

# Embodied construction grammar as layered modal languages

Anders Søgaard

Center for Language Technology

Njalsgade 80

DK-2300 Copenhagen

anders@cst.dk

## Abstract

One can think of scalability in terms of complexity or granularity, but in both cases, modal languages seem to be of interest: Modal languages are robustly decidable, and they encode in a natural way the idea of refinement by layering, which implies scalability in terms of granularity. The contribution of this paper is to introduce the general technique of translating problems of scalable natural language understanding into layered modal languages; for illustration, a translation is sketched for Embodied Construction Grammar (ECG). On the basis of this translation, an upper bound on the complexity of ECG is established: the universal recognition problem of ECG is solvable in EXPTIME (if some dynamic resetting is assumed). If the use of the **evokes**-operator is polynomially bound, the recognition problem turns NP-complete.

## 1 Introduction

The purpose of this paper is to demonstrate that problems of scalable natural understanding translate into layered modal languages. Since the properties of such languages are well-studied, such a translation is an important achievement. In this paper, this is illustrated by providing a translation from ECG into a layered extension of modal logic with nominals from

which complexity results can be derived. On the general note, ECG is a step in the direction of a formal specification and interpretation of various theories proposed in the cognitive linguistics community. Many questions pop up with the introduction of such a theory, for instance: Does ECG faithfully reflect the intuitions of the original proposals, e.g. mental space theory (see below)? Can it be implemented on existing grammar engineering platforms? What is its computational complexity? Etc. Some of these questions are answered in this paper, some are not; in particular, we do not examine the “faithfulness” of ECG to the original sources. Our purpose here is to investigate its computational properties. This purpose is, however, in the author’s view, secondary to the introduction of layered modal languages as a method for scalable natural language understanding. Let us first begin with some technicalities, a simple example of a layered modal language:

**Example 1.1.** Consider a propositional modal logic ( $\mathcal{L}$ ) of two domains  $\mathbb{D}$  and  $\mathbb{E}$  and  $\mathbb{D} \cap \mathbb{E} = \emptyset$ .  $\mathbb{D}$  and  $\mathbb{E}$  are nodes in a Kripke model  $\langle \mathcal{W}, \mathcal{R}, \mathcal{V} \rangle$  (worlds, relations, valuation), s.t.  $\mathcal{W} \subseteq \mathbb{D} \cup \mathbb{E}$ . Let  $\delta$  and  $\epsilon$  be propositions. Say  $\delta$  in the model denotes a subset of  $\mathbb{D}$ , i.e.  $\llbracket \delta \rrbracket \subseteq \mathbb{D}$ , and  $\epsilon$  denotes a subset of  $\mathbb{E}$ . Consider the axioms  $\delta \vee \epsilon$ ,  $\neg(\delta \wedge \epsilon)$ , and  $\langle \tau \rangle \phi \rightarrow \delta \wedge \langle \tau \rangle (\epsilon \wedge \phi)$ , formulated in  $\mathcal{L}$ . The first two axioms ensure that  $\mathbb{D}$  and  $\mathbb{E}$  are disjoint and exhaustive, and the last axiom ensures that  $R_\tau \in \mathcal{R}$  is a transition relation from  $\mathbb{D}$  to  $\mathbb{E}$ . Call this language  $\mathcal{L}^{\mathbb{D}, \mathbb{E}}$ .  $\mathcal{L}^{\mathbb{D}, \mathbb{E}}$  is a language with a complex ontology, i.e. it distinguishes between two disjoint domains.

On the logical side, the paper is about layered modal languages with rich ontologies. A language is introduced that extends  $\mathcal{L}$  in a number of respects, e.g. it is both hybrid and dynamic. If the reader has no idea about the nature of such a language in advance, please have patience. The language is formally defined below. It is used to encode ECG, a complex theory of natural language understanding that comprises several domains. In fact, the ontology of ECG is further extended here with an inference component. This is the second advantage of our translation, besides from the complexity results: Since layering is fully scalable, it is easy to integrate ECG with a database of real-world knowledge and inference components. The inference component really refers to a structured set of subdomains, incl. a referential, spatial and a temporal subdomain. On the historical note, the layering techniques, as they are employed here, were introduced, in a systematic way, in the logic community by Blackburn and de Rijke (1997), but see references therein for earlier applications of the techniques. It is of course also possible to layer fundamentally different logics.

The structure of the paper is simple: The second section introduces our modal language and its expressivity. The language is, on arbitrary frames, decidable in deterministic exponential time (Areces et al., 1999, Theorem 3.5), and equivalent to propositional dynamic logic with converse, enriched with nominals and global modalities. The next section is a brief presentation of ECG and its translation into our layered modal language. In the fourth section, the complexity results are established. The results are conjectured to transfer to related linguistic formalisms. Our results also indicate that ECG can be implemented on common grammar engineering platforms, and an implementation is sketched for the Lexical Knowledge Base (Copestake, 2001). Finally, a modal language of greater expressivity (and higher complexity) is considered.

## 2 Logic

Our language is, as mentioned, a dynamic and hybrid extension of modal logic, but not only that. It extends  $\mathcal{L}$  with nominals (propositions that denote singleton subsets), satisfaction operators, global modalities, non-deterministic operators and the converse operator. Call the language  $\text{ScU}$ .

**Definition 2.1.** In  $\text{ScU}$ , formulas are built over a set of propositions  $\text{PROP}$ , denoted by  $\{p, q, \dots\}$  or type names such as *word* or *phrase*, and a set of nominals  $\text{NOM}$ , denoted by  $\{i, j, \dots\}$ , s.t.

$$\phi := p|i|\neg\phi|\phi \vee \psi|\langle\alpha\rangle\phi|@_i\phi|\langle\alpha^{-1}\rangle\phi|\langle\alpha^*\rangle\phi|A\phi$$

are wellformed. The satisfaction definition is as follows, wrt. Kripke models  $\langle\mathcal{W}, \mathcal{R}, \mathcal{V}\rangle$ , where  $\mathcal{W}$  is a set of worlds,  $\mathcal{R}$  a set of binary relations, and  $\mathcal{V}$  is the valuation function, i.e. a mapping from  $\text{PROP}$  and  $\text{NOM}$  to the power set of  $\mathcal{W}$ :

$$\begin{array}{lll} M, w \models p & \text{iff} & w \in \mathcal{V}(p) \\ M, w \models i & \text{iff} & \mathcal{V}(i) = \{w\} \\ M, w \models \neg\phi & \text{iff} & M, w \not\models \phi \\ M, w \models \phi \vee \psi & \text{iff} & M, w \models \phi \text{ or } M, w \models \psi \\ M, w \models \langle\alpha\rangle\phi & \text{iff} & \exists w' (R_\alpha(w, w') \& M, w' \models \phi) \\ M, w \models @_i\phi & \text{iff} & M, w' \models \phi \& \mathcal{V}(i) = \{w'\} \\ M, w \models \langle\alpha^{-1}\rangle\phi & \text{iff} & \exists w' (R_\alpha^{-1}(w, w') \& M, w' \models \phi) \\ M, w \models \langle\alpha^*\rangle\phi & \text{iff} & \exists w' (w R_\alpha^* w' \& M, w' \models \phi) \\ M, w \models A\phi & \text{iff} & \forall w' M, w' \models \phi \end{array}$$

where  $R_\alpha^*$  means  $\bigcup_n (R_\alpha)^n$ , and  $n$  is non-deterministically chosen, and where  $R_\alpha^{-1}$  is the converse of  $R_\alpha$ .

For abbreviatory use,  $;$ ,  $\cup$  are defined s.t.

$$;: \langle\alpha\rangle\langle\beta\rangle\phi \leftrightarrow \langle\alpha; \beta\rangle\phi$$

$$\cup: \langle\alpha\rangle\phi \vee \langle\beta\rangle\phi \leftrightarrow \langle\alpha \cup \beta\rangle\phi$$

(The next paragraphs motivate the various operators of  $\text{ScU}$  and characterizes its expressivity, in part. If the reader is unaccustomed to the literature on modal languages, the paragraphs may be somewhat dense, and she is invited to skip the rest of this section.) Our extensions of  $\mathcal{L}$  by nominals, satisfaction operators, global modalities, non-deterministic operators and the converse operator are meaningful, as evidenced by an investigation of the invariance properties of  $\text{ScU}$ . If a logic is invariant under some operation, you may recall, it means that it preserves

validities under this operation, and that properties that are affected by it, cannot be defined.  $\mathcal{L}$  is characterized in part by its invariance under disjoint unions, bounded morphisms and bisimulation. Briefly put, if a logic is invariant under disjoint unions, it means that any validities that hold for some model  $M$ , also hold in the disjoint union  $M \uplus M'$  for any model  $M'$ . A bounded morphism  $h$  ensures that for every  $w \in \mathcal{W}$ ,  $w$  and  $h(w)$  satisfy the same proposition letters,  $h$  is a homomorphism wrt. the relations in  $\mathcal{R}$  (the forth condition), and for every  $R \in \mathcal{R}$ , if  $R'h(w)w'$ , then there exists  $w''$  s.t.  $Rww''$  and  $h(w'') = w'$  (the back condition) (Blackburn et al., 2001, 59). A bisimulation between (unimodal) models  $M = \langle \mathcal{W}, R, \mathcal{V} \rangle$  and  $M' = \langle \mathcal{W}', R', \mathcal{V}' \rangle$  is a non-empty binary relation  $Z$  between their points s.t. whenever  $wZw'$  we have that:

- (i)  $w$  and  $w'$  satisfy the same proposition symbols,
- (ii) if  $Rvw$ , then there exists a point  $v' \in M'$  s.t.  $vZv'$  and  $R'w'v'$ , and
- (iii) if  $R'v'w'$ , then there exists a point  $v \in M$  s.t.  $vZv'$  and  $Rvw$ .

This generalizes straight-forwardly to the polymodal case. ScU is not invariant under any of these operations. Invariance under disjoint unions is lost, once you add global modalities. This is obvious, since global modalities allow you to say, for instance, “there is no state in  $\phi$ ”. This may hold true for  $M$ , but not for  $M'$ , and thus not for  $M \uplus M'$  either. ScU clearly encodes hybrid logic ( $\mathcal{H}$ ), which is defined as the extension of  $\mathcal{L}$  with nominals and satisfaction operators. Since  $\mathcal{H}$  is non-invariant under bounded morphisms (ten Cate, 2005), the non-invariance of ScU follows. The notion of bisimulation, finally, does not take nominals into account. The operation can be modified a bit, however. The bisimulation operation for  $\mathcal{H}$ , call it  $\mathcal{H}$ -bisimulation, adds a fourth clause to its definition, namely that

- (iv) all points named by nominals are related by  $Z$ .

In addition, clause (i) takes  $\text{NOM} \cup \text{PROP}$  as its domain, rather than just  $\text{PROP}$ . Unfortunately, this is not enough. The global modalities force us to add a fifth requirement, namely that the operation is total, i.e. s.t.  $\forall \nu_i \in W \exists \nu_j \in W' \nu_i Z \nu_j$ , and vice versa. The obvious reason is that global modalities are sensitive to information in non-local (disconnected) substructures. Call this total bisimulation for  $\mathcal{H}(E)$ -bisimulation (ten Cate, 2005, 47).

**Lemma 2.2.** *ScU is invariant under  $\mathcal{H}(E)$ -bisimulation.*

*Proof.* This follows from the invariance of  $\mathcal{H}$  extended with global modalities ( $\mathcal{H}(E)$ ) under  $\mathcal{H}(E)$ -bisimulation (ten Cate, 2005), and that  $*$  and  $-1$  are safe for  $\mathcal{H}(E)$ -bisimulation. Van Benthem (1998) proved that  $*$  is safe for bisimulation. Consequently, if  $Z$  is a bisimulation from  $M = \langle \mathcal{W}, R, \mathcal{V} \rangle$  to  $M' = \langle \mathcal{W}', R', \mathcal{V}' \rangle$ , it is also a bisimulation between  $M = \langle \mathcal{W}, R^*, \mathcal{V} \rangle$  and  $M' = \langle \mathcal{W}', (R')^*, \mathcal{V}' \rangle$ . If  $Z$  is a  $\mathcal{H}(E)$ -bisimulation it also satisfies the fourth clause of the definition above, i.e. all points named by nominals are related by  $Z$ , and it must have a full domain and range (be total). These properties are not dependent on the transition relations, however, since they only depend on the domains and valuations of the models.  $*$  is thus safe for  $\mathcal{H}(E)$ -bisimulation too. The safety of the converse operator  $-1$  follows from the equivalence of  $\mathcal{H}(E)$ -bisimulation and so-called power bisimulation, introduced recently by van Benthem for a logic of games. Since this logic included  $-1$  and was invariant under power bisimulation, it follows that  $-1$  is also safe for  $\mathcal{H}(E)$ -bisimulation.  $\square$

It can be inferred from the non-invariance of  $\mathcal{H}$  under bounded morphism that the extension of  $\mathcal{L}$  by nominals and satisfaction operators is meaningful, while the non-invariance under disjoint union tells us that global modalities also add expressive power to our language. The extension of  $\mathcal{L}$  by non-deterministic operators is meaningful too, evidenced by the loss of compactness (Blackburn et al., 2001, 240). (Since  $\mathcal{L}$  has a translation into first order logic, it is

compact.) Converse, finally, is also meaningful. Consider, for instance, the model  $\{R^{-1}(w, w')\}$ , which cannot be singled out by any  $\mathcal{L}$  or propositional dynamic formula.

Note that ScU, unlike  $\mathcal{L}$ , does *not* have the tree model property, i.e. it is not the case that  $\phi$  is satisfiable iff it is satisfiable in a tree-like Kripke model. This is important, since the grammar formalisms considered here all employ reentrancies, a property that violates the tree model property. In other words, if ScU had the tree model property, it would not be able to encode or reconstruct the formalisms this paper concerns.

### 3 Grammar

The fundamental concept in ECG is that of feature structures. The reader is invited to think of such feature structures as Kripke structures. On this view, with some mild abuse of notation, it holds that, for instance,

$$\left[ \begin{array}{c} \text{FUNCT} \left[ \text{NUM} \boxed{\phantom{a}} \right] \\ \text{ARG} \left[ \text{NUM} \boxed{\phantom{a}} \right] \end{array} \right] \\ \models \langle \text{FUNCT} \rangle \langle \text{NUM} \rangle i \wedge \langle \text{ARG} \rangle \langle \text{NUM} \rangle i$$

The tags are used to “co-index” two values. Consequently, they denote reentrancies. In ScU, tags translate into nominals, since they denote singleton sets of states. The standard ECG notation look different from standard attribute-value matrices, but to a large extent the schemas of ECG can be viewed as notational variants of Kripke structures. For instance,

<b>schema</b> Trajector-Landmark <b>subcase of</b> Image-Schema <b>roles</b> trajector: $a$ landmark: $b$
---

is equivalent to

$$\left[ \begin{array}{c} \text{trajector\_landmark} \\ \text{ROLES} \left[ \begin{array}{c} \text{TRAJECTOR } a \\ \text{LANDMARK } b \end{array} \right] \end{array} \right] \\ \wedge \text{trajector\_landmark} \sqsubseteq \text{image\_schema}$$

The last formula says that *trajector\_landmark* is a subtype of *image\_schema*. The root node is

thus in the denotation of both propositions. The *trajector\_landmark* type is a schema. Schemas and constructions constitute two different domains in ECG. Schemas, in general, encode conceptual structure and constrain information about conceptual roles and the relations among them. Constructions are linguistic of nature and pair up form and meaning (two subdomains identified by separate attributes). The third domain is that of spaces. The ontology, summarized here, is presented in more detail in Chang et al. (2002). Let *schema* denote the domain of schemas, *constr* denote the domain of constructions, while *ms* denotes the domain of (mental) spaces. The domain structure is defined in the same way as in Example 1.1. A map modality is also introduced; this is used to encode the **map**-operator, also introduced in Chang et al. (2002), which is said to identify “correspondences across a pair of conceptual domains (e.g. between two schemas, or between two spaces).”

- (1)  $\text{schema} \vee \text{constr} \vee \text{ms}$
- (2)  $\neg(\text{schema} \wedge \text{constr} \vee \text{schema} \wedge \text{ms} \vee \text{constr} \wedge \text{ms})$
- (3)  $\langle \text{map} \rangle \phi \rightarrow (\text{schema} \wedge \langle \text{map} \rangle (\text{schema} \wedge \phi)) \vee (\text{ms} \wedge \langle \text{map} \rangle (\text{ms} \wedge \phi))$

Reentrancies are denoted by  $\leftrightarrow$ . In other words,

$$\boxed{a \leftrightarrow b}$$

corresponds to

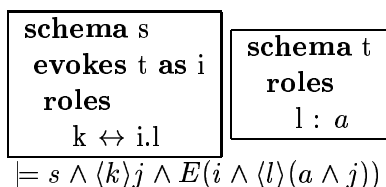
$$\left[ \begin{array}{c} A \boxed{\phantom{a}} \\ B \boxed{\phantom{a}} \end{array} \right]$$

Feature structures, more generally, are governed by a number of axioms; see for instance Wedekind (1997). Feature structures have (mostly) deterministic attributes, and they are acyclic, connected and rooted. This can be encoded in ScU in this way, if *root* is a nominal:

- (4)  $\langle \alpha \rangle i \rightarrow [\alpha] i$  (functionality)
- (5)  $i \rightarrow \neg \langle \alpha_1 \cup \dots \cup \alpha_n \rangle^+ i$  (acyclicity)
- (6)  $E(\text{root} \wedge \neg \langle \alpha^{-1} \rangle \top) \wedge A(\langle \alpha_1 \cup \dots \cup \alpha_n \rangle^* \text{root})$  (connectedness and rootedness)

$R_\alpha$  is, as in the above, any relation in  $\mathcal{R}$ ,  $\mathcal{R} = \{\alpha_1 \dots \alpha_n\}$ , and axioms are multiplied in the set of nominals.

Up to this point, ECG looks exactly like other unification-based formalisms (the feature geometry and its linguistic use aside). ECG employs two novel operators, however; most significantly, the so-called **evokes** operator, which triggers a new schema and allows reference to it. In ScU, this corresponds to existential quantification and naming, e.g.



The other complication is the “::” constraints. Since this constraint involves quantification over times, it is discussed in the section on inference just below, where a temporal domain is introduced.

ECG usually employs rather flat constructions, and linearization can be constrained in two ways: either by weak or immediate precedence (Bryant, 2004). We use standard terminology and let  $\prec$  denote weak precedence and  $\ll$  denote immediate precedence. It is an option to augment the ECG feature geometry with a domain of substructures for phonological strings (distinguished by a feature PHON, for instance). The standard encoding of lists as trees can be employed, and ScU used to constrain the tree structures. One can, alternatively, represent strings by linear relations. This is the solution adopted here. This is possible if the input to the parsing procedure is the lexical entries rather than the strings themselves. It is of course trivial to do this substitution (lexical ambiguity just blows up the number of inputs).  $\prec$  and  $\ll$  are then interpreted directly over substructures (or type propositions, really), i.e.  $\ll$  is a deterministic version of the  $\prec$  modality s.t.

$$(7) \quad \prec \phi \Leftrightarrow \ll^+ \phi$$

Mental space theory, originally due to Gilles Fauconnier, is used as a semantic representation language in an extension of ECG proposed by

Mok et al. (2004). The implementation suggested there does not increase the formal complexity of ECG. It only extends the feature geometry. A separate domain for spaces was already introduced.

We are now in a position to show how ECG analysis works, and more importantly, a parsing procedure is identified. Contrary to the parsing procedure hinted at in (Bryant, 2004), this algorithm is entirely model-theoretic. The toy implementation presented below is comparable to Bryant’s proposal, in that the Lexical Knowledge Base employs generative rules in a bottom-up fashion. The basics of our parsing procedure is:

1. The string is encoded in ScU and conjoined with the axioms of ECG.
2. If this conjoined formula is satisfiable, the string is accepted.

It is easy to represent strings in our language, as already indicated. The job is to identify the proper axioms. Some were already defined, i.e. (1-7). It is now necessary to define the type hierarchy, which includes the various types of schemas, constructions, and spaces. It is also necessary to define some start or root type, which is existentially quantified over. Why is that? This is to ensure that all the relevant information is derived in the parse. The definition of the type hierarchy, and total well-typing in particular (Carpenter, 1992), is half the way, but it is necessary to ensure, somehow, that the model which satisfies the string is forced to contain a full linguistic analysis (and not just the empty model or some arbitrary connection of the constituent substructures derived from the lexicon). This is easily obtained by existential quantification over the root and connectivity. If there is a root and a set of constituent substructures, and the feature structure has to be connected, the additional structure falls out by implications. It is necessary, however, since this is a typed system, to ensure that the “universal” phrase, i.e. the supertype *phrase*, cannot apply. In generative or rule-based systems, such as the

Lexical Knowledge Base, this is ensured by stipulating directly which types count as rules. On a model-theoretic set-up, it is necessary to add some implication that every phrase is a particular type of phrase.<sup>1</sup>

### 3.1 Inference

To this we add an inference component. The inference component has a complex structure and comprises multiple subdomains. Only three domains are included here, for simplicity. The domains include a referential (R), a spatial (S), and a temporal one (T). The transitions are called  $R_{r2s}$ , from the referential to the spatial domain, and  $R_{s2t}$ , from the spatial to the temporal domain. The domains are denoted by propositions *ref*, *space* and *time*.

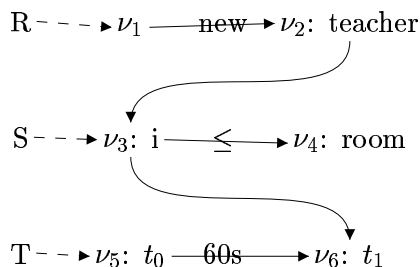
**Example 3.1.** Our layering is illustrated by this simple example:

- (8) The new teacher enters the room in a minute.

Think of the room as a spatial region that the teacher at some point (in time) is part of.  $t_0$  is the time of the utterance.

$$\begin{aligned} & E(\text{time} \wedge t_1 \wedge @_{t_0}\langle 60s \rangle t_1) \\ \wedge & E(\text{space} \wedge i \wedge \leq \text{room} \wedge @_i \langle s2t \rangle t_1) \\ \wedge & E(\text{ref} \wedge \langle \text{new} \rangle (\text{teacher} \wedge \langle r2s \rangle i)) \end{aligned}$$

The model that satisfies the formula will look like this:



Here R is the referential domain, S is the spatial domain, and T is the temporal one. The curved lines are the projections from one domain to another, i.e.  $R_{r2s}$  and  $R_{s2t}$ . The R domain says that there is a teacher who is new. The S domain says that his spatial instantiation is part of the spatial region that is occupied by

<sup>1</sup>This is roughly the role of the immediate dominance principle in head-driven phrase structure grammar.

the room at time  $t_1$ . The T domain tells us that  $t_1$  is 60 seconds from now.

*Remark 3.2.* A remark on the  $::$  constraints of ECG was promised you. In Chang et al. (2002, Figure 2), two such constraints are used to say that in the schema “Translational-Motion”, the mover’s location changes from that of the source to that of the goal at the time of the motion, i.e. because of the motion. Call this temporal state  $t_m$ . The constraints

$$\begin{aligned} & \dots \\ & \text{before} :: \text{mover.location} \leftrightarrow \text{source} \\ & \text{after} :: \text{mover.location} \leftrightarrow \text{goal} \end{aligned}$$

thus translate into

$$\begin{aligned} & A(\langle \langle \rangle t_m \rightarrow \langle s2t^{-1} \rangle \text{source} \rangle) \text{ and} \\ & A(\langle \langle \rangle t_m \rightarrow \langle s2t^{-1} \rangle \text{goal} \rangle) \end{aligned}$$

## 4 Computational issues

ScU is decidable in deterministic exponential time (Areces et al., 1999, Theorem 3.5). It follows from the parsing procedure and the completeness of our translation that so is ECG. If the inference component is ignored, the **evokes**-free fragment of ECG is apparently more efficient, since an NP-completeness result can be established for it. Obtaining the result is not too complicated. The domain minimal Kripke model that satisfies an input formula in these theories is bound by a size lemma. ECG contains no unary constructions (as it is shown below, **evokes** does the job of unary projections), but such can be freely added, provided that unary rules only apply once to the same unary extension. If a unary extension is always the result of a single application of a unary rule,  $u = 1$ .

**Lemma 4.1.** *A ScU formula that encodes a recognition problem for the **evokes**-free fragment of ECG for some string  $\sigma$  is, if satisfiable, satisfied by a structure  $M$  of at most  $(2|\sigma| - 1)(u + 1) \times \mathbf{paths}$  cardinality, where  $2|\sigma| - 1$  is the maximum number of nodes in a binary tree,  $u$  is the number of unary rules, and **paths** is the number of paths licensed by the non-recursive part of the feature geometry of ECG. In particular,  $\mathbf{paths} = |\{\pi \in \mathbf{Lb1s}^* | \text{no label occurs twice in } \pi\}|$ .*

Since any ScU formula  $\phi$  that represents an **evokes**-free ECG recognition problem for a string of  $c$  length, can be evaluated in a model of size polynomial to  $c$ , a suitable model can be non-deterministically chosen, and  $\phi$  is evaluated in polynomial time, since model checking in ScU and similar languages is decidable in polynomial time (Franceschet and de Rijke, 2006). It follows that

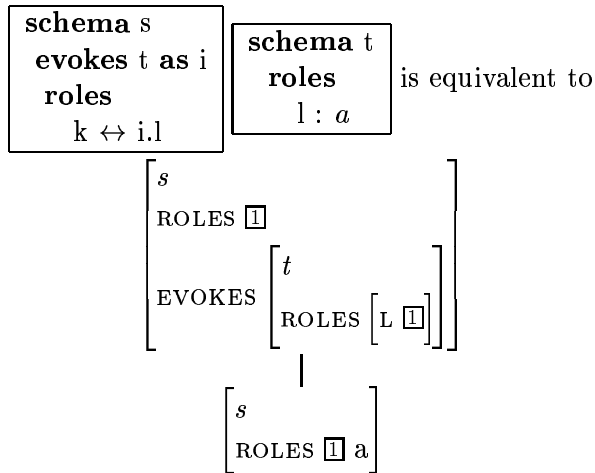
**Theorem 4.2.** *The universal recognition problem of **evokes**-free ECG is decidable in non-deterministic polynomial time.*

*Remark 4.3.* The result in Theorem 4.2 transfers to unification categorial grammar and construction grammar where  $u$  is bound in the same way.

#### 4.1 Implementation

In this subsection, it is sketched how ECG, at least in part, can be implemented in the Lexical Knowledge Base (Copestake, 2001). This has already been hinted at in different places. The reader familiar with both the computational platform and the literature on ECG may envisage how to implement the **evokes**-free fragment. Consequently, we focus only on how to translate the **evokes** operator by unary rules. Lexical rules could also have been used, but lexical rules are rather heavy and unruly machineries, and they are best avoided if possible.

We consider the example from above. It holds that



The **evokes** operator is thus replaced by a recursive attribute which embeds the new structure. Since the EVOKES attribute is defined for

this sole purpose, it is really no different from the disconnected substructure intended in the ECG-style notation. The other complication, the  $::$  operator, deserves a brief remark too. If the relational constraints of the schemas are augmented with events, that is, each proposition is defined relative to some event (as in Davidsonian semantics), the  $::$ -style constraints can be translated into ordinary relations. This extends the feature geometry a bit, of course, but enables us to implement ECG in the Lexical Knowledge Base.

#### 4.2 Using a richer logic

There is one qualification to our formalization that has not been mentioned yet (except for in the abstract), namely that certain undesired reentrancies may appear in the context of the **evoke** operator (and schemas and constructions, more generally), if the nominals used in the reduction are not somehow dynamically reset. The intuition is this: If a schema runs an **evoke** operation, it existentially asserts a substructure and names some of its nodes by nominals. However, the next time this schema applies, the substructure in the scope of the operator, receives the same names, unless they are somehow changed (dynamically). It is possible to avoid this complication, if one “steps up” our logic a bit, i.e. if we add quantification over nominals. We are then able to quantify over both nominals and states. Call the relevant quantifier  $\exists$ . Its satisfaction definition is

$$M, g, w \models \exists x \phi \quad \text{iff} \quad \exists g' g \stackrel{c}{=} \& M, g', w \models \phi$$

Note that assignments have now been added to the satisfaction definitions, for obvious reasons. The standard translation into first order logic illustrates its intended meaning:

$$ST_x(\exists y \phi) = \exists y ST_x(\phi)$$

Our logic now, with this new extension, embeds hybrid logic extended with the  $\Downarrow$  operator,<sup>2</sup> and since this logic has been shown to be undecidable (Blackburn and Seligman, 1995), our logic is thus undecidable itself. However, NP-completeness can still be established for a useful

<sup>2</sup>This follows from the fact that  $\Downarrow x \phi \doteq \Downarrow x A(\phi)$ .

fragment. The intuition behind the fragment is, as we saw in our computational implementation, that **evokes** operations are much like unary projections. Constraint-based grammars are often (only) undecidable in the absence of off-line parsability constraints, and in fact an “off-line parsability” restriction can be placed on the use of the **evokes** operator. Simply label all the **evokes** operators employed in the schemas. Say that any schema that evokes a schema or space that evokes another one, etc., can never evoke itself in this chain of evocations. Call our new language  $\text{ScU}^+$  and let **evocators** be the number of **evokes** operators employed by the grammar. Call the restriction on the use of **evokes** the “acyclic evocation restriction.” The following size lemma now holds:<sup>3</sup>

**Lemma 4.4.** *A  $\text{ScU}^+$  formula that encodes an ECG recognition problem for some string  $\sigma$  under the acyclic evocation restriction is, if satisfiable, satisfied by a structure  $M$  of at most  $(2|\sigma| - 1)\mathbf{evocators} \times \mathbf{paths}$  cardinality, where **evocators** and **paths** are defined as in the above.*

It follows as a corollary that

**Theorem 4.5.** *The universal recognition problem of ECG under the acyclic evocation restriction is decidable in non-deterministic polynomial time.*

*Proof.* Similar to the proof of Theorem 4.2.  $\square$

## 5 Conclusions

The technique of using layered modal language for scalable natural language understanding was introduced and applied to ECG. Certain interesting results were derived. ECG is decidable in exponential time (under dynamic resetting), its **evokes**-free fragment is NP-complete, and so is ECG under the acyclic evocation restriction, i.e. if we put a bound on the length of evocations which is polynomial in the length of the input. It was also shown to be likely that ECG can be implemented in the Lexical Knowledge Base.

<sup>3</sup>It is, for simplification, assumed that each schema or space contains at most one **evokes** operator.

## References

- Carlos Areces, Patrick Blackburn, and Maarten Marx. 1999. A road-map on complexity for hybrid logics. In *Proceedings of the Annual Conference of the European Association for Computer Science Logic*, volume 1683 of *Lecture Notes in Computer Science*, pages 307–321, Berlin. Springer.
- Patrick Blackburn and Maarten de Rijke. 1997. Zooming in, zooming out. *Journal of Logic, Language and Information*, 6:5–31.
- Patrick Blackburn and Jerry Seligman. 1995. Hybrid languages. *Journal of Logic, Language and Information*, 4:251–272.
- Patrick Blackburn, Maarten de Rijke, and Yde Venema. 2001. *Modal logic*. Cambridge University Press, Cambridge, England.
- John Bryant. 2004. Scalable construction-based parsing and semantic analysis. Presented at Scalable Natural Language Understanding 2004.
- Bob Carpenter. 1992. *The logic of typed feature structures*, volume 32 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, England.
- Nancy Chang, Jerome Feldman, Robert Porzel, and Keith Sanders. 2002. Scaling cognitive linguistics: formalisms for language understanding. Presented at Scalable Natural Language Understanding 2002.
- Ann Copestake. 2001. *Implementing typed feature structure grammars*. CSLI, Stanford, California.
- Massimo Franceschet and Maarten de Rijke. 2006. Model checking for hybrid logics. *Journal of Applied Logic*. In press.
- Eva Mok, John Bryant, and Jerome Feldman. 2004. Scaling understanding up to mental spaces. Presented at Scalable Natural Language Understanding 2004.
- Balder ten Cate. 2005. *Model theory for extended modal languages*. Ph.D. thesis, University of Amsterdam, Amsterdam, the Netherlands. ILLC Dissertation Series DS-2005-01.
- Johan van Benthem. 1998. Program constructions that are safe for bisimulation. *Studia logica*, 60:311–330.
- Jürgen Wedekind. 1997. Approaches to unification in grammar. In Patrick Blackburn and Maarten de Rijke, editors, *Specifying syntactic structure*, pages 245–280. CSLI Publications, Stanford, California.