Construction-Driven Language Processing

Jerry T. Ball (jerry.ball@mesa.afmc.af.mil)
Air Force Research Laboratory
6030 S. Kent St., Mesa, AZ 85212

Abstract

Construction Grammar is an emerging linguistic theory based on the notion of *constructions*—linguistic representations of form, function and meaning. The key insights of Construction Grammar are beginning to have a significant impact on other linguistic formalisms. However, to date, Construction Grammar has had little impact on research in language processing. This paper describes an approach to language processing during comprehension based on the *activation*, *selection*, and *integration* of *constructions* corresponding to the linguistic input. Whereas activation is based on parallel spreading activation, selection and integration rely on serial processing combined with a mechanism of *context accommodation*—a cognitively plausible alternative to algorithmic backtracking.

In considering the use of constructions as the basis for language representation and processing, it becomes clear that fully integrated representations may not in principle be possible. Instead, representations are likely to be integrated just to the extent supported by the constructions activated by the input and selected for integration, with different constructions often representing different tiers or dimensions of meaning that are not fully integratable.

Construction Grammar

Construction Grammar (Fillmore, 1988; Fillmore and Kay, 1993; Goldberg, 1995) is an emerging linguistic theory based on the notion of *constructions*. "Constructions are stored pairings of form and function, including morphemes, words, idioms, partially lexically filled and fully general linguistic patterns…any linguistic pattern is recognized as a construction as long as some aspect of its form and function is not strictly predictable from its component parts" and even fully predictable constructions may be stored "as long as they occur with sufficient frequency" (Goldberg, 2003:219). A classic example of a construction is the *transitive verb clause* consisting of a *subject*, *transitive verb* and *object* as exemplified by "the man$_{\text{subject}}$ hit$_{\text{trans-verb}}$ the ball$_{\text{object}}$". A less common construction is the *caused-motion* construction as exemplified by "she$_{\text{subject}}$ sneezed$_{\text{intrans-verb}}$ the napkin$_{\text{object}}$ off the table$_{\text{direction}}$" (Goldberg, 1995). The caused-motion construction is interesting in that a verb which is normally *intransitive* as exemplified by "she sneezed" occurs with an object "the napkin" and directional prepositional phrase "off the table". Many normally intransitive verbs can occur in this construction. (An alternative viewpoint is that the caused-motion construction is integrated with a distinct intransitive verb construction in this example.) Although Construction Grammar began with the exploration of many unusual constructions (e.g. the "let alone" construction in Fillmore, Kay and O'Connor, 1988), it has come to be recognized that the basic principles of Construction Grammar apply to common constructions as well. In fact, a basic claim of Construction Grammar is that "the network of constructions captures our knowledge of language *in toto* – in other words, it's constructions all the way down" (Goldberg, 2003).

The key insights of Construction Grammar are beginning to have a significant impact on other linguistic formalisms including Cognitive Grammar (Langacker, 1987, 1991; Talmy 2000; Lakoff, 1987), HPSG (Sag and Wasow, 1999; Sag, 1997) and even Generative Grammar as reformulated by Culicover and Jackendoff (2005). However, to date, Construction Grammar has had little impact on research in language processing (exceptions include Bergen & Chang, 2005; Steels & De Beule, 2006).

Phrase and Clause Level Constructions

Constructions are learned chunks of linguistic knowledge that tie subordinate linguistic elements together. The elements of phrase and clause level constructions may be specific lexical items (e.g. "is", "was"), lexemes (e.g. "be") or linguistic categories. Fully lexicalized constructions containing multiple words are called *multiword expressions*. The more general a construction, the more likely it is to contain categories as elements rather than specific lexical items. Categories may be *form-based* or *functional*, although the focus of this paper is on functional categories. For example, the `[subject predicator object]`$_{\texttt{clause}}$ construction (where `predicator` roughly corresponds to *verb group*—i.e. verb + tense, aspect, modality, and polarity) describes a sequence of three functional categories, whereas the `[subject` *`hit`*$_{\texttt{predicator}}$ `object]`$_{\texttt{clause}}$ construction is specific to the verb "hit". For the most part, constructions are sequence specific, although the possibility of constructions whose elements are not sequence specific is not precluded.

The following notation is used for the representation of constructions:

$[A_{sub}\ \textit{B}\ C_{sup}]_D$

In this representation, square brackets enclose the construction, which consists of an ordered list of elements A, B and C. The elements in a construction may be specific lexical items, lexemes (i.e. abstracted dictionary forms) or functional categories (i.e. functionally typed variables). A subscript, $_{sub}$ or $_{sup}$, on an element may be used to indicate a functional subcategory or super type (and conceivably a form-based category). The functional category of the construction is indicated by the subscripted $_D$ to the right of the construction. Lexical items are italicized to distinguish them from lexemes.

Over the course of a lifetime, humans acquire a large knowledge base of constructions at multiple levels of abstraction and generalization. For language comprehension, the most lexically specific constructions matching the input are likely to activated, selected and integrated, and language comprehension can be viewed as lexically driven within the context of constructions. For example, the [subject (*kicked the bucket*)$_{predicate}$]$_{clause}$ construction (where predicate roughly corresponds to *tensed VP*) will be preferred over the [subject kick$_{predicator}$ object]$_{clause}$ construction where both are activated by the input, since the former is more lexically specific. In addition, constructions which match the largest chunks of input are likely to be preferred (cf. Grossberg and Myers, 1999). Thus, [subject *ate*$_{predicator}$ object]$_{clause}$ will be preferred over [subject *ate*$_{predicate}$]$_{clause}$ given the input "she ate the sandwich".

It should be noted that constructions may contain actual and ambiguous lexical items. For example, the construction [(*take a hike*)$_{predicate}$]$_{imperative\text{-}clause}$ contains the ambiguous lexical items "take", "a" and "hike", although the construction as a whole unambiguously means "go away" in its idiomatic interpretation.

Construction-Driven Language Processing

A processing mechanism based on the *activation*, *selection* and *integration* of constructions is proposed. Constructions are activated in memory by a parallel, automatic *spreading activation process* to the extent that they match the current input and prior context. The most highly activated constructions are selected for integration by a (largely) serial *control process*. Selected constructions with categorical elements and as yet unrealized lexical items establish expectations which drive the processing mechanism. Category expectations in constructions can function to establish the category of the prior input or to set the context for processing the subsequent input and also determine how inputs are integrated. For example, the [subject *hit*$_{predicator}$ object]$_{clause}$ construction, activated by the word "hit", establishes the expectations that the prior input is functioning as a subject and the subsequent input is functioning as an object. A prior input capable of functioning as a subject and a subsequent input capable of functioning as an object can be integrated into this construction. Of course, expectations may be violated and when they are, the violations must be accommodated. Possible mechanisms of *accommodation* include the selection and integration of a different construction (in the context of the expectation violation and not via algorithmic backtracking), modification of the selected construction (Ball, 2004), or construal of the *to be integrated* element as being of the required functional type (Langacker, 2000)—as in construal of the infinitive phrase "to be integrated" as a nominal head modifier in this sentence. For example, in the context of the construction [*the* head]$_{nominal}$, activated by the processing of the word "the" within the expression "the hit", the word "hit" can subsequently be integrated as the head. The [subject *hit*$_{predicator}$ object]$_{clause}$ construction which is also activated by "hit" may or may not be selected for integration during processing. Note that instantiating "hit", a type of action, as the head of a nominal construction involves construing the action that the nominal refers to as though it were an object. This is a common form of construal in English—especially for words describing actions which occur instantaneously and are easily objectified.

A construction-driven language processing system is likely to lead to messier representations than those typically posited in other computational linguistic or cognitive science approaches. Although constructions can be integrated to some extent, there is no guarantee that this integration will lead to anything like a well-formed tree, let alone a binary branching tree (Kayne, 1994). In fact, to the extent that constructions are independent of each other, they can only be integrated via the lexical items and categories they share. Further, it is likely that constructions will often conflict with each other, leading to representations that are in part inconsistent (in the sense that they assign different, often competing, representations to the same input). Issues in determining the basic structure of clauses—is it SVO or Subject-Predicate—are a reflection of this inconsistency. The subject has a saliency in the Subject-Predicate construction that it does not have in the more symmetric SVO construction. Both constructions are likely to be available in the inventory of constructions available to fluent comprehenders of English. Which one gets activated and selected (or perhaps both) is likely to vary from utterance to utterance depending on the prior context and variability in the manner and form of expression of the current utterance. For example, in

John hit (pause) and Sue kicked (pause) the door

the Subject-Predicate construction is unlikely to be selected given the grammatical separation of the subject and verb from the object which would normally form part of the predicate (combining with the verb). Similarly, in

He's hitting the ball

the cliticization of "is" with "he" argues against selection of a Subject-Predicate construction (assuming the auxiliary verb is normally part of the *predicate*). In fact, there is very likely to be a specialized [*he's* predication]$_{clause}$ construction (where predication roughly corresponds to *untensed VP*) that gets activated and selected. Finally, question forms argue against the necessary activation and selection of a Subject-Predicate construction. Consider

Where is he going?

which suggests a specialized construction like [*where* be subject predication]$_{wh\text{-}clause}$.

In general, there are a number of different constructions which come in to play in the processing of clausal heads. These constructions overlap in various respects, but all of them can be motivated by different linguistic expressions—especially expressions involving conjunction:

(he's) **kicking the ball** and **throwing the rock** → $[\mathtt{V}_{head}\ \mathtt{obj}_{comp}]_{predication}$
(he) **is kicking** and **was hitting** (the ball) → $[\mathtt{be}_{spec}\ \mathtt{V\text{-}ing}_{head}]_{predicator}$
(he) **kicked the ball** and **threw the rock** → $[\mathtt{V\text{-}ed}_{spec/head}\ \mathtt{obj}_{comp}]_{predicate}$
(why did) **he kick the ball** and **she throw the rock** → $[\mathtt{subj}_{comp}\ \mathtt{kick}_{head}\ \mathtt{obj}_{comp}]_{proposition}$

The term *predication* is used to describe a construction consisting of an untensed clausal head along with its non-subject complements. The term *predicator* is used to describe a construction consisting of a clausal head along with its tense specification, but without the non-subject complements. Note that the head of a predication or predicator need not be a verb, nor is an object required in a predication. In "he is *running*", the verb "running" is the head, in "he is *sad*", the adjective "sad" is the head, and in "he is *there*", the adverb "there" is the head—and there is no object in these examples (Ball, 2005). The functional categories *predicator* and *predication* generalize over these alternative phrasal forms. The term *predicate* is used to describe a construction consisting of a clausal head along with its tense specification and non-subject complements. The term *proposition* is used to describe a construction consisting of an untensed clausal head along with its complements (including the subject).

A Processing Example

During the processing of the sentence

He is kicking the ball

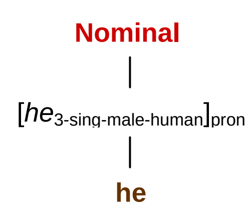
the following constructions are likely to be activated:

he → $[\mathit{he}_{3\text{-}sing\text{-}male\text{-}human\text{-}pron}]_{nominal}$
is → $[\mathtt{be}_{3\text{-}pres\text{-}sing}]_{verb}$
he is → $[\mathtt{ref\text{-}pt}_{comp}\ \mathtt{be}_{spec}\ \mathtt{predn}_{head}]_{clause}$
kicking → $[\mathtt{kick}_{v\text{-}ing}]_{verb}$
kicking → $[\mathtt{subj}_{comp}\ \mathtt{kick}_{head}\ \mathtt{obj}_{comp}]_{proposition}$
kicking → $[\mathtt{V}_{head}\ \mathtt{obj}_{comp}]_{predication}$
is kicking → $[\mathtt{be}_{spec}\ \mathtt{V\text{-}ing}_{head}]_{predicator}$
is kicking → $[\mathtt{be}_{spec}\ \mathtt{V\text{-}ing}_{head}\ \mathtt{obj}_{comp}]_{predicate}$
the → $[\mathit{the}_{spec}\ \mathtt{head}]_{nominal}$
the ball → $[\mathit{the}_{spec}\ \mathit{ball}_{head}]_{nominal}$

The $[\mathit{he}_{3\text{-}sing\text{-}male\text{-}human\text{-}pron}]_{nominal}$ construction encodes the knowledge that pronouns like "he" (3rd person, singular, male, human) can function as full nominals, encoding both a *referential specifier* function and an *objective head* function (Ball, 2005). The $[\mathtt{be}_{3\text{-}pres\text{-}sing}]_{verb}$ construction encodes the status of "is" as the 3rd person, present tense, singular form of the verb "be". The $[\mathtt{ref\text{-}pt}_{comp}\ \mathtt{be}_{spec}\ \mathtt{predn}_{head}]_{clause}$ construction captures the use of a *reference point* complement (Taylor, 2000) and a referential specifier ($\mathtt{be}_{spec}$) to tie a predication functioning as head of a clause to the larger discourse situation via the reference point and referential specifier. The $[\mathtt{kick}_{v\text{-}ing}]_{verb}$ construction captures the "V-ing" verb form of "kicking". The $[\mathtt{subj}_{comp}\ \mathtt{kick}_{head}\ \mathtt{obj}_{comp}]_{proposition}$ construction captures the basic relational meaning of the verb "kick" which combines with a subject and object complement to form a *proposition*. This construction is closely related to the basic SVO form of a clause. The $[\mathtt{V}_{head}\ \mathtt{obj}_{comp}]_{predication}$ construction captures the combining of a tenseless verb head with an object complement to form a *predication* that functions as the head of the $[\mathtt{ref\text{-}pt}_{comp}\ \mathtt{be}_{spec}\ \mathtt{predn}_{head}]_{clause}$ construction. The $[\mathtt{be}_{spec}\ \mathtt{V\text{-}ing}_{head}]_{predicator}$ construction captures the combining of the auxiliary verb "be" functioning as a specifier with the progressive form of a verb functioning as the head in forming a *predicator*. The $[\mathtt{be}_{spec}\ \mathtt{V\text{-}ing}_{head}\ \mathtt{obj}_{comp}]_{predicate}$ construction captures the combining of the auxiliary verb "be" functioning as a specifier with the progressive form of a verb functioning as the head and an object complement in forming a *predicate*. The $[\mathtt{the}_{spec}\ \mathtt{head}]_{nominal}$ construction captures the encoding of a referential specifier and objective head to form a nominal. The $[\mathtt{the}_{spec}\ \mathtt{ball}_{head}]_{nominal}$ construction captures the encoding of "ball" as the head of the $[\mathtt{the}_{spec}\ \mathtt{head}]_{nominal}$ construction.

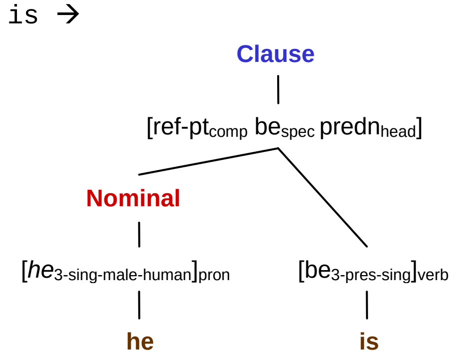
The actual processing of this utterance is likely to proceed as follows:

he →



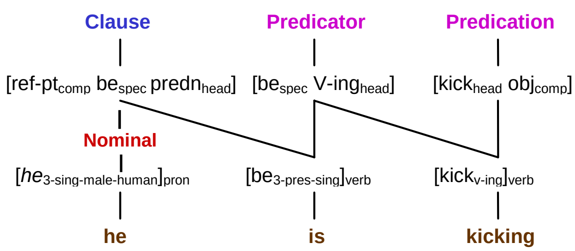
The word "he" activates a *nominal* construction which is capable of referring to some object independently of any larger linguistic unit in which it may participate.

he is →



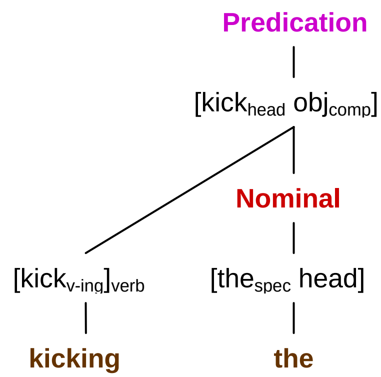
The word "is" following "he" activates a *clause* construction. The assumption here is that the nominal "he" and auxiliary verb "is" are immediately integrated into the $[\texttt{ref-pt}_{comp}\ \texttt{be}_{spec}\ \texttt{predn}_{head}]_{clause}$ construction. In general, delaying integration of linguistic elements into constructions is likely to lead to processing difficulties since the need to retain separate linguistic units in memory will run up against limits on the number of unintegrated linguistic elements which can be separately retained in *working memory*.

`he is kicking →`



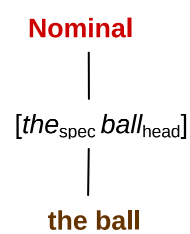
Two additional constructions—*predicator* and *predication*—are activated by "kicking" and immediately integrated to the extent possible. It is assumed that the *predicate* and *proposition* constructions are not selected for integration in this example, even though they are activated. Time constraints and selection competition are likely to preclude integration of all activated constructions and the *predicate* and *proposition* constructions are not likely to be as strongly activated as the *predicator* and *predication* constructions in this example. Note the implication that neither the basic SVO nor the basic Subject-Predicate clause construction is integrated into this representation!

`the →`

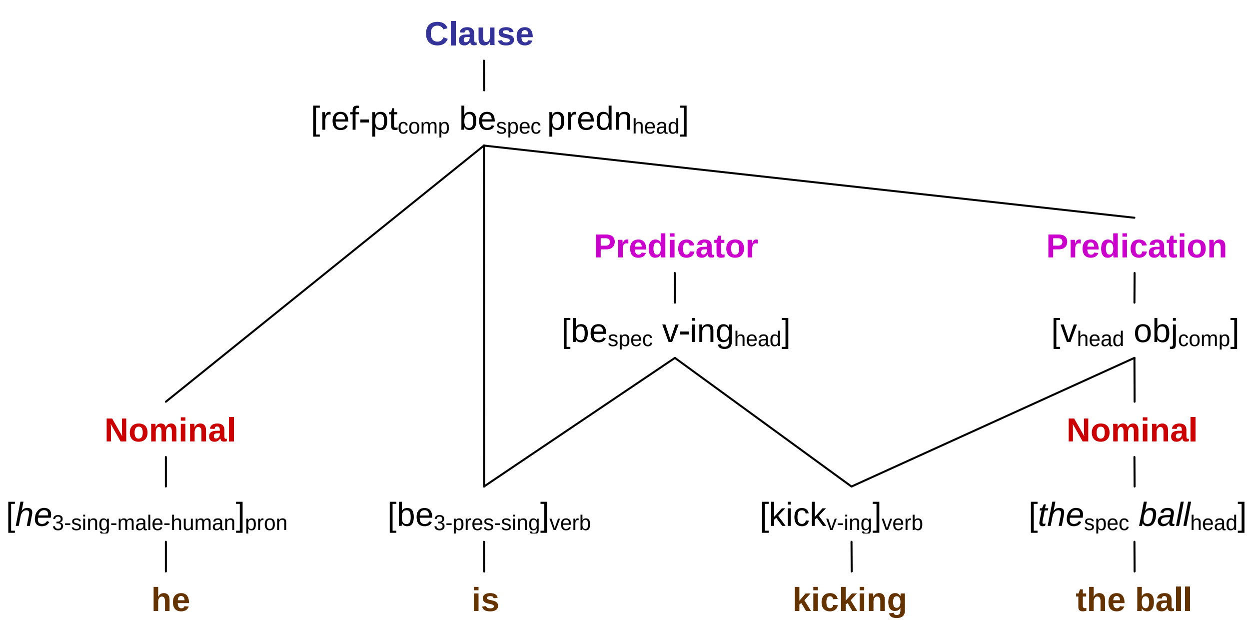


The word "the" activates a *nominal* construction and integrates "the" as the specifier. This nominal construction is integrated as the object of the *predication* construction even before the head of the nominal construction is processed and integrated into the nominal!

`the ball →`



The noun "ball" is integrated as the head of the nominal "the ball". After processing, the linguistic representation for the utterance "he is kicking the ball" is shown below:



In this example, it is assumed that a single clause level construction [ref-pt$_{comp}$ be$_{spec}$ predn$_{head}$]$_{clause}$ was selected for integration. However, two "predicate" level constructions [be$_{spec}$ V-ing$_{head}$]$_{predicator}$ and [V$_{head}$ obj$_{comp}$]$_{predication}$ were selected for integration. One way of viewing such representations is as having multiple tiers corresponding to different grammatical dimensions of meaning encoded via constructions. These different tiers of meaning get integrated just to the extent that selected constructions have overlapping lexical items and functional categories. Such an approach opens up the possibility of having additional tiers to capture meaning distinctions conveyed by topic-focus and given-new contrasts, among others. A tiered approach to representing different grammatical dimensions of meaning is in alignment with current tiered theories of phonology (cf. Kaye, 1989) and (to some extent) with lexical semantic approaches which assume a multidimensional space for representing the meaning of words (Landauer & Dumais, 1997).

Activation, Selection and Integration

Activation is a *parallel* process that biases or constrains the selection and integration of corresponding declarative memory (DM) elements into a linguistic representation. The activation mechanism is based on the spreading activation mechanism of the ACT-R cognitive architecture (Anderson et al. 2004). A computational cognitive model intended to validate the processing mechanism is being implemented in this architecture (Ball, Heiberg & Silber, in preparation). Based on the input and prior context, a collection of DM elements is activated in parallel.

The selection mechanism is based on the *serial* retrieval mechanism of ACT-R—an alternative to the parallel *competitive inhibition* mechanism typical of connectionist models (cf. Vosse & Kempen, 2000). Retrieval occurs as a result of selection and execution of a production—only one production can be executed at a time—whose right-hand side provides a retrieval template that specifies which type of DM chunk is eligible to be retrieved. The single, most highly activated DM chunk matching the retrieval template—subject to random noise—is retrieved. The retrieval template varies in its level of specificity in accord with the production selected for execution. For example, when a production that retrieves a DM chunk of type *word* executes, the retrieval template may specify the form of the input (e.g. "airspeed") in addition to the DM type *word*. When a production that retrieves a DM chunk of type *part of speech* (POS) executes, the retrieval template may specify the word without specifying the POS—allowing the biasing mechanism to constrain POS determination.

The retrieved DM chunk is matched on the left-hand side of another production which, if selected and executed, determines how to integrate the retrieved DM chunk into the representation of the preceding input. Production selection is driven by the matching of the left-hand side of the production against a collection of buffers (e.g. goal, retrieval, context, short-term working memory) which reflect the current goal, current input and previous context. The production with the highest utility—learned on the basis of prior experience—which matches the input and prior context, is selected for execution—subject to random noise. A default production which simply adds the retrieved DM chunk to a short-term working memory (ST-WM) stack executes if no other production matches. The ST-WM stack—which is limited to four linguistic elements—constitutes part of the context for production selection and execution.

Context Accommodation

A key element of the integration process is a mechanism of *context accommodation* which provides for *serial processing without backtracking*. According to Crocker (1999), there are three basic mechanisms of language processing: 1) serial processing with backtracking, 2) parallel processing, and 3) deterministic processing. Context accommodation is an alternative non-backtracking, serial processing mechanism. The basic idea behind this mechanism is that when the current input is inconsistent with the preceding context, the context is modified to accommodate the current input without backtracking. This mechanism is demonstrated using the example "no airspeed or altitude restrictions". The processing of the word "no" leads to retrieval of a nominal construction containing the following functional elements: specifier, modifier, head, post-head modifier (an extension of the earlier example):

[specifier modifier head post-mod]$_{nominal}$

"No" is integrated as the specifier in this nominal construction and expectations are established for the occurrence of the remaining functional elements.

[*no*$_{spec}$ mod head post-mod]$_{nominal}$

This nominal construction is made available in the ST-WM stack to support subsequent processing. The processing of the noun "airspeed" leads to its integration as the head of the nominal construction, since nouns typically function as heads of nominals.

[*no*$_{spec}$ mod *airspeed*$_{head}$ post-mod]$_{nominal}$

The processing of the conjunction (or disjunction) "or" leads to its addition to the ST-WM stack since the category of the first conjunct of a conjunction cannot be effectively determined until the linguistic element after the conjunction is processed—due to rampant ambiguity associated with conjunctions. Note that delaying determination of the category of the first conjunct until after processing of the linguistic element following the conjunction provides a form of deterministic processing reminiscent of Marcus's deterministic parser (1980). The processing of the noun "altitude" in the context of the conjunction "or" and the nominal "no airspeed" with head noun "airspeed" results in the accommodation of "altitude" such that the head of the nominal construction is modified to reflect the disjunction of the nouns "airspeed" and "altitude".

[*no*$_{spec}$ mod (*airspeed or altitude*)$_{head}$ post-mod]$_{nominal}$

The processing of “restrictions” in the context of the nominal “no airspeed or altitude” results in the accommodation of “restrictions” such that the current head “airspeed or altitude” becomes the pre-head modifier and “restrictions” becomes the head. The final representation has the form:

[*no*$_{spec}$ (*airspeed or altitude*)$_{mod}$ *restrictions*$_{head}$ post-mod]$_{nominal}$

This representation was arrived at using a serial processing mechanism without backtracking, despite the rampant local ambiguity of the utterance!

Context accommodation is a powerful serial processing mechanism which overcomes the limitations and cognitive implausibility of serial processing mechanisms which rely on backtracking, without sacrificing the advantages of serial processing over parallel and deterministic processing. It is unrealistic to expect a parallel processing mechanism to carry forward more than a few possible representations at once, which means a mechanism like context accommodation is needed in any case, and deterministic mechanisms require delaying integration of linguistic elements for indeterminate periods—requiring their separate representation—which is likely to exceed the limited capacity of ST-WM if used extensively.

Summary

This paper presents an approach to language comprehension based on the *activation*, *selection* and *integration* of *constructions* corresponding to the linguistic input. Multiple, often conflicting, constructions are likely to be activated by each lexical item in the input. The resulting linguistic representations depend crucially on which activated constructions are selected for integration, and the degree to which selected constructions are integratable. Whereas activation is based on a parallel spreading activation mechanism, selection and integration rely on a serial processing mechanism combined with a mechanism of *context accommodation*—a cognitively plausible alternative to algorithmic backtracking.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S, Lebiere, C, and Qin, Y. (2004). An Integrated Theory of the Mind. *Psychological Review* 111, (4). 1036-1060

Ball, J. (2004). A Cognitively Plausible Model of Language Comprehension. In *Proceedings of the 13th Conference on Behavior Representation in Modeling and Simulation*, 305-316. ISBN: 1-930638-35-3

Ball, J. (2005). A Bi-Polar Theory of Nominal and Clause Structure and Function. In *Proceedings of the 2005 Cognitive Science Conference*. Sheridan Printing.

Ball, J., Heiberg, A. & Silber, R. (in preparation). Toward a large-scale, functional model of language comprehension in ACT-R 6.

Bergen, B. & Chang, N. (2005). Embodied construction grammar. In J. Östman and M. Fried, eds., *Construction Grammars: Cognitive Grounding and Theoretical Extensions*. Amsterdam: John Benjamins

Crocker, M. (1999). Mechanisms for Sentence Processing. In Garrod, S. & Pickering, M. (eds), *Language Processing*, London: Psychology Press.

Culicover, P. & Jackendoff, R. (2005). *Simpler Syntax*. Oxford: Oxford University Press.

Fillmore, C. (1988). The Mechanisms of Construction Grammar. *BLS* 14: 35-55.

Fillmore, C. and Kay, P. (1993). *Construction Grammar Coursebook*. Berkeley, CA: Copy Central.

Fillmore, C., Kay, P. and O'Connor, M. (1988). Regularity and Idiomaticity in Grammatical Constructions: The Case of let alone, *Language*, Vol. 64, Number 3, 501-538

Goldberg, A. (1995). *A Construction Grammar Approach to Argument Structure*. Chicago: The University of Chicago Press.

Goldberg, A. (2003). Constructions: a new theoretical approach to language. *TRENDS in Cognitive* Sciences. Vol. 7, No. 5, pp. 219-224.

Grossberg, S. & Myers C. (1999). The Resonant Dynamics of Speech Perception: Interword Integration and Duration-Dependent Backward Effects. Technical Report CAS/CNS-TR-99-001. Boston, MA: Boston University.

Kaye, J. (1989) *Phonology, a cognitive view*. Hillsdale, NJ: LEA.

Kayne, R. (1994). *The Antisymmetry of Syntax*. Cambridge, MA: The MIT Press.

Lakoff, G. (1987). *Women, Fire and Dangerous Things.* Chicago: The University of Chicago Press.

Landauer, T. & Dumais, S. (1997). A solution to Plato's problem: the Latent Semantic Analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, *104(2)*, 211-240.

Langacker, R. (1987, 1991). *Foundations of Cognitive Grammar*, Volumes 1 & 2. Stanford, CA: Stanford University Press.

Langacker, R. (2000). Why a mind is necessary: Conceptualization, grammar and linguistic semantics. In Albertazzi**,** L. (ed.), *Meaning and Cognition*. Amsterdam: John Benjamins, 25–38.

Marcus, M. (1980). *A Theory of Syntactic Recognition for Natural Language*. Cambridge, MA: The MIT Press.

Sag, I. (1997). English Relative Clause Constructions. *Journal of Linguistics* . 33.2: 431-484.

Sag, I. & Wasow, T. (1999). *Syntactic Theory, A Formal Introduction*. Stanford: CSLI Publications.

Steels, L. & De Beule, J. (2006). A (very) Brief Introduction to Fluid Construction Grammar. In *Proceedings of the Third International Workshop on Scalable Natural Language Understanding*.

Talmy, L. (2000). Toward a Cognitive Semantics, Vols I and II. Cambridge, MA: The MIT Press

Taylor, J. (2000). Possessives in English, An Exploration in Cognitive Grammar. Oxford: Oxford University Press.

Vosse, T. & Kempen, G. (2000). Syntactic structure assembly in human parsing. *Cognition*, 75, 105-143.