An efficient Trie for binding (and movement)

Cristiano Chesi
NETS - IUSS
P.zza Vittoria 15
I-27100 Pavia (Italy)
cristiano.chesi@iusspavia.it

Abstract

English. Non-local dependencies connecting distant structural chunks are often modeled using (LIFO) memory buffers (see Chesi 2012 for a review). Other solutions (e.g. *slash* features in HPSG, Pollard & Sag 1994) are not directly usable both in parsing and in generation algorithms without undermining an incremental left-right processing assumption. Memory buffers are however empirically limited and psycholinguistically invalid (Nairne 2002). Here I propose to adopt *Trie* memories instead of *stacks*. This leads to simpler and more transparent solutions for establishing non-local dependencies both for *wh-* argumental configurations and for anaphoric pronominal coreference.

Italian. *Nell'implementazione di dipendenze non locali che mettano in connessione due costituenti arbitrariamente distanti in una struttura frasale, spesso si è ricorsi all'uso di memorie a pila (LIFO; si veda Chesi 2012 per una panoramica sul tema). Le altre soluzioni proposte (e.g. tratti* slash *in HPSG, Pollard & Sag 1994) non risultano implementabili in modo trasparente, né in generazione né in parsing, con algoritmi che tengano conto del requisito di incrementalità del processamento. Tuttavia, viste le limitazioni psicolinguistiche ed empiriche delle memorie a pila (Nairne 2002), qui si propone di adottare memorie di tipo* Trie *per codificare i tratti rilevanti nello stabilire dipendenze non locali nel caso di strutture che impiegano elementi wh- argomentali e nel legamento pronominale anaforico.*

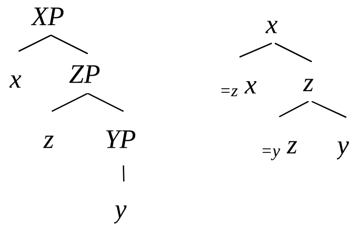
1 Introduction

Relations among structural chunks in a sentence are not always resolvable using strictly local dependencies. This is the case of argumental *wh*-items in languages like English (or Italian), where the argument and the predicate can be arbitrarily distant, (1).a. Another case of non-local dependency is pronominal coreference that in some cases can also be cross-sentential, (1).b-b', (1).b-b''.

(1) a. [$_X$ Cosa] (tu) pensi che (io) [$_Y$ mangi_]?
what (you) think that (I) eat$_{SUBJ\text{-}1P\text{-}Sing}$
what do you think I eat?
b. [$_X$ Gianni]$_i$ saluta [$_Z$ Mario]$_j$.
G. says hello (to) M.
b'. Poi *pro* $_i$ [$_Y$ si]$_i$ lava.
then (he) himself$_j$ washes.
then he washes himself
b''. Poi *pro* $_i$ [$_Y$ lo]$_j$ lava.
then (he) him$_j$ washes.
then he washes him

From a purely structural perspective, the chunks *X* and *Y* enter a non-local dependency relation when some material *Z* intervenes between them. A long tradition of different approaches addressed this issue from different perspective (see Nivre 2008, for instance, for a comparison among Stack-based and List-based algorithms in parsing). Most of the time these approaches rely on transformations of the grammar into a deductive system for both parsing (Shieber et al. 1995) and generation (Shieber 1988). A loss of transparency with respect to the linguistic intuitions that motivated a specific grammatical formalism is then at issue. Here I will argue in favor of a simple derivational and deterministic perspective in which phrases are considered the result of the recursive application of structure building operations (Chomsky 1995). In its simplest format, classic structural descriptions, (2).a, reduce to lexicalized trees, (2).a', in which *x* and *z* creates a constituent (get *merged*) either if *x selects z* ($_{=Z}$ x, in Stabler's 1997 formalism) or the way around ($_{=X}$ z). Leaves are linearly ordered and constituents labels reduce to the selecting lexical items.

(2) a. a'.



By definition, *x* and *y* cannot enter a local dependency whenever an intervening item *z* blocks a local selection between *x* and *y*. There are cases, however, in which *x* and *y* should enter a local selection relation: in (1).a, *x* receives a thematic role from *y*, hence *y* should select *x* according to the uniformity of theta-role assignment hypothesis (Baker 1988). In this case, a non-local dependency must be established. Implementing the *movement* metaphor (Stabler 1997) in top-down terms, Chesi (2017) proposes that an item *x* is *moved* into a Last-In-First-Out (LIFO) memory buffer (M) whenever it brings into the computation features that are unselected: if a (categorial) feature $_X$ is selected and a lexical item *a* brings $_X$ but also $_Y$ from the lexicon (i.e. [$_{X\ Y}$ *a*]]), then *a* gets merged (i.e. [~~$_X$~~[~~$_X$~~ $_Y$ *a*]]), but the unselected feature [$_Y$ *(a)*] is *moved* into the last position (the most prominent one) of the M-buffer. As soon as a feature $_Y$ will be selected ($_{=Y}$), the last item in the memory buffer, if bearing the relevant $_Y$ category, will be remerged in the structure before any other item from the lexicon, the satisfying a local selection requirement. After its re-merge, the item is removed from the M-buffer.

This paper proposes a theoretical solution for simplifying this memory-based approach without losing any descriptive adequacy: here I will do away with the buffer idea (and, as a consequence, with the LIFO restrictions) by postulating a memory Trie (Fredkin 1960) based on the features merged in the structure during the derivation. I will show that this solution is psycholinguistically more plausible than LIFO buffers used so far and computationally sound.

1.1 Implementing non-local dependencies

Phase-based Minimalist Grammars (PMG, Chesi 2007) express top-down, left-right derivations that can be used directly both in generation and in parsing (Chesi 2012, see Chesi 2017 for some advantages for predicting difficulty in parsing). Non-local dependencies of the (1).a kind are established whenever a constituent lexicalizes an expected feature but also brings into the structure unexpected features that should be selected later on, in order for the sentence to be grammatical. This is implemented using PMGs able to deal with non-local dependencies as discussed below.

1.1.1 A simpler PMG formalization

PMGs are lexicalized grammars in which structure building operations are included in the grammatical formalism (Chesi 2007 and Collins & Stabler 2016 for a recent formalization of MGs). Unlike other formalisms (e.g. CFGs, HPSGs, TAGs or CCGs) PMGs do not simply express a declarative knowledge but also a deterministic procedure (Marcus 1980, Shieber 1983) that explicitly produces, step-by-step, a full derivation which should be common both in parsing and in generation (Momma & Phillips 2018). Below the basic definitions representing a simplified formalization of the crucial components of a PMG: categories, feature structures, lexical items, structure building operations and their triggers.

Definition 1 *A* category *is a morpho-syntactic feature with a(n optional) value specification:* [$_{cat(:value)}$]. *Each derivation starts with a (default) projection of a specific category (*phase edge*).*

Even if this is not strictly necessary here, for simplicity, categories will be divided into *functional* (e.g. [$_{D:definite}$] or simply [$_D$] for a definite determiners/articles), *phase edges* (functional categories introducing a new phase, in the sense of Chomsky 2008), and *lexical* (e.g. nominal or verbal categories, namely the sole categories, a part from the default root selection that starts the derivation, entitled to select new phase edges).

Definition 2 *A* lexical item *is a ordered feature structure (Attribute-Value Matrix) encoding* phonetic *(/phon),* semantic *(#sem) and* category *features:* [$_{cat\_1(:v\_1)\ \dots\ cat\_n(:v\_n)}$ *#sem /phon*]

Neither phonetic (instruction for pronouncing a lexical item) nor semantic features (instruction for interpreting the item both lexically, e.g. WordNet synset, Miller 1995, and compositionally, e.g. specification of a functional application, Heim & Kratzer 1998) will be discussed here. I will use simpler entries like [$_N$ man] (by default: *num:sg, gen:male*). Certain items might be optionally specified for some categories: [$_{(F)\ X}$ ...] indicates that the $_F$ category (*focus*) can be present or not (this has semantic and a derivational impact).

Definition 3 *A* phrase structure *is a hierarchical feature structure combining categories and lexical items; a phrase structure is* fully lexicalized *iff each* category *in it is associated to a* lexical item.

Definition 4 *An* edge category *is the most prominent feature, namely the target of any* structure building operation*;*

By default, edge categories (that will be *underlined* below) are the left-most feature of any lexical item and the right-most feature of any unlexicalized phrase structure. If an optional category is present, this is the edge of the lexical item.

Definition 5 Structure building operations *are functions taking in input phrase structures and returning modified phrase structures.* Merge, Move *and* Expect *are structure building operations.*

Definition 6 Merge *is a binary* structure building operation *that unifies the edge categories in a* phrase structure *and a* lexical item*:*
Merge([X ... [Y]], [Y ... lex]) → [X ... [~~Y~~[~~Y~~ ... lex]]]

Definition 7 Expect *takes as input a select feature and introduce it in the structure:* [=X] → [~~=X~~ [X]]

An *expectation/expansion* is then a lexically or categorically encoded select feature; whenever *categories* in the lexicon are specified for select features (e.g. [X =Z]), those select features must be expanded after lexicalization (i.e. first *merge*: [~~X~~[~~X~~ ...] =Z], then *expect*: [~~X~~[~~X~~ ...] ~~=Z~~[Z]])

Definition 8 *An* unexpected category *is any* unselected feature *introduced in the derivation by merging a lexical item bearing both the expected feature(s) and unexpected one(s).*

e.g. merge([... [Y]], [Y Z ... *a*]) → [... [~~Y~~[~~Y~~ Z *a*]]] *Unselected item after* merge: [~~Y~~ Z *(a)*]

Definition 9 Move *is the operation storing items with* unexpected features *in a LIFO M(emory)-buffer*. [... [~~Y~~[~~Y~~ Z *a*]]] → M:<[~~Y~~ Z *(a)*]>

Since the lexical items is already pronounced, phonetic features will not be re-merged, hence *(a)*.

Definition 10 M-buffer *must be empty at the end of the derivation.* Lexical items *stored in the memory buffer must be (re-)merged, as soon as a compatible* expectation *is introduced, before any other* lexical item.

1.1.2 A toy grammar exemplifying processing of non-local dependencies

Given the (simplified) lexicon in (3), the generation of (1).a proceeds as indicated in (4):

(3) simplified lexicon for generating and parsing sentences in (1):

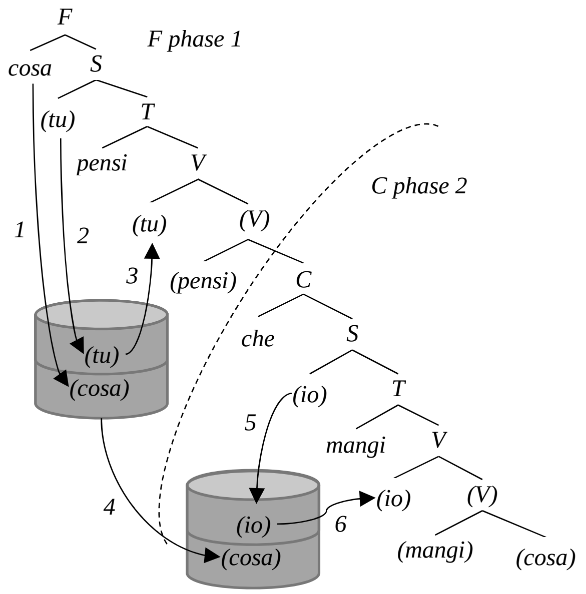
Lexicon
[(S) D N anim G./M.], [F D gen:fem N cosa], [D:reflex si$_x$], [(S) D pers:1 case:nom N (io)], [(S) D pers:2 case:nom N (tu)], [C che], [C poi], [Pers:1 T V mangi =D:case:nom =D:case:acc], [pers:2 T V pensi =D:case:nom =C], [pers:3 T V lava =D:reflex:anim =D:case:acc]
Categories
Phase edges (functional categories): [C =S], [F =S], [D =N] *Other functional categories*: [S =T], [T =V] *Lexical categories*: [N], [V]

(4) Generation of (1).a
Cosa$_i$ (tu) pensi che (io) mangi _$_i$?

1. [F =S] *(default root phase edge expectation)*
2. [~~F~~[~~F~~ D ... cosa] =S] *(merge)*
3. [~~F~~[~~F~~ D ... cosa] =S] M<[D ... (cosa)]> *(move)*
4. [~~F~~[~~F~~ D ... cosa] ~~=S~~[S =T]] *(expect)*
5. [~~F~~[~~F~~ D ... cosa] ~~=S~~[~~S~~[~~S~~ D ... (tu)] =T]] *(merge)*
6. [~~F~~[~~F~~ D ... cosa] ~~=S~~[~~S~~[~~S~~ D ... (tu)] =T]] *(move)*
M<[D ... (cosa)], [D ... (tu)] >
7. [~~F~~[~~F~~ D ... cosa] ~~=S~~[~~S~~[~~S~~ D ... (tu)] ~~=T~~[T =V]]] *(expect)*
8. [~~F~~[~~F~~ D ... cosa] ~~=S~~[~~S~~[~~S~~ D ... (tu)] ~~=T~~[~~T =V~~[~~T V~~ pensi =D =C]]]] *(merge)*
9. ... [... pensi ~~=D~~[D =N] =C] *(expect)*
10. ... [... pensi ~~=D~~[~~D =N~~ [~~D~~ ... (tu)]] =C] *(merge from M)*
11. ... ~~=C~~[C =S]] *(expect)*
12. ... ~~=C~~[~~C~~[~~C~~ che] =S]] *(merge)*
13. ... ~~=C~~[~~C~~[~~C~~ che] ~~=S~~[S =T]]] *(expect)*
14. ... ~~=C~~[~~C~~[~~C~~ che] ~~=S~~[~~S~~[~~S~~ D... (io)] =T]]] *(merge)*
15. ... ~~=C~~[~~C~~[~~C~~ che] ~~=S~~[~~S~~[~~S~~ D... (io)] =T]]] *(move)*
M<[D ... (cosa)], [D ... (io)] >
16. ... ~~=C~~[~~C~~[~~C~~ che] ~~=S~~[~~S~~[~~S~~ D... (io)] ~~=T~~[T =V]]]] *(expect)*
17. ... [~~T =V~~ [~~... T V~~ mangi =D =D]] *(merge)*
18. [... mangi ~~=D~~[D =N] =D]] *(expect)*
19. [... mangi ~~=D~~[~~D =N~~ [~~D~~ ... (io)]] =D]] *(merge from M)*
20. [... mangi ~~=D~~[~~D =N~~ [~~D~~ ... (io)]] ~~=D~~[D =N]]] *(expect)*
21. [... mangi ~~=D~~[~~D =N~~ [~~D~~ ... (io)]] ~~=D~~[~~D =N~~ [~~D~~ ... (cosa)]]]] *(merge from M)*

The sentence is grammatical iff the M-buffer is emptied by the end of the derivation and no expectations are pending. The structural description (to be considered as the history of the derivation, which is also a representation of all the useful structural restrictions) is represented in (5). The features triggering *Merge*, *Move* and *Expect* are omitted in the tree for simplicity (refer to (3) and (4) for the full set of features and for the step by step derivation). Notice that "vacuous" movements of the null subjects in Italian is the main difference between generation and parsing: in parsing, an underspecified (for number and person) null subject is postulated then re-merged (unified with the relevant feature values) after the verbal morphology has been analyzed. Moreover, using the toy grammar in (3), 3 expectations could initialize the parsing (*C*, *F* and *D*), but only the first one (*F*) would result compatible with the "cosa pensi" incipit of the sentence (cf. Earley 1977).

(5) Tree diagram summarizing the step-by-step derivation in (4)



1.2 Non-local pronominal coreference

The same strategy cannot be used for pronominal binding, e.g. (1).b-b', since:

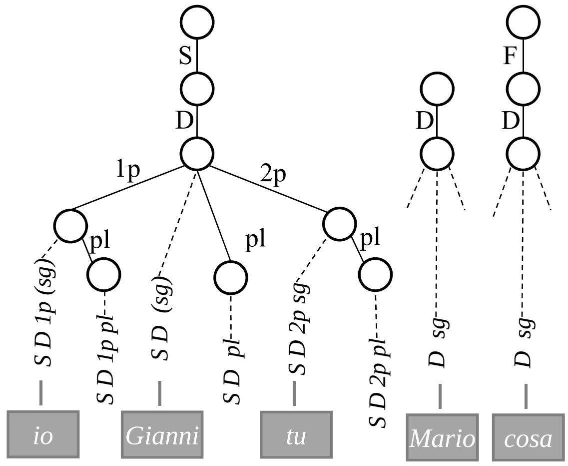
i. LIFO memory buffers are populated only for a short amount of time, then got emptied as soon as the relevant features are selected; referential items should stay in memory longer after the item has been selected for capturing also (cross-sentential) binding effects.
ii. LIFO structure is not suitable to capture crossing dependencies like the one in (1).b-b'.

Problem i. has been discussed and resolved both by Schlenker (2005) and Bianchi (2009) by postulating "referential buffers" of the kind we discussed in §1.1 in which referential NPs are stored and used without being removed for binding (i.e. coindexing) in anaphoric items. Bianchi (2009) shows how local and global referential buffers are sufficient to capture violation of binding principles: local buffers are phase-specific, hence nested phase buffers are inaccessible from higher phase-buffers, higher phase-buffers are accessible from lower phases, while a global referential buffer is accessible by all phases. With this distinction, Principle C effects (rephrasing Chomsky 1981, a pronoun cannot be co-referent with a non-pronominal that it c-commands: "He said that Bill is funny". He ≠ Bill) is the result of the application of a non-redundancy principle, favoring the usage of a anaphor instead of a referential expression that would re-insert a referential item already present in the referential buffer. Bianchi (2009) also notices that for retrieving the correct referent from a referential buffer we need to depart from the LIFO structure assumed so far.

2 Trie memories for capturing non-local dependencies

One way to implement Bianchi's idea (§1.2) in an efficient way is to use *Trie* memories. Tries (from re*trie*val), in their simplest form, are hierarchical, acyclic data structures that guarantee fast insertion, search and deletion of information (Fredkin 1960). Tries are often used in parsing for efficient encoding of phrase structures (Leermakers 1992 and Moore 2000 a.o.). Indeed, more efficient formats for representing, for instance, CFG phrase rules exist: *Minimized FSAs*, compared to *Tries*, perform generally better (Klein & Manning 2001). Here I will argue that, despite their lower performance compared to other phrase structure transformations, they better support correct empirical predictions both in case of coreferential binding and *wh*- movement, so they are worth to be considered both for empirical and psycholinguistic reasons. The original part of this proposal is related to the storage, in Tries format, of referential features encoded in the phrase structure built so far as indicated below (root node omitted):

(6) Trie memory fragment



Each referential NP is identified by a specific path starting from the root and reaching one leaf of the common Trie representing in a compact way all the relevant features related to any referential item inserted in the derivation. If "you" is merged in the structure as a subject, its root would be the "S" (topic) feature; "cosa" would be identified by the path F-D (3rd person being the default person, or no person, Sigurdsson 2004 and singular the default number); "io" would be S-D-1p, "tu" S-D-2p, "Gianni" S-D and "Mario" simply D (other irrelevant features being omitted for clarity). Few interesting facts are worth highlighting here:

1. Two NPs will be distinct if and only if a distinct path identifies them: with such a feature structure, "cosa" and "casa" would be undistinguishable; for separating the two, extra features must be added to the Trie (e.g. *animacy*);
2. The more similar a path, the faster the insertion in memory would be, but the easier it would also be to confound them at retrieval: storing "tu" after "voi" would be faster than storing "io" after "tu"; similarly, confounding "tu" with "voi" is expected to be easier than confounding "tu" with "io", though the number of features stored is the same;

It is clear that the fragment in (6) must be expanded including "semantic" features like *animacy*, *mass*/*countable* etc. that can be selected by the relevant predicate then creating distinct paths. Nevertheless, these two facts are already sufficient to subsume the similarity effects discussed in Chesi (2017) without relying to memory stacks.

2.1 Capturing pronominal coreference

An anaphoric item, for receiving its correct co-referent binding index, triggers an inspection of the features that qualify the items in memory as good binders, namely *topics* matching *person*, *number* and *gender* features. In (1).b-b' and (1).b-b'' a (third person, in this case) null subject is (always) used anaphorically in Italian, then, in order to be correctly interpreted it must be co-referent with a 3rd person, animate, singular, male binder. This would be only compatible with "Gianni" which is first merged in a topic (S) position and it has all the relevant features. Even though "G" shares any other feature with the direct object "Mario", its topic insertion position is crucial from selecting G instead of M. The Trie idea then supports the correct retrieval forcing distinct traversal starting with the highest feature encoded. This is much more efficient than revisiting LIFO assumptions. Notice also that this does not overgenerate: according to the binding principles, an anaphor "si" and not a "pronoun", should be co-indexed in its "local" domain. This is obtained by letting "si" look for the topic encoded feature while "lo" would inspect only compatible, non-locally topicalized, items (e.g. "M" in (1).b-b'').

2.2 Capturing movement in general

While referents in this Trie are not removed once an item is retrieved (but possibly receive a boost in its accessibility, Lewis & Vasishth 2005), a movement-based dependencies need to remove the relevant item after remerge. Here I propose to use the very same Trie representation, (6), and mark the "unexpected" features identifying an unselected item. Remember that in order to remerge the correct item, the features cued by the selecting head must be selected and a distinct path should be found in the Trie: steps 10 and 19 in (4) require a specific set of features to be retrieved that in the Trie correspond to the path D-2p and D-1p respectively. This path identifies uniquely the item "tu" and "io", while another item ("cosa", D-sg) is stored in memory. Without need of a LIFO structure we can then retrieve effectively the correct item without confusion, then removing the "unexpected" marks from the features for the unique path identifying the remerged item just retrieved.

3 Conclusion

In this paper, I presented a revision of the memory buffer used for parsing and generation in PMGs: instead of using a classic LIFO memory, proved to be sufficient to capture locality effects (Friedmann et al. 2009) when "similar" NPs are processed (Warren & Gibson 2005, Chesi 2017), but not fully plausible from a psycholinguistic perspective (no serial order seems to be relevant at retrieval, Nairne 2002, as we saw also in case of pronominal binding), I defined a Trie memory replacement, based on feature hierarchies sensitive to the structural insertion point of the memorized item. This prevents order of insertion from being strictly relevant at retrieval, without losing any ability to discriminate the correct items to be recalled for establishing a relevant (non-local) structural dependency both in thematic role assignment or anaphoric binding contexts. The Trie structure here proposed is clearly a bit simplistic, though based on a relevant evidence suggesting that person features are "higher" in the structure than "number" features (Mancini et al. 2011). Other (semantic) features should be included (e.g. animacy) as well as prosodic/salience markers (Topic, New Information/Contrastive Focus, Kiss 1998) that clearly play a role in making salient (i.e. unique in a Trie) a specific item, possibly relating the "fluctuation" of prominence of items stored in memory (Lewis & Vasishth 2005) to precise structural proprieties.

Reference

Baker, M. C. 1988. *Incorporation: A theory of grammatical function changing*. Chicago: University of Chicago Press

Bianchi, V. 2009. A note on backward anaphora. *Rivista di Grammatica Generativa*, 34, 3-34.

Chesi C. 2017. Phase-based Minimalist Parsing and complexity in non-local dependencies. *Proceedings of CLiC-it 2017*. CEUR workshop proceedings, ROMA:CEUR. Rome, 11-13 Dec 2017, doi: urn:nbn:de:0074-2006-4

Chesi, C. 2007. An introduction to Phase-based Minimalist Grammars: why move is Top-Down from Left-to-Right. *Studies in Linguistics*, 1, 49-90.

Chesi, C. 2012. *Competence and Computation: toward a processing friendly minimalist Grammar*. Padova: Unipress.

Chomsky, N. 1981. *Lectures on government and binding: The Pisa lectures*. Berlin: Walter de Gruyter.

Chomsky, N. 1995. *The Minimalist Program*. Cambridge (MA): MIT Press.

Chomsky, N. 2008. On phases. In C. Freidin, P. Otero, M. L. Zubizarreta (eds.) *Foundational issues in linguistic theory: Essays in honor of Jean-Roger Vergnaud*. MIT Press.

Collins, C., & E. Stabler. 2016. A formalization of minimalist syntax. *Syntax*, 19, 1: 43-78.

Earley, J. 1970. An efficient context-free parsing algorithm. *Communications of the Association for Computing Machinery*, 13(2), February.

Fredkin, E., 1960. Trie memory. *Communications of the ACM*, 3(9), pp.490-499.

Friedmann, N., Belletti, A., & Rizzi, L. 2009. Relativized relatives: Types of intervention in the acquisition of A-bar dependencies. *Lingua*, 119(1), 67-88.

Heim I. & A. Kratzer. 1998. *Semantics in generative grammar*. Oxford: Blackwell.

Kiss, K. É. 1998. Identificational focus versus information focus. *Language*, 74(2), 245-273.

Klein, D., & Manning, C. D. 2001. Parsing with treebank grammars: Empirical bounds, theoretical models, and the structure of the Penn treebank. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics* (pp. 338-345). Association for Computational Linguistics.

Leermakers, R. 1992. A recursive ascent Earley parser. *Information Processing Letters*, 41:87–91.

Lewis, R. L., & Vasishth, S. 2005. An activation-based model of sentence processing as skilled memory retrieval. *Cognitive science*, 29(3), 375-419.

Mancini, S., Molinaro, N., Rizzi, L., & Carreiras, M. 2011. A person is not a number: Discourse involvement in subject–verb agreement computation. *Brain research*, 1410, 64-76.

Marcus, M. P. 1980. *Theory of syntactic recognition for natural languages*. MIT press.

Miller, G. A. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11), 39-41.

Momma, S., & Phillips, C. (2018). The relationship between parsing and generation. *Annual Review of Linguistics*, 4, 233-254.

Moore R. C. 2000. Improved left-corner chart parsing for large context-free grammars. In *Proceedings of the Sixth International Workshop on Parsing Technologies*.

Nairne, J. S. 2002. The myth of the encoding-retrieval match. *Memory*, 10(5-6), 389-395.

Pollard, C. and Sag, I.A., 1994. *Head-driven phrase structure grammar*. University of Chicago Press.

Schlenker, P. 2005. Non-redundancy: Towards a semantic reinterpretation of binding theory. *Natural Language Semantics*, 13(1), 1-92.

Shieber, S. M. 1983. Sentence disambiguation by a shift-reduce parsing technique. In *Proceedings of the 21st annual meeting on Association for Computational Linguistics* (pp. 113-118). Association for Computational Linguistics.

Shieber, S. M. 1988. A uniform architecture for parsing and generation. In *Proceedings of the 12th conference on Computational linguistic*s. Vol. 2 (pp. 614-619). Association for Computational Linguistics.

Shieber, S. M., Schabes, Y., & Pereira, F. C. 1995. Principles and implementation of deductive parsing. The Journal of logic programming, 24(1-2), 3-36.

Sigurdsson, H. A. 2004. The syntax of person, tense and speech features. *Italian Journal of Linguistics*, 16, 219-251.

Stabler, E. 1997. Derivational minimalism. In *International Conference on Logical Aspects of Computational Linguistics* (pp. 68-95). Springer, Berlin, Heidelberg.

Stabler, E. 2013. Two Models of Minimalist, Incremental Syntactic Analysis, *Topics in Cognitive Science*, 5:611–633, doi:10.1111/tops.12031.

Van Dyke, J. A., & McElree, B. 2006. Retrieval interference in sentence comprehension. *Journal of Memory and Language*, 55(2), 157-166.

Warren, T., & Gibson, E. 2005. Effects of NP type in reading cleft sentences in English. *Language and Cognitive Processes*, 20(6), 751-767.