Grammaticality Checking
Devices in the SA-Model*

Sosei Aniya

1 Preliminary Remarks

This paper addresses a problem with a grammaticality checking device in the SA-Model (Aniya, 2001), a working model which is an extension of ideas developed and formalized in Lexically Based Algebra (Brame and Kim, 1998). The problem is spelled out as: Specifically, how does the model determine whether or not a language fragment violates a grammatical condition? Since the SA-Model aims at providing an integrated system of grammar, it should internalize a self-governed device for checking well-formedness or grammaticality. To this end, I offer an algebra-oriented solution in Section 3. In the course of pursuing the solution, possible problems come into view, such as: What happens to a language fragment after its production?, How are grammatical conditions formally defined?, and Is there a relation between/among grammatical conditions? Those problems are essential. Therefore, they will be discussed and given lexically-based algebraic explanations in Section 4.

2 Problems

2.1 Brame and Kim's (1998) LBA Model

In Brame and Kim (1998: 124) there is a set theoretically-defined general device for producing language fragments, whose definition is shown here under (1):

(1) *Definition.* Let **LEX**=(*LEX!*, f, **1**, *T*) be a lexically based production algebra.

    We say that **LEX** generates or produces the language **L** provided the following equation is satisfied.

$$L=\{x\mid [x, j]\ Œ\ LEX!\ \&\ j\ Œ\ T\}$$

    The production mechanism of the binary operation f is seen in the following formulae (Brame and Kim, 1998: 120).

(2) *Lexical Composition* (*LC*)

    f: *LEX!*× *LEX!   LEX!*

x, j

Brame and Kim's (1998) LBA model has accomplished a unique achievement in the field of theoretical linguistics.   The LBA model can accommodate both the production and recognition of language fragments real-time within an algebraically constructed system. Their pioneering work deserves credit.   However, the LBA model has not been equipped to check for grammaticality of language fragments. In order to remedy this shortcoming, Aniya (2001) devises an extended model of LBA, to which we will now turn.

2.2 Aniya's (2001) SA-Model

Aniya (2001: 11) proposes a quadruplet system of grammar named the SA-Model, which is reproduced here under (6).   In this model, the initial component is the same as the **LEX** in Definition (1), while the rest is an innovation.

(6) *Definition*

SA-Model=(**LEX**, *GC*, *WFC*, *LDS*)

The second component is a set called *GC* (*Grammatical Conditions*), whose elements are shown in (7).

(7) *Grammatical Conditions* (*GC*)

GC={PhonC, MorpC, SynC, SemC, PragC}

PhonC={$phoc_i$, ..., $phoc_n$}

MorpC={$morpc_i$, ..., $morpc_n$}

SynC={$sync_i$, ..., $sync_n$}

SemC={$semc_i$, ..., $semc_n$}

PragC={$pragc_i$, ..., $pragc_n$}

As shown above, each element in *GC* is in turn a set of grammatical conditions.

The third component is a grammaticality checking device termed the *Well-Formedness Criterion*.   Given below is its definition.

(7) *Well-Formedness Criterion* (*WFC*)

Definition. Let $p \bullet x = y$ be a well-formedness algebra with the following terms:

i. Let *p* be a lexical composition product, and assign *p* value 1;

    ii. If *p* violates *GC*, then assign *x* value 0, otherwise value 1;

    iii. If *y* is 1, then *p* is well-formed; if *y* is 0, then ill-formed.

By applying *WFC* to the word *spring* [sprɪŋ] in (8a), for example, we see that the word in question is declared well-formed, whereas the word *sbring* [sbrɪŋ] in (8b) is deemed ill-formed. The latter word violates condition phoc[17] (see Appendix 1), which prohibits voiceless-voiced consonant clusters at the onset in English.

(8) Example

    a. [sprɪŋ]•1=1•1=1   (well-formed)

       p • *x* = *y*

    b. [sbrlŋ]•0=1•0=0   (ill-formed)

Let us now consider possible problems related to the *Well-Formedness Criterion* (*WFC*). Term (ii) raises two problems: (i) How do we know whether or not *p* violates a grammatical condition in *GC*?, and (ii) The mapping of *p* into *GC* should be a one-to-many mapping rather than a one-to-one mapping. This point is not guaranteed within the system. Having located the problems, let us now consider a solution for each of them.

3 Solutions

3.1 Grammaticality verification mechanism

Let us now consider a solution for each of the two problems discussed in the preceding section in a step by step fashion. First, I incorporate a grammaticality checking device into the SA-Model. Consider the following definition.

(9) *Definition*. Let *GRAMMATICALITY* (*GRAM*) = (*P*, *G*, *F*, *W*) be a grammaticality checking device, where;

    i. *P* is a set of language fragments;

    ii. *G* is a set of grammatical conditions;

    iii. *F* is a function *F* (*p*, *g*), which reads "*F* maps *p* into *g*, where *p* is a language fragment and *g* is a grammatical condition".

iv. *W* is a well-formedness criterion satisfying the following terms:

    a. Let $p=1 \cdot g$ be a well-formedness algebra;

    b. If p violates g, then assign g value 0, else value 1;

    c. If p's value is 1, then p is well-formed; if 0, then ill-formed.

Term (9iv.b) calls for the accurate definition of grammatical conditions. Therefore, I incorporate the following proposition into the SA-Model. In formalizing the proposition, I employ the Boolean logic of implication as it appears in Hewitt (2000: 58).

(10) *Proposition*. Grammatical conditions of *GC* satisfy the following Boolean logic[1]:

    i. Implication: A>B[2]

    ii. Boolean: If A is true, then the truth value for B determines the truth value of A>B:

        If B is true, then A>B is true;

        If B is false, then A>B is false;

   If A is false, then A>B is false, and the truth value of B is irrelevant.

Proposition (10) is justified for two reasons. First, *Proposition* (10) formally defines the grammatical conditions of (9ii) in terms of an 'if-then' implication. Second, grammatical conditions are defined in accordance with the above Boolean terms (see Appendix 1): If the truth value of proposition A (i.e., the 'if-antecedent clause' of a grammatical condition) is false, then the truth value of proposition B (i.e., the 'then-consequent clause' of a grammatical condition) becomes irrelevant. Without this implicational formalization, there is no way of knowing whether grammatical conditions themselves are correct or wrong. Notice also that *Definition* (9) incorporates the *Well-Formedness Criterion* (*WFC*) in a modified fashion as in (9iv). Therefore, the old *WFC* in Aniya (2001) is now dispensed with.

Now let us examine closely *Definition* (9iii) to see whether *F* is a one-to-one relation or a one-to-many relation. The checking of *p* against *g* is the task of *F* as defined in (9iii). The mapping function of

F should be a one-to-many relation.   This must be so because *p* can correspond to one or more grammatical conditions in *GC* as shown in the diagram under (11).

(11) *Proposition*. *F* in *GRAM* is a one-to-many relation.

  Example:

$$p \nearrow \text{phoc}_i \quad \searrow \text{morpc}_j \quad \rightarrow \text{sync}_k \quad \nwarrow \text{semc}_l \quad \text{pragc}_m$$
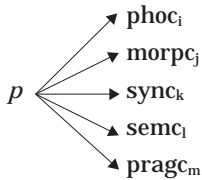
  As for additional support of one-to-many grammaticality mapping, consider the existential *there*-construction or the *tough*-construction in English.   Such constructions, like many others, are subject to syntactic, semantic, and pragmatic conditions/constraints (Aniya, 1992, 1998).

  Let us now show how the grammaticality verification mechanism works by looking at two grammatical transactions: One involving production, recognition, resolution, and the other involving the grammaticality checking of a language fragment.   Consider as an initial step the following chart with respect to the production, recognition, and resolution of *John is easy to please*.

(12) Example

PRODUCTION	RECOGNITION	RESOLUTION
[I , 1] [*John*, S^TD V]	[*John is easy to please*, S^T] [*John*[1], VD S^T]	[*John is easy to please*, S^T] [I , $S^T S^T$]
[*John*, S^TD V] [*is*, VD A^d]	[*is easy to please*, VD] [*is*[1], A^d DV]	[*John is easy to please*, S^T] [I , VV]
[*John is*, $S^T A^d$] [*easy*, A^ϕT]	[*easy to please*, A^d] [*easy*[1], T A^d]	[*John is easy to please*, S^T] [I , $A^d A^d$]
[*John is easy*, S^TT] [*to*, T V]	[*to please*, T] [*to*[1], VT]	[*John is easy to please*, S^T] [I , T T]
[*John is easy to*, S^TV] [*please*, V]	[*please*, V] [*please*[1], V]	[*John is easy to please*, S^T] [I , VV]
[*John is easy to please*, S^T]	[I , 1]	[*John is easy to please*, S^T]

  Let us take, as an example, the final product [*John is easy to*

please,   S^T] at the bottom of PRODUCTION, and look into a grammaticality checking procedure in terms of *GRAM* (*MATICALITY*). By term (iii) of *GRAM*, [*John is easy to please*,   S^T] is mapped into *GC* for grammaticality checking.   This mapping is a one-to-many relation as defined in *Proposition* (11).   Specifically, the language fragment is mapped into the five elements of *GC*: *PhonC*, *MorpC*, *SynC*, *SemC*, and *PragC*, each of which is a set of grammatical conditions. Bearing in mind the procedure followed so far, let us agree that the language fragment [*John is easy to please*,   S^T] does not violate any of the relevant grammatical conditions listed in Appendix 1. Granting this, we obtain the following model computation, where *g* represents a relevant grammatical condition.

(13)

> a. [*John is easy to please*,   S^T] is assigned a *p*.   (by 9iii);
>
> b. $p=1 \cdot g$ (by (9iv.a));
>
> c. $p=1 \cdot 1$ (by (9iv.b));
>
> d. $p=1$ ([*John is easy to please*,   S^T] is declared well-formed by
> (9iv.c)).

We have seen the grammaticality checking procedure of *GRAM* in action.   Let us now consider in the next section 'stock-memory access' and 'speech-time axis' mechanisms.   The two mechanisms are essential for improving a working model of the SA-Model.   The task of incorporating the second mechanism into the model, however, has not been developed.   This issue, therefore, is taken up for a future study.

4 Stock-memory Access and Speech-time Axis Mechanisms

4.1 Stock-memory access

The first formula of *Lexical Composition* (*LC*) given under (2) entails that a product of the binary operation goes back into a language *L*.   This is significant.   By applying the binary operation cyclically, a stock of language fragments is produced.   It can be assumed that the stock of language fragments parallels that of the stock memory of language fragments, to which the speaker can make access and

compose a new set of sentences.   Let us agree to call the above assumption the 'stock-memory access' assumption.   This assumption predicts that the speaker does not always need to resort to a method of putting together a sentence in a piece-meal fashion.   Therefore, the assumption runs counter to the point of view of generative grammar advocates, who seem to adhere to the idea that, in essence, sentences are created bit-by-bit, combining component pieces.   For those advocates, the lexicon is taken to be a set of words together with syntactic, semantic, and phonological information.

Although rudimentary, psycholinguistic and neurolinguistic experiments seem to give indirect support for the stock-memory access assumption.   Let us consider this point in a phased explanation. First, there seems to be a close connection between storage and the processing of memories. Coleman (1998: 304) concludes, after reviewing a large number of neurological experiments, that "there is a phonological lexicon located in the superior temporal gyri of both hemispheres, close to the auditory processing areas."   This does not directly constitute evidence for the stock-memory access assumption, but it does tell us that phonological words are stored in a specific place in the cerebral cortex.   Silveri and Misciagna's (2000: 137) neurolinguistic study claims that there is a phonological short-term memory component "that allows the retention of verbal information for a short period of time, thus permitting specific verbal processing such as word repetition, sentence comprehension and new language acquisition."   Again, this is not direct evidence for the stock-memory access assumption.   It does, however, support the assumption that language fragments are stored.   And if there is a phonological short-term memory, we might as well expect that there is a phonological long-term memory.   As a result of an experiment done on a Finnish aphasic patient, Nenonen et al. (2002: 55) conclude that "even a severely dyslectic and agrammatic patient is able to read noun phrases when they are idioms.   It seems that the noun phrase idioms are more like holistic 'long words' than the verb phrase idioms, i.e., the former

are presumably retrieved from the lexicon as such." From this outcome, we can assume that formulaic words, phrases, and sentences of high frequency in use are more likely to be stored as stock memory items.

Based on a psycholinguistic study, Dabrowska (2000: 84) reports that "child usage is highly formulaic and it progresses from rote-learned formulas to adult-like productivity." Dabrowska's (2000: 83) study offers strong evidence in support of findings against "the claim that children learn abstract transformational rules like subject-auxiliary inversion and *wh*-movement. It also confirms the view that children begin with a fixed repertoire of lexically-based patterns or formulas, both invariant formulas such as *Whassis*? and *Whatchadoing*? and formulaic frames like *Where's* ____? and *What's* ____*doing*?" Therefore, we can assume that children, like adults, store language fragments/chunks in the brain and retrieve them for use in normal and exigent communicative situations. A piece of support for this assumption comes from children's language acquisition research. Conducting cognitive linguistic analyses, Tomasello (2000: 77) concludes that "when young children have something they want to say, they sometimes have a set expression readily available and so they simply retrieve that expression from their stored linguistic experience. When they have no set expression readily available, they retrieve linguistic schemas such as *Where's the X*?, *I wanna X*, *It's a X*, *I'm X-ing it*, *Put X here*, *Let's X it*, *There is a X*, etc. and items they have previously mastered and then 'cut and past' them together as necessary for the communicative situation at hand." Tomasello (ibid: 62) observes that "such item-based linguistic expressions are stored and produced as single units (see Bybee and Scheibman 1999 for psycholinguistic evidence focused on *I dunno*)." Therefore, according to Tomasello (ibid: 74), "the child does not put together each of her utterances from scratch, morpheme by morpheme, but rather, she puts together her utterances from a motley assortment of different kinds of pre-existing psycholinguistic units." Based on the above development,

I believe that, like children, adult speakers can produce utterances by making access to a set of stored fixed phrases and analogy-oriented schematic expressions (analogous to Dabrowska's (2000: 83) formulaic frames, and Tomasello's (2000: 77) linguistic schemas).

4.2 Internal structure of *Grammatical Conditions*

The set of *Grammatical Conditions* (*GC*) defined under (7) raises at least two issues: one dealing with the internal structure, and the other pertaining to an interface among components. The first issue addresses a question such as: How are grammatical conditions arranged? The second issue raises a question like: Is there an interface among *PhonC*, *MorpC*, *SynC*, *SemC*, and *PragC*? The two issues regarding internal structure and interface are closely intertwined, therefore they should be dealt with correlatively.

The definition of *GC* assumes that neither the quadruple components nor the elements of each component are ordered. Therefore, *GC* is simply a set of unordered sets of grammatical conditions with no special precedence arrangements. This does not account for a widespread and intuitively correct view that grammar is a single whole consisting of interfacially organized components. In order to make consonant with the above view, I introduce a grammatical condition network algebra under (14).

(14) *Definition*. Grammatical condition network algebra (GCNA):

$\boxtimes$: *GC!× GC!   GC!*

$g_i \boxtimes g_j = g_i \wedge g_j$, where g is a grammatical condition, and i   j.

The GCNA entails that the binary operation $\boxtimes$ creates a set, which may consist of mutually connected grammatical conditions such as $g_i \wedge g_j$. The idea behind this is the possible existence of a grammatical-condition network, to which the speaker makes access whenever grammaticality exigencies arise. By making access to a desired network, the speaker delivers grammaticality judgments about language fragments. I assume that the grammatical-condition network is localized closely with the memory responsible for storage

and processing.   Coleman (1998: 301) claims that "storage and processing (of memories) are not clearly separable, in fact in computational neural network' models, this is the case." (The underline and the words in parentheses are provided by the present author.)   If storage and processing are inseparable, we might as well assume that the grammatical-condition network is also interwoven and closely associated with the storage and processing of memories.   Caramazza (1998: 270) proposes the Organized Unitary Content Hypothesis (OUCH), which assumes that "strongly correlated properties are represented in adjacent neural tissue and that members of natural kind categories (e.g., animals) share many properties in common, therefore the semantic properties of natural objects would be more likely to be found near each other and, consequently, more likely to be damaged together.   The OUCH predicts that the properties of members of semantic categories tend to cluster together and, therefore, tend to be damaged together, leading to category-like effects."   On the basis of OUCH, we might assume that closely related grammatical conditions tend to be tightly organized and form a network cluster.

4.3 Speech-time coordinate axis

The SA-Model as it stands does not tell much about the pragmatic side of language fragments the model produces.   By examining the definition under (1), for example, we do not know into which dialect L is grouped, nor into what time frame L is classified.   In this respect, the speech-time axis identification of language fragments is essential in accounting for grammaticality variations within a language.   For example, the grammaticality judgment of examples shown in (15) varies between speakers of American English and British English, thereby creating a hindrance to a unified account.

(15)

  a. There wanna be a few changes made round here. (Postal & Pullum, 1978: 16, footnote 7)

  b. Who do you wanna drive the car? (Pullum, 1997: 96)

Any grammatical theory with no accounting device for dialect/language variations inevitably leads to patching and darning of central mechanisms essential to the theory. As far as my knowledge goes, none of the existing grammar models such as the Minimalist Program, Derivation by Phase, HPSG, Categorial Grammar, Functional Grammar, or Cognitive Grammar have offered a workable solution for this issue. Therefore, as a possible solution for dialect/language variations, I propose a speech-time coordinate axis system. In this coordinate system, a language fragment is located in terms of a pair (s, t) of coordinates, whose initial component represents a speech sample, and the second component represents a time frame. By particularizing the coordinate as s=a Californian speech, and t=1970's, for example, we can pinpoint a language fragment as being from Californian speech in the 1970's. It should be noted here that the speech-time coordinate can be made more specific like s=a Bakersfield speech in California, and t=1976.

The above speech-time coordinate system motivates the following definition.

(16) *Definition*. Let L_x={s | (s, t) sŒ*Speech* & tŒ*Time*} be a speech-time algebra satisfying the following terms:

    i. *Time* (T)={t_i, ..., t_n}, where t_i represents a point in time, and i<n;

    ii. *Speech* (S)={s_i, ..., s_n}, where s_i represents a speech/dialect.

Given the above speech-time coordinates system, any dialect/ language can be identifiable in terms of speech and time axes. This system forces according changes in **LEX** and *Grammatical Conditions*. **LEX** is now seen as a set, which includes at least a set of individual languages, a set of languages of speech communities, and a set of all languages. The set of *Grammatical Conditions*, on the other hand, is taken to be a set, which includes a set of individual speakers' grammatical conditions, a set of speech communities' grammatical conditions, and a set of universal grammar conditions. The above idea might strike the reader as grandiose and abstract with no substance, but I believe the basic assumption is on a right track for the

construction of an integrated, all-round model of grammar.

4.4 Overview of the SA-Model

The organization of the SA-Model can be shown in a very succinct formula as:

(17) SA-Model = (***LEX***, *GRAM*, *LDS*)

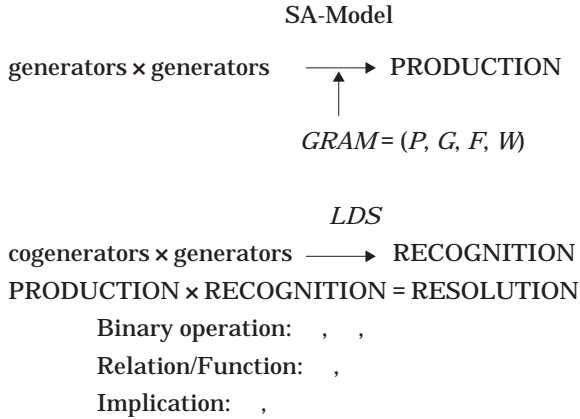The triad, however, has more to it than is readily apparent. The initial component is defined as follows:

(18) ***LEX*** = (*LEX!*, f, **1**, T)

LEX! is the closure of *LEX*. The set *LEX* is a set of generators such as [*Math theses*, S^{M}D V], [*type*, VD A^d], [*slowly*, A^d], etc. The associative binary operation f is responsible for 'Production', 'Recognition', and 'Resolution'. In Production, the binary operation binds generators and composes language fragments such as [*Math theses type slowly*, S^M]. The set *LEX* is closed under the binary operation f since for all elements in set *LEX*, the result of the binary operation f is in set *LEX*. Furthermore, ***LEX*** is taken to be a derived set. (Recall the assumption discussed in the beginning of Section 4.1: Language fragments once produced go back into the lexicon.) In Recognition, the binary operation cancels out production products by combining generators and cogenerators, the latter of which are duals of generators. In Resolution, a production product and a recognition product combine together to produce an idempotent (see the chart in (5) and (12)) as required by the definition of group[3]. The third component **1** is the identity type. The fourth component T is a set of directed types such as S^{M}D V , VD A^d , A^d, etc. *GRAM* is a grammaticality checking device and it is defined as a quadruple as shown in (9). *LDS* is Labelled Deductive System, which is a semantics-pragmatics unified model of utterance interpretation developed by Kempson (1996). Therefore, the SA-Model can be thought of as a working model, which unifies both production and recognition of language fragments involving phonology, syntax, semantics, and pragmatics together with the grammaticality

verification of the language fragments.

Given below is a schematic view of the SA-Model.   As shown at the bottom, the key mechanisms are classified into three processes: binary operation, relation/function, and implication.

(19) Schematic view of the SA-Model

<div align="center">

SA-Model

generators× generators   ——→ PRODUCTION

$GRAM$ (P, G, F, W)

LDS

cogenerators× generators ——→ RECOGNITION

PRODUCTION× RECOGNITION   RESOLUTION

</div>

        Binary operation: ,   ,
        Relation/Function: ,
        Implication: ,

5. Concluding Remarks

This paper has introduced three modifications into the SA-Model, a working model proposed in Aniya (2001): *GRAMMATICALITY*, 'stock-memory access', and 'speech-time coordinate axis'.   Due to the first device, the grammaticality checking of a language fragment p is made possible by a one-to-many mapping of p against a grammatical condition g.   The second device entails that language fragments produced by the binary operation of *Lexical Composition* go back into the set of *LEX!*.   This means that a language fragment once produced becomes a generator, a building block of language production.   This accounts for psycho-linguistic and neurolinguisitc observations in which the speaker practices language production by accessing a stored set of language fragments.   The third device specifies a language fragment in terms of two coordinate axes: speech and time.   The identification of language fragments is crucial in grammaticality

judgement.   Most grammatical conditions are subject to language/speech specific applications, though universal grammatical conditions are not.   Therefore, by specifying a language fragment in the speech-and-time axis, the overgeneralization of grammatical conditions can be avoided.

Appendix 1

Relevant grammatical conditions in *GC*

phoc[17]:

  IF   A word begins with a $\begin{bmatrix} C \\ voiced \end{bmatrix}\begin{bmatrix} C \\ voiced \end{bmatrix}$ cluster at the onset

  THEN It is ill-formed.

sync[11]:

  IF   *p* is a tough construction

  THEN It contains a tough adjective, which is followed by an optional *for*-phrase and an obligatory *to*-infinitive clause.

sync[12]:

  IF   *p* is a tough construction

  THEN Its *to*-infinitive clause contains an object gap which is coreferential with the subject of a matrix clause.

semc[22]:

  IF   *p* is a tough construction

  THEN It conveys the speaker's idea, belief, or knowledge regarding the inherent characteristic or permanent property of the matrix subject.

semc[23]:

  IF   *p* is a tough construction

  THEN Its subject allows a definite or generic reading but not an indefinite interpretation.

semc[24]:

  IF   *p* is a tough construction

  THEN Its *to*-infinitive clause obligatorily expresses self-controllable action.

pragc[51]:

    IF p contains a referential expression

    THEN Its reference is subject to a shared knowledge requirement between the speaker and the hearer.

Notes

Caramazza, A. 1998. "The Interpretation of Semantic Category-specific Deficits: What do they reveal about the organization of conceptual knowledge in the brain?" *Neurocase* 4: 265–272.

Dabrowska, E. 2000. "From Formula to Schema: The acquisition of English questions," *Cognitive Linguistics* 11–1/2: 83–102.

Hewitt, M. 2000. "Boolean Operations and Constraint Interactions in Optimality Theory," The University of North Carolina at Chapel Hill ms. Available through ROA.

Myers, J. 1999. "Lexical Phonology and the Lexicon," Concordia University ms. Available through ROA.

Nenonen, M., J. Niemi, and M. Laine. 2002. "Representation and Processing of Idioms: Evidence from aphasia," *Journal of Neurolinguistics* 15.1: 43–58.

Postal, P. and G. Pullum. 1978. "Traces and the Description of English Complementizer Contraction," *Linguistic Inquiry* 9. 1: 1–29.

Pullum, G. 1997. "The Morpholexical Nature of English *to*-Contraction," *Language* 73. 1: 79–102.

Reiss, C. 2002. "Explaining Analogy," National Chung Cheng University ms. Available through ROA.

Silveri, M. and S. Misciagna. 2000. "Langage, Memory, and the Cerebellum," *Journal of Neurolinguistics* 13, 2–3: 129–143.

Tomasello, M. 2000. "First Steps toward a Usage-based Theory of Language Acquisition," *Cognitive Linguistics* 11–1/2: 61–82.