Delta_2 \vdash_{\Sigma} u : \alpha}{\Gamma; \Delta_1, \Delta_2 \vdash_{\Sigma} (t\,u) : \beta}\ \text{(l. app.)}$$

$$\frac{\Gamma, x : \alpha; \Delta \vdash_{\Sigma} t : \beta}{\Gamma; \Delta \vdash_{\Sigma} \lambda x.t : \alpha \rightarrow \beta}\ \text{(abs.)} \qquad \frac{\Gamma; \Delta \vdash_{\Sigma} t : \alpha \rightarrow \beta \qquad \Gamma; \vdash_{\Sigma} u : \alpha}{\Gamma; \Delta \vdash_{\Sigma} (t\,u) : \beta}\ \text{(app.)}$$

Definition 9 (Lexicon). Let $\Sigma_1 = \langle A_1, C_1, \tau_1 \rangle$ and $\Sigma_2 = \langle A_2, C_2, \tau_2 \rangle$ be two higher-order signatures, a lexicon $\mathscr{L} = \langle F, G \rangle$ from Σ_1 to Σ_2 is such that:

- $F : A_1 \rightarrow \mathscr{T}(A_2)$. We also note $F : \mathscr{T}(A_1) \rightarrow \mathscr{T}(A_2)$ its homomorphic extension[4];

[4] such that $F(\alpha \multimap \beta) = F(\alpha) \multimap F(\beta)$ and $F(\alpha \rightarrow \beta) = F(\alpha) \rightarrow F(\beta)$

- $G : C_1 \to \Lambda(\Sigma_2)$. We also note $G : \Lambda(\Sigma_1) \to \Lambda(\Sigma_2)$ its homomorphic extension;
- F and G are such that for all $c \in C_1$, $\vdash_{\Sigma_2} G(c) : F(\tau_1(c))$ is provable.

We also use $\mathscr{L}$ instead of F or G.

Definition 10 (Abstract Categorial Grammar and vocabulary). An *abstract categorial grammar* is a quadruple $\mathscr{G} = \langle \Sigma_1, \Sigma_2, \mathscr{L}, S \rangle$ where:

- $\Sigma_1 = \langle A_1, C_1, \tau_1 \rangle$ and $\Sigma_2 = \langle A_2, C_2, \tau_2 \rangle$ are two higher-order signatures. Σ_1 (resp. Σ_2) is called the *abstract vocabulary* (resp. the *object vocabulary*) and $\Lambda(\Sigma_1)$ (resp. $\Lambda(\Sigma_2)$) is the set of *abstract terms* (resp. the set of *object terms*).
- $\mathscr{L} : \Sigma_1 \to \Sigma_2$ is a lexicon.
- $S \in \mathscr{T}(A_1)$ is the *distinguished type* of the grammar.

Given an ACG $\mathscr{G}_{name} = \langle \Sigma_1, \Sigma_2, \mathscr{L}_{name}, S \rangle$, we use the following notational variants for the interpretation of the type α (resp. the term t): $\mathscr{L}_{name}(\alpha) = \beta$, $\mathscr{G}_{name}(\alpha) = \beta$, $\alpha :=_{name} \beta$, and $[\![\alpha]\!]_{name} = \beta$ (resp. $\mathscr{L}_{name}(t) = u$, $\mathscr{G}_{name}(t) = u$, $t :=_{name} u$, and $[\![t]\!]_{name} = u$). The subscript may be omitted if clear from the context.

Definition 11 (Abstract and Object Languages). Given an ACG $\mathscr{G}$, the *abstract language* is defined by

$$\mathscr{A}(\mathscr{G}) = \{t \in \Lambda(\Sigma_1) \mid \vdash_{\Sigma_1} t : S \text{ is derivable}\}$$

The *object language* is defined by

$$\mathscr{O}(\mathscr{G}) = \{u \in \Lambda(\Sigma_2) \mid \exists t \in \mathscr{A}(\mathscr{G}) \text{ s.t. } u = \mathscr{L}(t)\}$$

3.2 *The Syntax-Semantics Interface as ACG Composition*

The lexicon defines the way structures are interpreted. It plays a crucial role in the way ACG models the syntax-semantics interface. The basic idea is to have a same (abstract) structure interpreted either as a surface form (e.g., a string) or as a meaning form (e.g., a logical formula). This boils down to having two interpretations that share the same abstract vocabulary, hence mapping a single structure into two different ones. This composition is illustrated by $\mathscr{G}_{form}$ and $\mathscr{G}_{meaning}$ sharing the $\Sigma_{abstract}$ vocabulary in Figure 4.

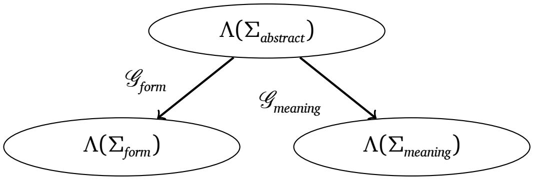

Figure 4: ACG composition the syntax-semantic interface

4 TYPE-THEORETIC SEMANTICS WITH FRAMES

We now provide the type-theoretic syntax-semantics interface allowing for a compositional building of the meanings. We use the architecture described in Figure 4. As we are concerned in this article with semantic modeling and quantification rather than with parsing, we use higher-order types for quantified noun-phrases.

All the following examples can be run and tested with the ACG toolkit[5] and the companion example files.[6]

4.1 *The ACG of Surface Forms*

At the abstract level, we use the signature defined with the type assignment of Table 2. It makes use of the usual syntactic types: *NP*, *S*, *N*, and *PP*. Note that following the usual type-logical approach, determiners have a higher-order type.

Table 2: $\Sigma_{abstract}$ type assignement

John, Mary	$: NP$	kisses	$: NP \rightarrow NP \rightarrow S$
man, woman, house	$: N$	every, some	$: N \rightarrow (NP \rightarrow S) \rightarrow S$
to, into	$: NP \rightarrow PP$	walked	$: PP \rightarrow NP \rightarrow S$

The object vocabulary of surface forms uses the standard modeling of strings as lambda terms. It is build on Σ_{form} that contains a single atomic type o, and the type σ (for strings) is defined by $\sigma \triangleq o \multimap o$. The concatenation is then defined as functional composition by $\cdot + \cdot = \lambda f\ g.\lambda z.f\ (g\ z) : \sigma \multimap \sigma \multimap \sigma$. It is associative, and it admits the identity function $\epsilon \triangleq \lambda^o x.x : \sigma$ as neutral element. Σ_{form} also contains the constants *John*, *Mary*, *kisses*, *every*, *man*... of type σ.

[5] ACGtk can be downloaded and installed from `http://www.loria.fr/equipes/calligramme/acg/#Software`.

[6] These files are available at `https://www.dropbox.com/s/r08r8ym2dvadxtr/acg-examples.zip?dl=0`.

The ACG $\mathscr{G}_{form}$ is then defined using the interpretations given in Table 3. The terms defined in Equation (6) correspond to the syntactic derivations of the sentence we want to provide a semantic representation to. Their surface forms are given by Equations (7)–(12).

$$\begin{aligned}
u_{2b} &= \text{kisses Mary John} \\
u_{3b} &= (\text{every man})\ (\lambda x.\text{kisses Mary } x) \\
u_{4b} &= (\text{every man})\ (\lambda x.(\text{some woman})\ (\lambda y.\text{kisses } y\ x)) \\
u_{4c} &= (\text{some woman})\ (\lambda y.(\text{every man})\ (\lambda x.\text{kisses } y\ x)) \\
u_{5b} &= (\text{every man})\ (\lambda x.(\text{some house})(\lambda y.\text{walked (to } y)\ x)) \\
u_{5c} &= (\text{some house})(\lambda y.(\text{every man})\ (\lambda x.\text{walked (to } y)\ x))
\end{aligned} \tag{6}$$

$$u_{2b} :=_{form} \textit{John} + \textit{kisses} + \textit{Mary} \tag{7}$$

$$u_{3b} :=_{form} \textit{every} + \textit{man} + \textit{kisses} + \textit{Mary} \tag{8}$$

$$u_{4b} :=_{form} \textit{every} + \textit{man} + \textit{kisses} + \textit{some} + \textit{woman} \tag{9}$$

$$u_{4c} :=_{form} \textit{every} + \textit{man} + \textit{kisses} + \textit{some} + \textit{woman} \tag{10}$$

$$u_{5b} :=_{form} \textit{every} + \textit{man} + \textit{walked} + \textit{to} + \textit{some} + \textit{house} \tag{11}$$

$$u_{5c} :=_{form} \textit{every} + \textit{man} + \textit{walked} + \textit{to} + \textit{some} + \textit{house} \tag{12}$$

John $:=_{form}$ *John*	Mary $:=_{form}$ *Mary*
man $:=_{form}$ *man*	woman $:=_{form}$ *woman*
house $:=_{form}$ *house*	
to $:=_{form}$ $\lambda n.\textit{to} + n$	into $:=_{form}$ $\lambda n.\textit{into} + n$
every $:=_{form}$ $\lambda n\ P.P\ (\textit{every} + n)$	some $:=_{form}$ $\lambda n\ P.P\ (\textit{some} + n)$
kisses $:=_{form}$ $\lambda o\ s.s + \textit{kissed} + o$	walked $:=_{form}$ $\lambda p\ s.s + \textit{walked} + p$

Table 3: $\mathscr{G}_{form}$ interpretation of the abstract atomic types and constants

4.2 *The ACG of Meaning Representations*

Accordingly to the ACG architecture of Figure 4, the syntax-semantics interface relies on sharing the abstract language of the two ACGs responsible for the surface interpretation on the one hand and for the semantic interpretation on the other hand. The abstract vocabulary we use is $\Sigma_{abstract}$, defined in the previous section.

Our goal is to associate every sentence with a hybrid-logical formula. It's important to note that we are not, at least in this work, interested in higher-order hybrid logic, not even first-order hybrid logic.

The binders and quantifiers we use only bind node variables, and not entities or higher-order predicates. So it should not be confused with quantified hybrid logic (QHL) (Egly and Fermüller 2002). We do not directly adopt either the Hybrid Type Theory (HTT) proposed in (Areces *et al.* 2011, 2014). Contrary to what could be expected from (Gallin 1975) type theory of higher-order modal logic, (Areces *et al.* 2014) do not use a specific type s to denote nodes (or worlds) and nominals are typed t as propositions.

We do introduce a specific type s for nominals, so that the set of atomic types of $\Sigma_{meaning}$ is $\{s, t\}$. We also introduce a coercion operator $\# : s \multimap t$ in order to use nominal into formulas. This ensures we only build formulas of Forms. Table 4 shows the semantic constants we use, including logical operators and quantifiers.

Table 4: Constant terms of the semantic language

event, *kissing*, *motion*, *person*, *John*, *Mary*, ...	$: t$
$\langle$AGENT$\rangle$, $\langle$THEME$\rangle$, $\langle$MOVER$\rangle$, $\langle$*part-of*$\rangle$, ...	$: t \multimap t$
$\# :$	$s \multimap t$
$\wedge, \Rightarrow$	$: t \to t \to t$
$@$	$: s \multimap t \multimap t$
$\boldsymbol{\exists}, \boldsymbol{\forall}$	$: t \to t$
$\downarrow, \exists$	$: (t \to t) \multimap t$

We can now define $\mathscr{G}_{meaning}$ using the interpretations of the atomic types of the constants of Table 5. We follow (Kallmeyer and Osswald 2013) in the semantics and meaning decomposition of motion verbs.

What the ACG framework does not express, though, are the lexical or meaning postulates that can be added to the logical theory. Such postulates are additional constraints that any model should also satisfy and that do not depend on the actual semantic representation that is being build. They include for instance the representation of the ontology of propositions (types, in the frame semantics terminology) such as: *man* $\Rightarrow$ *person*, or any standard modal-logical axiom such as $\Box(p \Rightarrow q) \Rightarrow (\Box p \Rightarrow \Box q)$.

Then the following equalities hold, where t_{2b} is the term in (2b), t_{3b} is the term in (3b), etc., such that every nominal variable is pre-

S,NP,N	$:=_{meaning}$	t	$PP :=_{meaning} t \to t$
John	$:=_{meaning}$	$John$	
Mary	$:=_{meaning}$	$Mary$	
man	$:=_{meaning}$	man	
woman	$:=_{meaning}$	$woman$	
house	$:=_{meaning}$	$house$	
some	$:=_{meaning}$	$\lambda P\ Q.\exists(\downarrow i.P \wedge (Q\ i))$	
every	$:=_{meaning}$	$\lambda P\ Q.\forall(\downarrow i.P \Rightarrow (Q\ i))$	
kisses	$:=_{meaning}$	$\lambda o\ s.\exists(kissing \wedge \langle \text{AGENT}\rangle s \wedge \langle \text{THEME}\rangle o)$	
walked	$:=_{meaning}$	$\lambda pp\ s.\exists a\ g.\exists(motion \wedge \langle \text{AGENT}\rangle a \wedge \langle \text{MOVER}\rangle a \wedge \langle \text{GOAL}\rangle g$ $\wedge \langle \text{PATH}\rangle path \wedge \langle \text{MANNER}\rangle walking \wedge @_a s \wedge (pp\ g))$	
to	$:=_{meaning}$	$\lambda n\ g.\exists r\ v\ w.event \wedge \langle \text{PATH}\rangle(path \wedge \langle \text{ENDP}\rangle v) \wedge$ $@_r \langle \text{AT-REGION}\rangle w \wedge @_v \langle part\text{-}of\rangle w \wedge @_r (g \wedge n)$	
into	$:=_{meaning}$	$\lambda n\ g.\exists r\ v\ w.event \wedge \langle \text{PATH}\rangle(path \wedge \langle \text{ENDP}\rangle v) \wedge$ $@_r \langle \text{IN-REGION}\rangle w \wedge @_v \langle part\text{-}of\rangle w \wedge @_r (g \wedge n)$	

Table 5: Semantic interpretation of the constants of $\Sigma_{abstract}$

ceded by the # coercion operator:

$$[\![\text{kisses Mary John}]\!] = t_{2b} \tag{13}$$

$$[\![(\text{every man})\ (\lambda x.\text{kisses Mary } x)]\!] = t_{3b} \tag{14}$$

$$[\![(\text{every man})\ (\lambda x.(\text{some woman})\ (\lambda y.\text{kisses } y\ x))]\!] = t_{4b} \tag{15}$$

$$[\![(\text{some woman})\ (\lambda y.(\text{every man})\ (\lambda x.\text{kisses } y\ x))]\!] = t_{4c} \tag{16}$$

Table 5 shows the interaction of the storing operator with path equalities. It compositionally derives from the verb and the preposition semantic interpretations. In the verb semantics, the path equalities specify that the MOVER and the AGENT attributes of the event are the same, and that the information provided by the *pp* argument should hold for the GOAL g. In its semantics, the preposition contributes on the one hand to the main event (as the *event* proposition is evaluated at the current state) and on the other hand by specifying that the g state (meant to be the target node of the verb that the proposition modifies, here the target of the GOAL attribute) should be identified to the n argument (the noun phrase which is argument of the preposition). This leads to the interpretations (5b) and (5c) of (5a) given in (17) and (18).

$$[\![(\text{every man})\ (\lambda x.(\text{some house})(\lambda y.\text{walked (to } y)\ x))]\!] = t_{5b} \tag{17}$$

$$[\![(\text{some house})(\lambda y.(\text{every man})\ (\lambda x.\text{walked (to } y)\ x))]\!] = t_{5c} \tag{18}$$

5 TYPE COERCION AS EXISTENTIAL QUANTIFICATION

We now have two ingredients at our disposal: the decomposition of the lexical semantics offered by frame semantics, and the power of binding states. We illustrate how to combine them in order to model semantic coercion. (19) shows how a predicate can take another event predicate as argument. On the other hand, (20) shows that the same predicate can take a noun phrase as argument. In the latter case, it however conveys the meaning that the entity referred to by the noun phrase should be part of some event. It is even the case that if this event is not salient in the context, it can be inferred from the lexicon, for instance using the *qualia* structure and the telic quale as defined by the Generative Lexicon (Pustejovsky 1998), or a subclass of the $\mathbf{S}_2$ lexical function in the framework of the Explanatory and Combinatorial Lexicology (Mel'čuk *et al.* 1995; Polguère 2003).

(19) *John began to read a book*

(20) *John began a book*

We first model (19). The assumed syntactic construction are given by the extension of the $\Sigma_{abstract}$ signature of Table 6 and by its interpretation by $\mathcal{G}_{form}$ of Table 7.

Table 6: Extension of $\Sigma_{abstract}$

begin_1	$: S_{inf} \multimap NP \multimap S$
begin_2	$: NP \multimap NP \multimap S$
to read	$: NP \multimap S_{inf}$

Table 7: Interpretation of types and constants of $\Sigma_{abstract}$ by $\mathcal{G}_{form}$

S_{inf}	$:=_{form} \sigma$
begin_1	$:=_{form} \lambda^o c\ s.s + began + c$
$\mathsf{begin}'_2, \mathsf{begin}_2$	$:=_{form} \lambda^o o\ s.s + began + o$
to read	$:=_{form} \lambda^o o.to + read + o$

Semantically, the idea is that events are structured (Moens and Steedman 1988). We in particular consider the structures required by aspectual predicates such as *begin* as in (Pustejovsky and Bouillon 1995). We structure the events with the notion of *transition* that has an ANTE attribute and a POST attribute (see Figure 5). When an event has begun, it is set as value of this attribute. This is what the interpretation of begin_1 in Table 8 states. This interpretation also requires the event argument to be a process (*proc*) or an accomplishment (*acc*) (Im and Lee 2015).

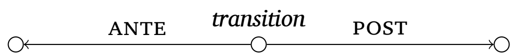

Figure 5: Event structure

Table 8: Interpretation of types and constants of $\Sigma_{abstract}$ by $\mathscr{G}_{meaning}$

S_{inf}	$:=_{meaning} t \multimap t$
begin_1	$:=_{meaning} \lambda^o c\ s.\exists(\mathit{transition} \wedge \langle \textsc{post}\rangle(\mathit{proc} \vee \mathit{acc} \wedge (c\ s)))$
begin'_2	$:=_{meaning} \lambda^o o\ s.\exists(\mathit{transition} \wedge \langle \textsc{post}\rangle(\mathit{proc} \vee \mathit{acc} \wedge \langle \textsc{agent}\rangle s \wedge \langle \textsc{ug}\rangle o))$
begin_2	$:=_{meaning} \lambda^o o\ s.\exists(\mathit{transition} \wedge \langle \textsc{post}\rangle (\downarrow s.\mathit{proc} \vee \mathit{acc} \wedge \langle \textsc{agent}\rangle s \wedge \langle \textsc{ug}\rangle(o \wedge \langle \mathit{proto}\ \rangle\langle \mathit{e\text{-}q}\ \rangle(\downarrow s'.@_s\langle \mathit{proto}\ \rangle s'))))$
$\mathsf{to\ read}$	$:=_{meaning} \lambda o\ s.\mathit{reading} \wedge \langle \textsc{agent}\rangle s \wedge \langle \textsc{theme}\rangle o$

$$(\mathsf{a\ book})\ (\lambda^o y.\mathsf{begin}_1\ (\mathsf{to\ read}\ y)\ \mathsf{John}) :=_{forms} \mathit{John + began + to + read + a + book} \tag{21}$$

$$(\mathsf{a\ book})\ (\lambda^o y.\mathsf{begin}_1\ (\mathsf{to\ read}\ y)\ \mathsf{John}) :=_{meaning} \exists(\downarrow i.\mathit{book} \wedge (\exists(\mathit{transition} \wedge\langle \textsc{post}\rangle(\mathit{reading} \wedge\langle \textsc{agent}\rangle(\mathit{person} \wedge \langle \textsc{name}\rangle \mathit{John}) \wedge\langle \textsc{theme}\rangle(\#\, i))))) \tag{22}$$

With the provided ACGs, we can then compute the semantic interpretation of the syntactic derivation associated to (19). Equation (21) shows the syntactic derivation indeed corresponds to the sentence, and Equation (22) shows its semantic interpretation. In order to be true, the model should have a node i where *book* holds, and a node where *transition* holds and from which there is a ⟨POST⟩⟨THEME⟩ path to i.

It is actually this path that we require to exist in the semantic recipe for *begin* when used with a direct object. This requirement appears in the interpretation of begin'_2 as given by Table 8 by specifying that the POST attribute itself has an UG that should target the direct object. This interpretation also accounts for the following constraints (Pustejovsky and Bouillon 1995): the subject of *begin* also is the agent of the argument event, and the latter is either a process or an accomplishment. Equation (23) and (24) show the achieved effects from the derivation of (20).

$$\text{(a book)}\ (\lambda^o y.\text{begin}'_2\ y\ \text{John}) :=_{forms} John + began + a + book \qquad (23)$$

$$\begin{aligned}\text{(a book)}\ (\lambda^o y.\text{begin}'_2\ y\ \text{John}) :=_{meaning}& \\ \exists(\downarrow i.book \wedge \exists(transition& \\ \wedge\langle\text{POST}\rangle(proc \vee acc \wedge \langle\text{AGENT}\rangle(person& \\ \wedge\langle\text{NAME}\rangle John) \wedge \langle\text{UG}\rangle(\# i))))&\end{aligned} \qquad (24)$$

It is not specified, though, what kind of event it is: *reading*, *writing*, etc. The latter lexically depends on the object. We want to model this dependency by adding lexically determined conditions on the possible models that make the formula true. We already met conditions in the form of meaning postulates, such as ($person \wedge \langle\text{NAME}\rangle Paul) \Rightarrow man$. The conditions we introduce now are different and also make use of another feature of hybrid logic that we have not used so far: actual nominals, and not only state variables. These nominals encode lexical properties of the entities to be used in the meaning representation of the lexical items.

So we introduce the nominals $i_{\text{book}}, i_{\text{reading}}, i_{\text{writing}}, i_{\text{translating}} \ldots$ corresponding to the proposition (or types, in the frame semantics terminology) *book*, *reading*, *writing*, *translating*...For each of these pairs, the following schema holds:

$$(\langle proto\ \rangle i_{\text{p}}) \Rightarrow p \qquad (25)$$

If we additionally require that each node has a *proto* attribute, each node in a frame should be associated with a prototypical node named by a nominal, and the proposition that hold at the former can be inferred from the latter.

We also encode that the i_{book} node is related through the *e-q* (event quale) relation to some event nominals, requiring the postulates of (26) to hold as well (see Figure 6).

$$\begin{aligned} &@_{i_{\text{book}}}\langle e\text{-}q\ \rangle i_{\text{reading}} \\ &@_{i_{\text{book}}}\langle e\text{-}q\ \rangle i_{\text{writing}} \\ &\ldots \end{aligned} \qquad (26)$$

Remark (Nominals as prototypical entities). It is very important that the postulates of (26) use nominals rather than properties. Stating these postulates directly with propositions, such as $\mathit{book} \wedge \langle \mathit{e\text{-}q} \rangle \mathit{reading}$, $\mathit{book} \wedge \langle \mathit{e\text{-}q} \rangle \mathit{wrting}$, etc., would amount to require any node where *book* holds to relate to every (quale) events with an $\langle \mathit{e\text{-}q} \rangle$ relation. These events would then be part of the model even if no linguistic element, such as *begin*, triggers them.

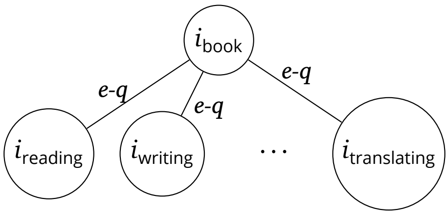

Figure 6: *Qualia* values associated to i_{book}

We can now state the following condition on a node s that has o as undergoer: when looking at the event quale of the prototype of o, if we call this event quale s', then s' should be a prototype of s. Equation 27 states this condition in hybrid logic terms. It is the formula used for the interpretations of begin_2 in Table 8. It can be paraphrased as follows: if s is a state that has o as undergoer, we set s' to be a quale event associated to o, via a prototype of o. For instance, if *book* holds at o, $o \wedge \langle \mathit{proto} \rangle$ is i_{book}. Then s' is one of the i_{reading}, i_{writing}, etc. Say it is i_{reading}. $@_s \langle \mathit{proto} \rangle s'$ finally ensures s prototype is i_{reading}. Together with (25), we thus have that *reading* holds at s.

$$\downarrow s.\langle \textsc{ug} \rangle (o \wedge \langle \mathit{proto} \rangle \langle \mathit{e\text{-}q} \rangle (\downarrow s'.@_s \langle \mathit{proto} \rangle s')) \tag{27}$$

Thus, as Equations (28) and (29) show, together with the postulates (25) and (26), we have the semantic coercion of the object (here a book) to its associated possible telic quales through the *prototype* relation.

$$(\text{a book})\ (\lambda^o y.\text{begin}_2\ y\ \text{John}) :=_{\mathit{forms}} \mathit{John} + \mathit{began} + \mathit{a} + \mathit{book} \tag{28}$$

$$\begin{aligned} &(\text{a book})\ (\lambda^o y.\text{begin}_2\ y\ \text{John}) :=_{\mathit{meaning}} \\ &\exists(\downarrow i.\mathit{book} \wedge \exists(\mathit{transition} \\ &\wedge \langle \textsc{post} \rangle (\downarrow s.\mathit{proc} \vee \mathit{acc} \wedge \langle \textsc{agent} \rangle (\mathit{person} \\ &\wedge \langle \textsc{name} \rangle \mathit{John}) \wedge \langle \textsc{ug} \rangle (\#\, i \wedge \langle \mathit{proto} \rangle \langle \mathit{e\text{-}q} \rangle (\downarrow s'.@_s \#\, s'))))) \end{aligned} \tag{29}$$

While accounting for the lexicon knowledge, this approach make no use of a possible specific context where it is *not* required to use the lexical information. For instance, (30) does not make sense without any context, as *ball* does not come with a telic quale.

(30) *John began his ball*

However, in a context that John was asked to paint a ball, for instance, this information could be used to correctly interpret (30). Such an account could possibly be provided by making use of a selection operator in some context, akin to the one proposed in (de Groote 2006; Lebedeva 2012). The introduction of node binder should indeed allow us to propose such an approach to a continuation based approach to event context. In particular we may *presuppose* in the semantics of begin_2 the path condition that apply to the object. If this property is already satisfied (for instance by a painting event), nothing more happens than the retrieval of this event. Otherwise, the telic quale of the object might be projected, possibly resulting in a failure if no prototypical telic quale is available.

6 CONCLUSION AND PERSPECTIVES

We used hybrid logic as a means to integrate logical operators with frame semantics. We illustrated the approach with the modeling of quantifier scopes. We embedded the proposed semantic representation within the Abstract Categorial Grammar framework in order to show how to compositionally derive different quantifier scope readings. We also showed how the key ingredients of hybrid logic, nominals and binders, can be used to model semantic coercion, such as the one induced by the *begin* predicate.

Binding nodes also offers the possibility of using continuation semantics in order to model a dynamic reference to events. In the particular case of semantic coercion, we plan to study how to integrate the model we proposed with a representation of the context. The projection of the telic quale of some (object) entity would then depend on the availability of some previously introduced events.

We also plan to take advantage of the semantic structuring induced by frame semantics to account for dot type objects representation and co-predication.

Finally, we plan to investigate the computational properties of the framework we propose with respect to the hybrid inferential systems (Egly and Fermüller 2002) and the specific properties induced by the frame models we consider, typically the functionality of the attribute relations (Schneider 2007).

REFERENCES

Carlos ARECES, Patrick BLACKBURN, Antonia HUERTAS, and María MANZANO (2011), Hybrid Type Theory: A Quartet in Four Movements, *Principia*, 15(2):225–247, DOI: 10.5007/1808-1711.2011v15n2p225.

Carlos ARECES, Patrick BLACKBURN, Antonia HUERTAS, and María MANZANO (2014), Completeness in Hybrid Type Theory, *Journal of Philosophical Logic*, 43(2-3):209–238, ISSN 0022-3611, DOI: 10.1007/s10992-012-9260-4.

Carlos ARECES, Patrick BLACKBURN, and Maarten MARX (1999), A Road-Map on Complexity for Hybrid Logics, in Jörg FLUM and Mario RODRIGUEZ-ARTALEJO, editors, *Computer Science Logic: 13th International Workshop, CSL'99 8th Annual Conference of the EACSL Madrid, Spain, September 20–25, 1999 Proceedings*, pp. 307–321, Springer Berlin Heidelberg, DOI: 10.1007/3-540-48168-0_22.

Carlos ARECES and Balder TEN CATE (2007), Hybrid logics, in Patrick BLACKBURN, Johan Van BENTHEM, and Frank WOLTER, editors, *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*, chapter 14, pp. 821–868, Elsevier, DOI: 10.1016/S1570-2464(07)80017-6.

Jason BALDRIDGE and Geert-Jan KRUIJFF (2002), Coupling CCG and Hybrid Logic Dependency Semantics, in *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pp. 319–326, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, DOI: 10.3115/1073083.1073137, ACL anthology: P02-1041.

Hendrik Pieter BARENDREGT (1984), *The lambda calculus*, volume 103 of *Studies in Logic and the Foundations of Mathematics*, North-Holland.

Lawrence BARSALOU (1992), Frames, concepts, and conceptual fields, in Adrienne LEHRER and Eva Feder KITTEY, editors, *Frames, fields, and contrasts: New essays in semantic and lexical organization*, pp. 21–74, Lawrence Erlbaum Associates, Hillsdale.

Patrick BLACKBURN (1993), Modal Logic and Attribute Value Structures, in Maarten DE RIJKE, editor, *Diamonds and Defaults*, volume 229 of *Synthese Library*, pp. 19–65, Springer Netherlands, ISBN 978-90-481-4286-6, DOI: 10.1007/978-94-015-8242-1_2.

Patrick BLACKBURN and Jerry SELIGMAN (1995), Hybrid languages, *Journal of Logic, Language and Information*, 4(3):251–272, DOI: 10.1007/BF01049415.

Johan Bos (1995), Predicate Logic Unplugged, in *Proceedings of the Tenth Amsterdam Colloquium*, http://www.let.rug.nl/bos/pubs/Bos1996AmCo.pdf.

Haskell Brooks Curry (1961), Some Logical Aspects of Grammatical Structure, in Roman Jakobson, editor, *Structure of Language and its Mathematical Aspects: Proceedings of the Twelfth Symposium in Applied Mathematics*, pp. 56–68, American Mathematical Society.

Philippe de Groote (2001), Towards Abstract Categorial Grammars, in *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference*, pp. 148–155, ACL anthology: P01-1033.

Philippe de Groote (2002), Tree-Adjoining Grammars as Abstract Categorial Grammars, in *Proceedings of the Sixth International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+6)*, pp. 145–150, Università di Venezia, http://www.loria.fr/equipes/calligramme/acg/publications/2002-tag+6.pdf.

Philippe de Groote (2006), Towards a Montagovian account of dynamics, in Masayuki Gibson and Jonathan Howell, editors, *Proceedings of Semantics and Linguistic Theory (SALT) 16*, DOI: 10.3765/salt.v16i0.2952.

Philippe de Groote and Sylvain Pogodalla (2004), On the expressive power of Abstract Categorial Grammars: Representing context-free formalisms, *Journal of Logic, Language and Information*, 13(4):421–438, DOI: 10.1007/s10849-004-2114-x, HAL open archive: inria-00112956.

Uwe Egly and Chritian G. Fermüller, editors (2002), *Tableaux for Quantified Hybrid Logic*, Springer Berlin Heidelberg, Berlin, Heidelberg, DOI: 10.1007/3-540-45616-3_4.

Charles J. Fillmore (1977), The case for case reopened, in Peter Cole and Jerrold M. Sadock, editors, *Grammatical Relations*, volume 8 of *Syntax and Semantics*, pp. 59–81, Academic Press, New York.

Massimo Franceschet and Maarten de Rijke (2006), Model checking hybrid logics (with an application to semistructured data), *Journal of Applied Logic*, 4(3):279–304, DOI: 10.1016/j.jal.2005.06.010.

Daniel Gallin (1975), *Intensional and Higher-Order Modal Logic*, North-Holland.

Thomas Gamerschlag, Doris Gerland, Rainer Osswald, and Wiebke Petersen, editors (2014), *Frames and Concept Types*, volume 94 of *Studies in Linguistics and Philosophy*, Springer International Publishing, DOI: 10.1007/978-3-319-01541-5.

Seohyun Im and Chunngmin Lee (2015), A Developed Analysis of Type Coercion Based on Type Theory and Conventionality, in Robin Cooper and Christian Retoré, editors, *Type Theory and Lexical Semantics*, ESSLLI 2015, Barcelona, Spain, http://www.lirmm.fr/tytles/Articles/Im.pdf.

Laura KALLMEYER and Rainer OSSWALD (2013), Syntax-Driven Semantic Frame Composition in Lexicalized Tree Adjoining Grammars, *Journal of Language Modelling*, 1(2):267–330, DOI: 10.15398/jlm.v1i2.61.

Laura KALLMEYER, Rainer OSSWALD, and Sylvain POGODALLA (2015), Progression and Iteration in Event Semantics - An LTAG Analysis Using Hybrid Logic and Frame Semantics, in *Colloque de Syntaxe et Sémantique à Paris (CSSP 2015)*, HAL open archive: hal-01184872.

Laura KALLMEYER and Frank RICHTER (2014), Quantifiers in Frame Semantics, in Glyn MORRILL, Reinhard MUSKENS, Rainer OSSWALD, and Frank RICHTER, editors, *Formal Grammar*, number 8612 in Lecture Notes in Computer Science, pp. 69–85, Springer, DOI: 10.1007/978-3-662-44121-3_5.

Laura KALLMEYER and Maribel ROMERO (2008), Scope and Situation Binding for LTAG, *Research on Language and Computation*, 6(1):3–52, DOI: 10.1007/s11168-008-9046-6.

Joachim LAMBEK (1958), The Mathematics of Sentence Structure, *American Mathematical Monthly*, 65(3):154–170.

Ekaterina LEBEDEVA (2012), *Expression de la dynamique du discours à l'aide de continuations*, Ph.D. thesis, Université de Lorraine, in English.

Sebastian LÖBNER (2014), Evidence for frames from human language, in Gamerschlag *et al.* (2014), pp. 23–67, DOI: 10.1007/978-3-319-01541-5_2.

Igor A. MEL'ČUK, André CLAS, and Alain POLGUÈRE (1995), *Introduction à la lexicologie explicative et combinatoire*, Éditions Duculot, Louvain-la-Neuve.

Marc MOENS and Mark STEEDMAN (1988), Temporal Ontology and Temporal Reference, *Computational Linguistics*, 14(2):15–28, ACL anthology: J88-2003.

Richard MONTAGUE (1974), The Proper Treatment of Quantification in Ordinary English, in *Formal Philosophy: Selected Papers of Richard Montague*, Yale University Press, re-edited in "Formal Semantics: The Essential Readings", Paul Portner and Barbara H. Partee, editors. Blackwell Publishers, 2002.

Rheinard MUSKENS (2013), Data Semantics and Linguistic Semantics, in Maria ALONI, Michael FRANKE, and Floris ROELOFSEN, editors, *The dynamic, inquisitive, and visionary life of ϕ, $?\phi$, and $\Diamond\phi$*, chapter 24, pp. 175–183, Pumbo.nl, http://www.illc.uva.nl/Festschrift-JMF/papers/23_Muskens.pdf.

Rainer OSSWALD and Robert D. VAN VALIN, Jr. (2014), FrameNet, Frame Structure, and the Syntax-Semantics Interface, in Gamerschlag *et al.* (2014), chapter 6, pp. 125–156, DOI: 10.1007/978-3-319-01541-5_6.

Wiebke PETERSEN (2007), Representation of Concepts as Frames, *The Baltic International Yearbook of Cognition, Logic and Communication*, 2:151–170, http://user.phil-fak.uni-duesseldorf.de/~petersen/paper/Petersen2007_proof.pdf.

Sylvain POGODALLA (2004), Computing Semantic Representation: Towards ACG Abstract Terms as Derivation Trees, in *Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms - TAG+7*, pp. 64–71, Vancouver, BC, Canada, HAL open archive: inria-00107768.

Alain POLGUÈRE (2003), *Lexicologie et sémantique lexicale*, Les Presses de l'Université de Montréal.

James PUSTEJOVSKY (1998), *The Generative Lexicon*, MIT Press.

James PUSTEJOVSKY and Pierrette BOUILLON (1995), Aspectual Coercion and Logical Polysemy, *Journal of Semantics*, 12(2):133–162, DOI: 10.1093/jos/12.2.133.

Thomas SCHNEIDER (2007), *The Complexity of Hybrid Logics over Restricted Classes of Frames*, Ph.D. thesis, University of Jena, Germany, http://www.cs.man.ac.uk/%7Eschneidt/publ/sch07_phd.pdf.

Balder TEN CATE and Massimo FRANCESCHET (2005), On the Complexity of Hybrid Logics with Binders, in Luke ONG, editor, *Computer Science Logic: 19th International Workshop, CSL 2005, 14th Annual Conference of the EACSL, Oxford, UK, August 22-25, 2005. Proceedings*, pp. 339–354, Springer Berlin Heidelberg, DOI: 10.1007/11538363_24.