Verifying context-sensitive treebanks and heuristic parses in polynomial time[*]

Anders Søgaard
Center for Language Technology
University of Copenhagen
Njalsgade 140–142
DK-2300 Copenhagen S
soegaard@hum.ku.dk

Abstract

A polyadic dynamic logic is introduced in which a model-theoretic version of nonlocal multicomponent tree-adjoining grammar can be formulated. It is shown to have a low polynomial time model checking procedure. This means that treebanks for nonlocal MCTAG, incl. all weaker extensions of TAG, can be efficiently corrected and queried. Our result is extended to HPSG treebanks (with some qualifications). The model checking procedures can also be used in heuristics-based parsing.

1 Introduction

First order logics and monadic second order logics have been used to query standard treebanks of context-free derivation structures (Kepser, 2004). The model checking problems for both logics are known to be PSPACE-complete (Blackburn et al., 2001), however. Moreover, treebanks are now being constructed that replace context-free derivation structures with context-sensitive ones, incl. The Prague Dependency Treebank (Hajičová et al., 2001), The Danish Dependency Treebank (Buch-Kromann, 2007), The LinGO Redwoods Treebank (English) (Oepen et al., 2002), and BulTreeBank (Simov et al., 2004).

Maier and Søgaard (2008) show that even German standard treebanks such as TIGER and NeGra contain mildly context-sensitive derivation structures. The dependency treebanks also use mildly context-sensitive derivation structures (Kuhlmann and Möhle, 2007); the frequency of non-context-free structures in these treebanks is estimated in Nivre (2006) and is similar to the frequency of such strucures in TIGER and NeGra (Maier and Søgaard, 2008). The HPSG treebanks (Redwoods and BulTreeBank) also contain context-sensitive derivation structures (and beyond). The obvious question to ask now is: Are there less complex logics that can be used to correct and query context-sensitive treebanks?

This paper introduces a polyadic modal logic called *decharge logic*. Its model checking problem can be solved in low polynomial time; a model checking algorithm is spelled out. It is shown that decharge logic captures context-sensitive nonlocal multicomponent tree-adjoining grammars (MCTAGs) (Becker et al., 1991) in the following sense: For each non-local MCTAG G, there exists a decharge logic D such that $\omega \in L(G)$ iff $\exists M.M \models_D \omega$, i.e. if a string is recognized by the grammar G it is satisfiable in the corresponding logic. D is thus a model-theoretic characterization of G.

Nonlocal MCTAG is context-sensitive, but not mildly context-sensitive (Rambow and Satta, 1992), and its fixed and universal recognition problems are NP-complete. Head-driven phrase structure grammar (HPSG) (Pollard and Sag, 1994) is strictly more expressive, i.e. it is possible to reconstruct nonlocal MCTAGs in the HPSG formalism (Søgaard, 2007). In other words, every nonlocal MCTAG is, formally, a HPSG. This doesn't tell us much, since, formally, most things are HPSGs: most formalizations of HPSG are Turing complete (Hegner, 1996). Even the model checking problem of the standard logical formalization of HPSG – known as relational speciate

[*]The model checking procedure described in this paper uses constructs from a model checking procedure introduced in joint work with Martin Lange. Thanks also to Laura Kallmeyer, Timm Lichte and Wolfgang Maier for introducing me to various extensions of tree-adjoining grammar, incl. nonlocal MCTAG.

reentrant logic (RSRL) (Richter, 2004) – is undecidable (Søgaard, 2007). HPSG is captured in the above sense (with some qualifications) by an extended version of decharge logic whose model checking problem remains low polynomial time solvable (Søgaard and Lange, 2009).

Note on style: Knowledge of tree-adjoining grammar and HPSG is assumed for brevity. Instead a more detailed introduction is given to the concepts from modal logic used in decharge logic. See Joshi and Schabes (1997) for a recent introduction to tree-adjoining grammar. Since the paper covers some ground, proofs are only presented as informal proof sketches.

In general, the point of the paper is to present decharge logic and its extension and to argue that these logics may be relevant for natural language processing. The technical results are sketched, but only informally. No motivation is provided for the move to context-sensitive formalisms itself. The point is simply: *if* you want to use context-sensitive treebanks and query them, decharge logic has better computational properties than the other logics proposed in the literature for linguistic theories such as nonlocal MCTAG and HPSG. The model checking algorithms can also be used in heuristics-based parsing. Since neither nonlocal MCTAG nor HPSG has efficient parsing procedures, real-life parsing will typically be heuristics-based. A derivation structure is guessed (though not in a completely arbitrary fashion), rather than derived, and model checking can be used to check if the derivation structure satisfies whatever linguistic principles not guaranteed by the heuristics.

2 Decharge logic

2.1 Modal and dynamic logic

The logics covered in this brief introduction are all modal extensions of propositional logic. Propositional logic is the classic logic over propositional variables and Boolean connectives. Basic modal logic extends propositional logic with monadic operators $\Diamond_i, \Diamond_j, \ldots$, or in a notational variant $\langle i\rangle, \langle j\rangle, \ldots$, known as "diamonds" and their duals known as "boxes" (written $\Box_i, \Box_j, \ldots$ or $[i], [j], \ldots$). See Blackburn et al. (2001) for an introduction. The monadic operators introduce binary relations. The diamonds intuitively mean "there is a relation from the current state to a state for which it holds that". For example, the formula $\langle i\rangle p$ means that there is a relation (indexed by i) from the current state to a state in the denotation of p. The relation indeces are called labels (Labels), and the propositional variables are called atoms (Atoms). The syntax of basic modal logic over a signature $\langle \mathsf{Labels}, \mathsf{Atoms}\rangle$ is:

$$\phi, \psi \quad \doteq \quad p \mid \phi \wedge \psi \mid \neg\phi \mid \langle a\rangle\phi$$

where $a \in \mathsf{Labels}$ and $p \in \mathsf{Atoms}$. $[a]\phi \doteq \neg\langle a\rangle\neg\phi$ for all $a \in \mathsf{Labels}$.

Semantics is defined in terms of satisfaction definitions over Kripke models (henceforth, models) $M = \langle \mathbb{W}, \{R_a \in a \in \mathsf{Labels}\}, \mathcal{V}\rangle$ where $\mathbb{W}$ is a finite set of states (or worlds), $R_a \subseteq \mathbb{W} \times \mathbb{W}$, and $\mathcal{V} : \mathsf{Atoms} \rightarrow 2^{\mathbb{W}}$ a valuation function. The satisfaction definitions are as follows:

$$\begin{array}{rcl} M, w \models p & \text{iff} & w \in \mathcal{V}(p) \\ M, w \models \phi \wedge \psi & \text{iff} & M, w \models \phi \;\&\; M, w \models \psi \\ M, w \models \neg\phi & \text{iff} & M, w \not\models \phi \\ M, w \models \langle a\rangle\phi & \text{iff} & \exists w'.R_a(w, w') \;\&\; M, w' \models \phi' \end{array}$$

Example 2.1. The model

$$\begin{tikzcd} & 0 : p, \neg q \arrow[dl] \arrow[dr] & \\ 1 : p, \neg q \arrow[dr] & & 2 : \neg p, \neg q \arrow[dl] \\ & 3 : \neg p, q & \end{tikzcd}$$

with all edges in R_a, except $(2, 3) \in R_b$, satisfies the formulas (i) $\langle b\rangle\top \rightarrow \langle b\rangle q$, since all edges in R_b lead to states in the denotation of q, and (ii) $\neg[a]\neg q$, since not all edges in R_a lead to states in the complement of the denotation of q.

Clearly, basic modal logic is not powerful enough to capture HPSG, since modal logic has the tree model property (Blackburn et al., 2001), i.e. if there exists a model that satisfies ϕ it is possible to unravel this model into a tree. Since reentrancies are used discriminatively in HPSG, it is clear that any logic that has the tree model property is too weak to capture HPSG. The reason that basic modal logic is too weak to capture nonlocal MCTAG is more subtle. Basic modal logic is invariant under generated substructures (Blackburn et al., 2001), i.e. if ϕ is true in all states of a model it is also true in all states of a submodel (by the tree model property also a subtree) generated in one of those states. Since set saturation, used in both nonlocal MCTAG and HPSG, relies on an "upwards query", i.e. if a set (labeled by some FEATURE in the case of HPSG) is introduced in a state w, then w must be dominated by a state with an empty set (labeled by some FEATURE in the case of HPSG), it is clear that any logic that is invariant under generated substructures is too weak to capture nonlocal MCTAG (and HPSG).

Propositional dynamic logic (PDL) is an extension of modal logic in which it is possible to do up- and downwards indeterministic queries such as "somewhere down/up the model it holds that". The syntax of PDL over a signature $\langle \mathsf{Labels}, \mathsf{Atoms} \rangle$ not only defines a set of formulas, but also a set of programs $\mathsf{Programs}$. Diamonds and boxes can now be indexed by programs rather than just labels, and relations are induced over models:

$$\begin{array}{lcl}\phi, \psi & \doteq & p \mid \phi \wedge \psi \mid \neg\phi \mid \langle \alpha \rangle \phi \\ \alpha, \beta & \doteq & \epsilon \mid a \mid \alpha;\beta \mid \alpha^* \mid \alpha \cup \beta \mid \alpha^{-1} \mid \phi^?\end{array}$$

where $a \in \mathsf{Labels}$ and $p \in \mathsf{Atoms}$. The satisfaction definitions are the same as for basic modal logic, except the last clause is generalized to programs:

$$M, w \models \langle a \rangle \phi \quad \text{iff} \quad \exists w'.R_a(w, w') \;\&\; M, w' \models \phi'$$

Each program α, as already mentioned, induces a relation R_α over a model with states $\mathbb{W}$ that is inductively defined:

$$\begin{array}{rcl}R_\epsilon & \doteq & \{(w,w) \mid s \in \mathbb{W}\} \\ R_{\alpha;\beta} & \doteq & \{(w,w') \mid \exists (w,v) \in R_\alpha \;\&\; (v,w') \in R_\beta\} \\ R_{\alpha^*} & \doteq & \bigcup_k R_{\alpha^k} \text{ w. } R_{\alpha^0} = R_\epsilon \& \; R_{\alpha^{k+1}} = R_{\alpha;\alpha^k} \\ R_{\alpha \cup \beta} & \doteq & R_\alpha \cup R_\beta \\ R_{\alpha^{-1}} & \doteq & \{(w,v) \mid (v,w) \in R_\alpha\} \\ R_{\phi?} & \doteq & \{(w,w) \mid M, w \models \phi\}\end{array}$$

Intuitively, ϵ is the empty transition, $\alpha;\beta$ is composition, α^* is Kleene closure, $\alpha \cup \beta$ is union, α^{-1} is converse and $\phi?$ is a test.

Example 2.2. The model

$$\begin{tikzcd} & 0 : \neg p, \neg q \arrow[dl] \arrow[dr] & \\ 1 : p, \neg q \arrow[dr] & & 2 : \neg p, \neg q \arrow[dl] \\ & 3 : p, q & \end{tikzcd}$$

with all edges in R_a satisfies the formulas (i) $\neg[a^*]p$, since $0 \notin \mathcal{V}(p)$, and (ii) $\langle a \rangle q$, since any state dominates a state in the denotation of q.

Note that PDL is not invariant under generated substructures. The formula $\langle (a^*)^{-1} \rangle p$, for example, is true in the model:

$$\begin{tikzcd} & 0 : p \arrow[dl] \arrow[dr] & \\ 1 : \neg p \arrow[dr] & & 2 : \neg p \arrow[dl] \\ & 3 : \neg p & \end{tikzcd}$$

with all edges in R_a, but not in any of its proper generated submodels. PDL still has the tree model property and is thus not adequate for HPSG (nor as a stand-alone logic for non-local MCTAG). A slight extension of PDL, namely PDL with intersection, has been proposed for simpler unification-based formalisms and basic tree-adjoining grammar (Keller, 1993; Blackburn and Spaan, 1993). The extension simply adds a clause $\alpha \cap \beta$ to the syntax of programs with semantics:

$$R_{\alpha \cap \beta} \quad \doteq \quad R_\alpha \cap R_\beta$$

PDL with intersection does not have the tree model property, since, for example, $\langle a \cap b \rangle \top$ is not satisfied by any tree-like model. The model checking problem for PDL with intersection can be solved in linear time (Lange, 2006). Consequently, querying simpler unification-based treebanks and treebanks based on tree-adjoining grammar can be done in time linear in the size of structures and in the length of queries.

PDL with intersection is not powerful enough to capture the kind of set saturation found in nonlocal MCTAG and HPSG in an intuitive way.[1] Decharge logic is an extension of PDL with intersection specially designed for this purpose. The standard logic for HPSG, which is adequate for nonlocal MCTAG too by the general inclusion result (Søgaard, 2007), as already mentioned has an undecidable model checking problem. So the main result of this paper is that decharge logic is adequate for nonlocal MCTAG and (with some qualifications) HPSG and has a low polynomial time model checking procedure.

2.2 Decharge logic

Decharge logic is a polyadic extension of deterministic PDL with intersection in the following sense. Our signatures are as usual. Our models, however, differ a bit from ordinary Kripke models.

Definition 2.3 (Semi-deterministic polyadic Kripke models)**.** A semi-deterministic polyadic Kripke model (SPKM) is a tuple $M = \langle \mathbb{W}, \{R_a \mid a \in \mathsf{Labels}\}, \mathcal{V} \rangle$ such that $\mathbb{W}$ is a set of worlds or states. Let $R_\dagger = \{(s_1, ..., s_n) \mid \forall i = 1, \ldots, n. \forall j = i+1, \ldots, n. s_i \neq s_j\}$ be the relation consisting of all tuples of worlds without multiple occurrences. Furthermore, for each $a \in \mathsf{Labels}$, $R_a \subseteq R_\dagger$ is a polyadic relation over $\mathbb{W}$. All atomic programs are

[1]Given a specific treebank, the maximum set size can be fixed. In this case, PDL with intersection may suffice as a logical query language, albeit less intuitive, but generally it is not expressive enough. Finally, such a trick is not possible in heuristics-based parsing.

required to be deterministic, i.e. whenever $\{(s, t_1, \ldots, t_n), (s, u_1, \ldots, u_m)\} \subseteq R_a$ for some $a \in \mathsf{Labels}$ then $n = m$ and $t_i = u_i$ for all $i = 1, \ldots, n$. Finally $\mathcal{V} : \mathbb{W} \rightarrow 2^{\mathsf{Atoms}}$ interprets propositional variables in worlds.

Note that labels are not associated with a particular arity. Relations may contain tuples of different lengths, since they will be used to encode set values in nonlocal MCTAG and HPSG.

Definition 2.4 (Syntax of decharge logic)**.** Formulas (ϕ, ψ) and programs (α_i) of decharge logic over the signature $\langle \mathsf{Labels}, \mathsf{Atoms} \rangle$ are defined as:

$$\begin{array}{rcl} \phi, \psi & \doteq & p \mid \phi \wedge \psi \mid \neg\phi \mid \langle\alpha\rangle(\phi_1, \ldots, \phi_n) \\ \alpha_1, \alpha_2 & \doteq & \epsilon \mid a \mid \alpha_1; a \mid \beta_1^* \mid \alpha_1 \cup \alpha_2 \mid \alpha_1 \cap \alpha_2 \\ & & \mid \ominus(\gamma, a, \alpha_3) \\ \beta_1, \beta_2 & \doteq & \epsilon \mid a \mid \beta_1 \cup \beta_2 \\ \gamma_i & \doteq & \epsilon \mid a \mid \gamma_i; a \end{array}$$

where $a \in \mathsf{Labels}$ and $p \in \mathsf{Atoms}$.

$\ominus$ is called the decharge operator. The semantics of the PDL operators are as usual, but over SPKMs, and the relation induced by the decharge operator is defined as follows:

$$\begin{array}{rcl} R_{\ominus(\alpha_1, \alpha_2, \alpha_3)} & \doteq & \{(w, v_1, \ldots, v_{j-1}, v_{j+1}, \ldots, v_n) \mid \\ & & \exists(w, w') \in R_{\alpha_1}, \exists(w', v_1, \ldots, v_n) \\ & & \in R_{\alpha_2}, \exists(w, v_j) \in R_{\alpha_3}\} \end{array}$$

$\ominus$ is a complement operator that nondeterministically removes an element from a list. Intuitively, α_1 is a pointer to somewhere in the structure, α_2 is the set value at the node that is pointed out, and α_3 the place where we put the element that has been removed. $\cap$ can then be used to place the new set.

2.2.1 Model checking

There exists a model checking procedure for decharge logic whose worst-case complexity is in $\mathcal{O}(|\phi|^2 \times |\mathbb{W}|^4)$ where ϕ is the input formula and $\mathbb{W}$ the world set of the SPKM. The proof goes as follows:

Let M be a SPKM with world set $\mathbb{W}$ and ϕ a decharge logic formula. First find all subformulas of the form $\alpha\langle\psi\rangle$ in ϕ. This can be done in time $\mathcal{O}(|\phi|)$. Then for each subformula compute the relation R_α over M. This can be done in time $\mathcal{O}(|\alpha| \times |\mathbb{W}|^4)$ by Lemma 5.4 in Søgaard and Lange (2009). Add R_α to M under a new atomic program name a_α in time $\mathcal{O}(|\mathbb{W}|^2)$ (the bound on the size of the new relations). Let M' be the resulting SPKM, and let ϕ' result from ϕ by replacing every $\langle\alpha\rangle\phi$ with $\langle a_\alpha\rangle\phi$ in a bottom-up fashion. Now $M, w \models \phi$ iff $M', w \models \phi'$, and $M', w \models \phi'$ is an instance of the model checking problem of ordinary polyadic modal logic (Blackburn et al., 2001) known to be solvable in time $\mathcal{O} = (|M'| \times |\phi'|)$ (Lange, 2006). Overall this gives an upper bound of $\mathcal{O}(|\phi|^2 \times |\mathbb{W}|^4)$ on the time needed to perform model checking for decharge logic.

2.3 Extended decharge logic

Decharge logic is not rich enough to cover all the basic constructs in HPSG (Pollard and Sag, 1994). Extended decharge logic bridges this gap (in part) without changing the worst-case complexity of the model checking problem. Formulas (ϕ, ψ) and programs (α_i) of extended decharge logic over a signature $\langle \mathsf{Labels}, \mathsf{Atoms} \rangle$ are defined as follows:

$$\begin{array}{rcl} \phi, \psi & \doteq & p \mid \phi \wedge \psi \mid \neg\phi \mid \langle\alpha\rangle(\phi_1, \ldots, \phi_n) \\ \alpha_1, \alpha_2 & \doteq & \epsilon \mid a \mid \alpha_1; a \mid \beta_1^* \mid \alpha_1 \cup \alpha_2 \mid \alpha_1 \cap \alpha_2 \mid \\ & & \alpha_1 \sqcap \alpha_2 \mid \mathbf{app}(\gamma_1, \gamma_2, \gamma_3, \gamma_4) \mid \\ & & \ominus(\gamma_1, a, \alpha_1) \\ \beta_1, \beta_2 & \doteq & \epsilon \mid a \mid \beta_1 \cup \beta_2 \\ \gamma_i & \doteq & \epsilon \mid a \mid \gamma_i; a \end{array}$$

where $a \in \mathsf{Labels}$ and $p \in \mathsf{Atoms}$. Note that two new operators are introduced, namely $\sqcap$ and **app**. $R_{\alpha \sqcap \beta}$ is defined as $\{(w, w') \mid \exists(w, v_1, \ldots, v_n) \in R_\alpha$ and $\exists(w, u_1, \ldots, u_m) \in R_\beta, \exists i, j. w' = v_i = u_j\}$, while $R_{\mathbf{app}(\alpha_1, \alpha_2, \alpha_3, \alpha_4)}$ is defined as $\{(x, \bar{y}_1, \ldots, \bar{y}_m, \bar{z}_1, \ldots, \bar{z}_n) \in R_\dagger \mid \forall i, j. \exists x', x''. (x', \bar{y}_i) \in R_{\alpha_2}, (x'', \bar{z}_j) \in R_{\alpha_4}, (x, \ldots x' \ldots) \in R_{\alpha_1}, (x, \ldots x'' \ldots) \in R_{\alpha_3}, (x, \bar{y}_1, \ldots, \bar{y}_m, \bar{z}_1, \ldots, \bar{z}_n) \in R_\dagger\}$.

Intuitively, the append operator (**app**) works this way: α_1 and α_3 are pointers to nodes in a feature structure. The operator then takes the arguments of α_2 and α_4 at the nodes to which the pointers lead, and conjoins them. In a sense, this gives us a virtual list value, a list value that is nowhere in the derivation structure; the notion of virtual lists and sets is similar to the notion of a chain in Richter (2004), albeit a very restricted one. The intersection operator is used to place this virtual list value somewhere in the structure. In extended decharge logic, lists are used as canonical representations of sets. The even richer logic in Søgaard and Lange (2009) represents all linearizations of sets in models, but has a PSPACE-complete model checking procedure.

The low polynomial time model checking procedure can be extended to this extension of decharge logic, as shown in Theorem 5.5 in Søgaard and Lange (2009). Consequently, the new operators do not add to asymptotic complexity.

3 Nonlocal multicomponent tree-adjoining grammar

Multicomponent tree-adjoining grammar (MCTAG) (Becker et al., 1991) is an extension of tree-adjoining grammar in which adjunction is simultaneous adjunction of all trees in a finite set (of fixed size) of auxiliary trees rather than just adjunction of a single tree. Tree-local and set-local MCTAG impose further restrictions on adjunction, while nonlocal MCTAG imposes no further restrictions.

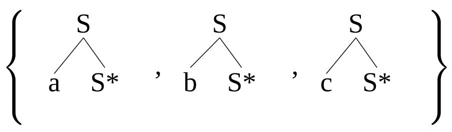
MCTAG was primarily invented to implement analyses of scrambling in languages such as German (Becker et al., 1991) and Korean (Kallmeyer and Yoon, 2004). A recent alternative to MCTAG uses tree tuples rather than sets (TT-MCTAG) (Lichte, 2007), also motivated by scrambling phenomena.

The key idea in all these analyses is to factorize the verb and its complements into different auxiliary trees that can then be permuted in derivation. For each verb with its complements a new tree set is adjoined.

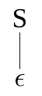
3.1 Computational complexity and generative capacity

Rambow and Satta (1992) present a proof that the fixed recognition problem of nonlocal MCTAG is NP-hard, generalized to a few restricted variants in Champollion (2007), while Søgaard et al. (2007) present a (weaker) proof of the NP-hardness of the universal recognition problem that is generalized to all variants of MCTAG. It follows from the linear upper bound on the size of derivation structures that the universal recognition problem can also be solved in nondeterministic linear space, which also implies that nonlocal multicomponent tree-adjoining languages can be recognized by linear bounded automata. Since any language that can be represented by a linear bounded automaton is context-sensitive (Landweber, 1963), it holds that nonlocal MCTAG is context-sensitive. It also follows from the result obtained in this paper, namely that model checking can be done in low polynomial time, that the universal recognition problem is in NP and thereby NP-complete. It is possible to nondeterministically guess a derivation structure linear in the length of the input string and verify it in low polynomial time.

On the other hand it is easy to prove that nonlocal MCTAG is *not* mildly context-sensitive; see also Rambow and Satta (1992). Consider the grammar with the auxiliary tree set:



and the initial tree:

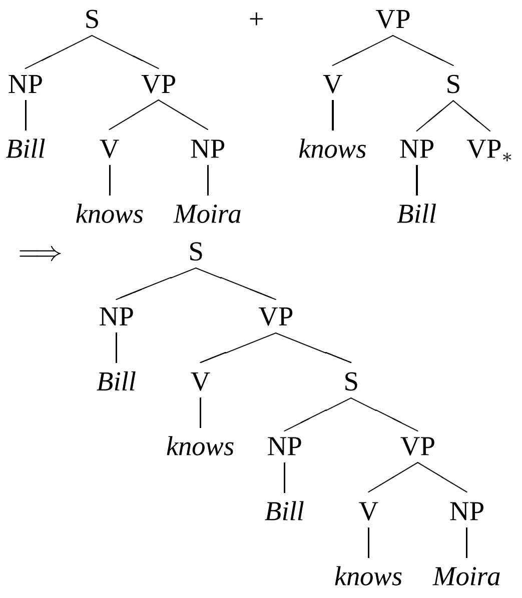


This grammar generates the MIX language which according to Marsh's conjecture is not even an indexed language. Tree-local MCTAG, on the other hand, is weakly (but not strongly) equivalent to tree-adjoining grammar and thus mildly context-sensitive.

3.2 Model-theoretic characterization

A model-theoretic version of nonlocal MCTAG in which a grammar is a set of axioms in decharge logic, and the language is the set of strings whose logical descriptions are satisfiable in conjunction with the grammar, is briefly sketched.

The first step of the reconstruction of nonlocal MCTAG in logical terms is similar to the model-theoretic characterization of tree-adjoining grammar in Keller (1993). Consider the translation of a case of adjunction in below, presented in Figure 1 in the more readable AVM notation known from HPSG and also used in Keller (1993), Blackburn and Spaan (1993) and Richter (2004), i.e. AVMs can, if we ignore the issue of underspecification for now, be seen as deterministic Kripke models (Blackburn and Spaan, 1993).



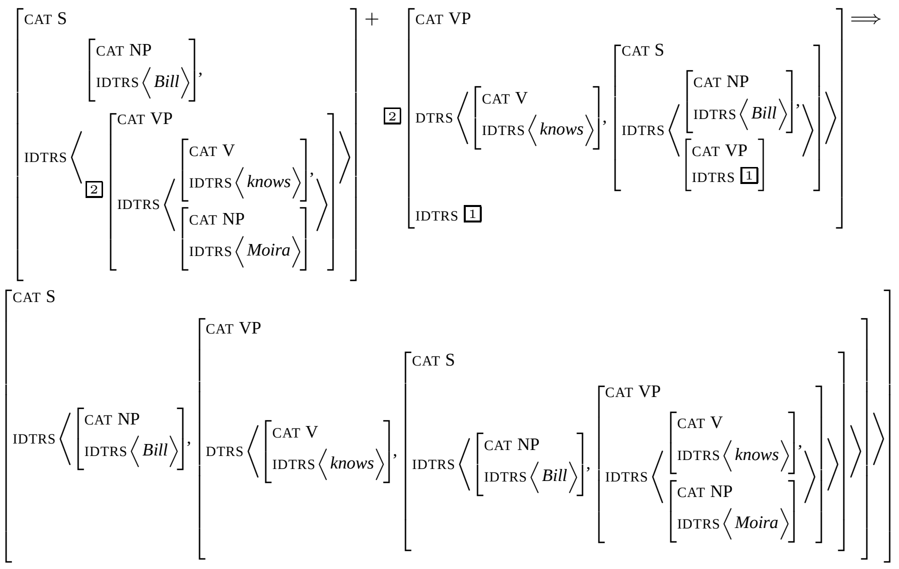


Figure 1: Adjunction in AVM notation

The idea behind the translation is that we duplicate trees. So we have an initial constituent structure embedded under IDTRS that adjunction can modify; if no adjunction takes place, the IDTRS and DTRS tree structures are unified. The axiomatization of TAG is such that every node in a model must be either a terminal node, an adjunction site or IDTRS and DTRS must be unified. See Keller (1993) for details.

The trick is now to introduce an additional feature TSET to encode sets of auxiliary trees. The decharge operator $\ominus$ is used to nondeterministically remove auxiliary trees from these sets one at a time in derivation. Saturation is ensured by the converse operator, as already described above.

4 Head-driven phrase structure grammar

HPSG (Pollard and Sag, 1994) is a popular, but very complex deep grammar theory or, perhaps more adequately, a complex deep grammar architecture. Its earliest version was unification-based, but this is no longer the case. It is, unlike nonlocal MCTAG, supposed to be model-theoretic. Consequently, logical formalizations already exist. Conventionally, an HPSG grammar is defined as a tuple $\langle\langle \mathsf{Types}, \sqsubseteq\rangle, \mathsf{Principles}\rangle$, where $\langle \mathsf{Types}, \sqsubseteq\rangle$ is the inheritance hierarchy, a finite bounded complete partial order, and $\mathsf{Principles}$ is a set of linguistic principles. The linguistic principles correspond intuitively to generative rules, but are constraints over a set of legitimate derivation structures. The inheritance hierarchy is formally simple and can be reconstructed in propositional logic (Moens et al., 1989). Consequently, the tricky part is the linguistic principles. The main challenges are set saturation, covered in extended decharge logic by the decharge operator, and union of sets. Note that set union cannot be expressed by the decharge operator.

Example 4.1. An example of a linguistic principle in HPSG that uses set union is the Nonlocal Feature Principle (Pollard and Sag, 1994):

> For each nonlocal feature, the INHERITED value on the mother is the union of the INHERITED values on the daughters minus the TO-BIND value on the head daughter.

In Pollard and Sag (1994), there are three nonlocal features on INHERITED, SLASH,QUE,REL.

4.1 Related formalizations

Reape (1994) formalizes an earlier version of HPSG in terms of a quantified hybrid logic $\mathcal{L}^{++}$. $\mathcal{L}^{++}$ is an extension of propositional logic with n-ary modalities, nominals and quantification over nominals. Nominals are a subset of the set of propositional variables that only denote singleton subsets in a model. Quantification is similar to first order logic. $\mathcal{L}^{++}$ is a polyadic version of H($\exists$). Set union is implemented in a first order theory of sets. The model checking problem is obviously PSPACE-hard.

(Hegner, 1996) defined a decidable extension of the Schönfinkel-Bernays class as a formalization of HPSG. In this logic, quantifiers or variables are typed relative to the inheritance hierarchy, and prefixes of the form $\forall_t \exists_t$ are allowed iff t, t' are incompatible types.The logic is clearly more expressive than the Schönfinkel-Bernays class, but it does not capture strong welltypedness (Carpenter, 1992). Consider, for instance, the HPSG-style strong welltypedness condition on phrases:

$$\forall x.\textit{hd-phr}(x) \rightarrow \exists y.\textit{head-dtr}(x, y)$$

saying that a headed phrase has a head daughter. The trouble is that a head daughter can itself be a headed phrase, so this condition cannot be expressed in the logic of (Hegner, 1996). In general, no decidable standard prefix-vocabulary class of first order logic characterizes the deterministic, connected and strongly welltyped structures used in HPSG (Søgaard, 2007).

The logic proposed in Richter (2004), RSRL, is an extension of description logic with global quantification similar to what can be obtained in PDL with intersection by $(a_1 \cup \ldots \cup a_n)^*$ with $\mathsf{Labels} = \{a_1, \ldots, a_n\}$, i.e. the master modality. RSRL is much more complex than PDL with intersection, though. In fact its model checking problem is known to be undecidable. Sets are still decomposed as in the first order theory of sets.

The relevant complexity results (and proofs thereof) for $\mathcal{L}^{++}$ and RSRL are presented in Søgaard (2007). PSPACE-hardness of model checking $\mathcal{L}^{++}$ and RSRL can be proven by reduction of Geography (Garey and Johnson, 1979), the undecidability of satisfiability by the tiling problem, and the undecidability of model checking RSRL can be proven by the Post correspondence problem.

The main difference between decharge logic and $\mathcal{L}^{++}$ and RSRL is that sets are first class citizens in decharge logic, i.e. sets of tuples denoted by relations of variable arity. This complicates the logical machinery in some respects, but means that first order machinery that leads to PSPACE-complete model checking, can be avoided.

4.2 Model-theoretic characterization

Here is possible formalization of the Nonlocal Feature Principle in Example 4.1 in extended decharge logic in the feature geometry in Pollard and Sag (1994) (w. *hd-dtr* = headed daughter):

$$\begin{array}{lll} \textit{hd-phr} & \rightarrow & \langle \mathtt{elem}(\ominus(\epsilon, \pi, \mathtt{dtrs;hd\text{-}dtr;synsem;} \\ & & \mathtt{nonlocal;to\text{-}bind;f}) \cap \\ & & \mathtt{synsem;nonlocal;} \\ & & \mathtt{inherited;f}) \rangle \top \end{array}$$

with $\langle \mathtt{elem}(\pi \cap \mathbf{app}(\mathtt{all\text{-}dtrs}, \mathtt{synsem;nonlocal;inherited;f}, \epsilon, \epsilon)) \rangle \top$ and $\pi \in \mathsf{Labels}$. F is a placeholder for the nonlocal features SLASH,QUE,REL.

See Søgaard and Lange (2009) for more examples. Our qualifications, mentioned multiple times in the above, are also made precise in Søgaard and Lange (2009). There are a few somewhat controversial HPSG principles, i.e. the Trace Principle and the Binding Theory, that do not seem to be definable in extended decharge logic.

5 Conclusion

This paper introduced a polyadic dynamic logic called decharge logic and an extension thereof to provide query languages for context-sensitive treebanks, e.g. treebanks with non-projective dependency structures, incl. the Prague Dependency Treebank and the Danish Dependency Treebank, the LinGO Redwoods Treebank and the BulTreeBank.

Common query tools for treebanks include CorpusSearch, ICECUP III (Wallis and Nelson, 2000) and TGrep2, but as pointed out by Kepser (2004) the query languages used in these tools are not even expressive enough to perform arbitrary queries on context-free derivations. They are, according to Kepser (2004), all subsumed by the existential fragment of first order logic. Other more expressive logics that have been introduced to characterize context-sensitive grammar formalisms (Reape, 1994; Richter, 2004) have model checking procedures with exponential runtime. It was shown that decharge logic and its extension have low polynomial time model checking procedures. The two logics thus make querying context-sensitive treebanks feasible.

Using decharge logics for querying treebanks is similar to using more common query tools. Say the following is a sentence in a treebank in TGrep2 input format:

```
(TOP (NP (NP (NN Budget)) (VP (VBD
          increased))))
```

In TGrep2, the following three lines of text are examples of queries:

```
(i)    NP ≺≺ NN
(ii)   NP ≺ NN
(iii)  NP !≺ NN
```

(i) matches all nodes labeled by NP that dominate a node labeled by NN (2 nodes); (ii) matches all nodes labeled by NP that immediately dominate a node labeled by NN (1 node); and (iii) matches all nodes labeled by NP that do not immediately dominate a node labeled by NN (1 node). The queries correspond to the following formulas in decharge logic:

(i') $np \wedge \langle(\mathtt{down};\mathtt{right}^*)^*\rangle nn$
(ii') $np \wedge \langle\mathtt{down};\mathtt{right}^*\rangle nn$
(iii') $np \wedge \neg\langle\mathtt{down};\mathtt{right}^*\rangle nn$

The query tools thus essentially model check the derivation structure wrt. some formula ϕ and output the set of nodes (states) that satisfy ϕ.

Decharge logic and its extension can also be used to verify heuristic parses.

References

Tilman Becker, Aravind Joshi, and Owen Rambow. 1991. Long-distance scrambling and tree adjoining grammars. In *EACL'91*, pages 21–26, Berlin, Germany.

Patrick Blackburn and Edith Spaan. 1993. A modal perspective on the computational complexity of attribute value grammar. *Journal of Logic, Language and Information*, 2(2):129–169.

Patrick Blackburn, Maarten de Rijke, and Yde Venema. 2001. *Modal logic*. Cambridge University Press, Cambridge, England.

Matthias Buch-Kromann. 2007. Computing translation units and quantifying parallelism in parallel dependency treebanks. In *ACL'07, Linguistic Annotation Workshop*, pages 69–76.

Bob Carpenter. 1992. *The logic of typed feature structures*. Cambridge University Press, Cambridge, England.

Lucas Champollion. 2007. Lexicalized non-local MCTAG with dominance links is NP-complete. In *MOL'07*, Los Angeles, California.

Michael Garey and David Johnson. 1979. *Computers and intractability*. W. H. Freeman & Co., New York, New York.

Eva Hajičová, Jan Hajič, Barbora Hladká, Martin Holub, Petr Pajas, Veronika Řezníčková, and Petr Sgall. 2001. The current status of the Prague Dependency Treebank. In *LNCS 2166*, pages 11–20. Springer, Berlin, Germany.

Stephen Hegner. 1996. A family of decidable feature logics which support HPSG-style set and list constructions. In *LACL'96*, Berlin, Germany.

Aravind Joshi and Yves Schabes. 1997. Tree-adjoining grammars. In *Handbook of Formal Languages*, volume 3, pages 69–124. Springer, Berlin, Germany.

Laura Kallmeyer and Sin-Won Yoon. 2004. Tree-local MCTAG with shared nodes. In *TALN'04*, Fes, Marocco.

Bill Keller. 1993. *Feature logics, infinitary descriptions and grammar*. CSLI Publications, Stanford, California.

Stephan Kepser. 2004. Querying linguistic treebanks with monadic second-order logic in linear time. *Journal of Logic, Language and Information*, 13:457–470.

Marco Kuhlmann and Mathias Möhle. 2007. Mildly context-sensitive dependency languages. In *ACL'07*, pages 160–167, Prague, Czech Republic.

Peter Landweber. 1963. Three theorems on phrase structure grammars of type 1. *Information and Control*, 6(2):131–136.

Martin Lange. 2006. Model checking propositional dynamic logic with all extras. *Journal of Applied Logic*, 4:39–49.

Timm Lichte. 2007. An MCTAG with tuples for coherent constructions in German. In *FG'07*, Dublin, Ireland.

Wolfgang Maier and Anders Søgaard. 2008. Treebanks and mild context-sensitivity. In *FG'08*, Hamburg, Germany.

Marc Moens, Jo Calder, Ewan Klein, Mike Reape, and Henk Zeevat. 1989. Expressing generalizations in unification-based grammar formalisms. In *EACL'89*, Manchester, England.

Joakim Nivre. 2006. Constraints on non-projective dependency parsing. In *EACL'06*, pages 73–80, Trento, Italy.

Stephan Oepen, Kristina Toutanova, Stuart Shieber, Cristopher Manning, Dan Flickinger, and Thorsten Brants. 2002. The LinGO Redwoods Treebank. In *COLING'02*, pages 1253–1257, Taipei, Taiwan.

Carl Pollard and Ivan Sag. 1994. *Head-driven phrase structure grammar*. The University of Chicago Press, Chicago, Illinois.

Owen Rambow and Giorgio Satta. 1992. Formal properties of nonlocality. In *TAG+'92*, Philadelphia, Pennsylvania.

Mike Reape. 1994. A feature value logic with intensionality, nonwellfoundedness and functional and relational dependencies. In *Constraints, language and computation*, pages 77–110. Academic Press, San Fransisco, CA.

Frank Richter. 2004. *A mathematical formalism for linguistic theories with an application in head-driven phrase structure grammar*. Phd thesis (2000), Universität Tübingen, Tübingen, Germany.

Kiril Simov, Petya Osenova, Alexander Simov, and Milen Kouylekov. 2004. Design and implementation of the Bulgarian HPSG-based treebank. *Research on Language and Computation*, 2(4):495–522.

Anders Søgaard and Martin Lange. 2009. Polyadic dynamic logics for HPSG parsing. *Journal of Logic, Language and Information*, 18(2):159–198.

Anders Søgaard, Timm Lichte, and Wolfgang Maier. 2007. On the complexity of linguistically motivated extensions of tree-adjoining grammar. In *RANLP'07*, Borovets, Bulgaria.

Anders Søgaard. 2007. *Complexity, expressivity and logic of linguistic theories*. Ph.D. thesis, University of Copenhagen, Copenhagen, Denmark.

Sean Wallis and Gerald Nelson. 2000. Exploiting fuzzy tree fragment queries in the investigation of parsed corpora. *Literary and Linguistic Computing*, 15(3):339–361.