

# Typed feature structure grammars and model generation

March 16, 2006

## Abstract

The purpose of this paper is to demonstrate the use of model generation techniques in the formal study and practical application of typed feature structure grammars. Our presentation employs a toy formalism at the intersection of construction grammar, unification categorial grammar and head-driven phrase structure grammar. Model generation techniques are used to establish the decidability, and more specifically, decidability in non-deterministic polynomial time, of the universal recognition problem of this formalism - a result that carries over to construction grammar and unification categorial grammar, and some restricted versions of head-driven phrase structure grammar, but an actual implementation is also presented, which has the advantage that it can easily be extended with conventional automated reasoning techniques for natural language understanding.

In many grammar formalisms, the language of a grammar is the set of strings that can be generated from some start node by application of the rules of the grammar. In a sense, one can think of the grammar as a model that satisfies a number of strings. The model-theoretic perspective is often said to turn things upside down. The language of a grammar is, under the common interpretation of this perspective, the set of strings that satisfy the grammar, so to speak. Or when it comes to typed feature structure grammars, the language of a grammar is the set of linearizations of the typed feature structures, which are in all respects equivalent to Kripke frames, that satisfy the axioms of the grammar (which roughly correspond to rules on the first, more traditional perspective). In this paper, this view is modified a bit. So the language of a grammar is now *the set of strings that are satisfiable in conjunction with the axioms of the grammar*.

For representational reasons, a toy formalism has been designed that captures the basics of construction grammar, unification categorial grammar and head-driven phrase structure grammar. The formal details are included in the full paper. Consider the derivation in Figure 1 for illustration.

In our toy formalism, a toy grammar was constructed to account for a simple fragment of English, namely

John walks (the path) slowly\*

The toy grammar was implemented on a first order model generation program called Paradox in the following manner: First the linguistic input was converted into first order logic, i.e. each word was mapped onto its lexical representation (a rigid lexicon was assumed for simplicity), which was then converted into an existential claim that there exists a set of relational substructures (one per constituent) and a relation that enforces a “linear order” on these substructures. Next some linguistic principles were encoded in first order logic, e.g. a strong (endocentric) version of the head feature principle, which in first order logic and our feature geometry amounts to

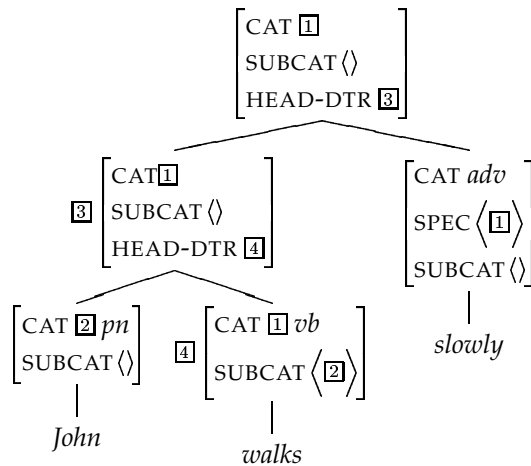


Figure 1: The derivation of *John walks slowly* in our toy formalism.

```

input_formula(hfp, axiom, (
! [X1, X2, X3]:
  (head(X1, X2)
   & cat(X1, X3)
   => cat(X2, X3))) .

```

Then the axioms of typed feature structures were written down in first order logic, incl. functionality of arcs, connectivity and acyclicity of feature structures, total welltyping, etc. The feature geometry, incl. the disjointness of the arcs and the value restrictions, and the type hierarchy were of course also defined. Finally, the parse is initiated by saying that there exists a saturated root of type *phrase*. Intuitively, the parse comes about in this way: The model generation program is told that there exists a number of substructures, which correspond to the lexical entries of the constituents of the string, and a root, and it is also told that the model it has to output is connected. It now tries to connect the root and the substructures according to the axioms of our toy formalism.

Our formalization has important theoretical and practical implications. On the practical side, it is now easy to integrate typed feature structure parsing with general-purpose automated reasoning. On the theoretical side, a complete formalization of a typed feature structure grammar is given to us in a classical logic. One thing that derives from this formalization is a decidability result for the universal recognition problem. The result is by reduction to the two-variable fragment of first order logic. The details are beyond the scope of this abstract. Finally, I will touch upon similar research with hybrid logic model generation programs in the full paper. In fact, this work can be used to obtain an even better result for the unrestricted formalism, which, I will show, embeds standard versions of construction grammar and unification categorial grammar, since our formalism has the polysize model property, i.e. the size of a model is polynomially bound by the length of the input (the string to be parsed), and the hybrid logic that is employed, has a model checking problem decidable in polynomial time: This means that the universal recognition problem of our toy formalism is decidable in non-deterministic polynomial time, i.e. this is an upper bound of the complexity of our formalism.