Chinese Number-Names, Tree Adjoining Languages, and Mild Context-Sensitivity

Daniel Radzinski*
Brown University

The Tree Adjoining Grammar formalism, both its single- as well as multiple-component versions, has recently received attention as a basis for the description and explication of natural language. We show in this paper that the number-name system of Chinese is generated neither by this formalism nor by any other equivalent or weaker ones, suggesting that such a task might require the use of the more powerful Indexed Grammar formalism. Given that our formal results apply only to a proper subset of Chinese, we extensively discuss the issue of whether they have any implications for the whole of that natural language. We conclude that our results bear directly either on the syntax of Chinese or on the interface between Chinese and the cognitive component responsible for arithmetic reasoning. Consequently, either Tree Adjoining Grammars, as currently defined, fail to generate the class of natural languages in a way that discriminates between linguistically warranted sublanguages, or formalisms with generative power equivalent to Tree Adjoining Grammar cannot serve as a basis for the interface between the human linguistic and mathematical faculties.

1. Introduction

In recent years, we have seen in the linguistic literature a number of arguments; e.g., Culy (1985), Huybregts (1984), Shieber (1985), which purport to demonstrate that the class of N(atural) L(anguage)s is not generated by formalisms of C(ontext)-F(ree) power. In the context of NLs, little has been said regarding the generative inadequacy of formalisms such as single- or multiple-component T(ree) A(djoining) G(rammar)s (Joshi 1985, 1987), H(ead) G(rammar)s (Pollard 1984; Roach 1987), L(inear) I(ndexed) G(rammar)s (Gazdar 1988), or Combinatory Categorial Grammars (Steedman 1985, 1987). These formalisms are among the so-called Mildly C(ontext)-S(ensitive) G(rammar)s since they are non-CF; i.e., strictly CS, but only to a limited extent. More will be said about these grammars and their object languages below.

Notable exceptions to the trend of demonstrating only non–context-freeness are Kac (1987) for English and Manaster-Ramer (1987a) for Dutch and German. In addition to demonstrating non–context-freeness, both these studies argue that the constructions used for their respective argumentations can serve as a basis for demonstrating that the NLs in question are generated neither by TAGs nor by HGs. However, these constructions rely crucially on coordination, and our current understanding of the properties of coordination is far from satisfactory. In this paper we show that the number-name system of Chinese, specifically of the Mandarin dialect, is neither a single- nor a multiple-component TAL,[1] raising doubts about whether it could be

* Department of Cognitive and Linguistic Sciences, Providence, RI 02912

1 Henceforth we use the acronym TAL to refer uniquely to single component Tree Adjoining Language, and MCTAL to refer to Multiple Component TAL. (Likewise TAG and MCTAG, *mutatis mutandis*).

considered a Mildly CSL at all. Our argument relies in no way on overt coordination operators.

In Section 2 we present an argument originally proposed in Zwicky (1963) wherein he showed that the English naming system for cardinal numbers is a non-CFL. We discuss possible objections to his claims. Some Chinese data are presented in Section 3. In Section 4 we deal with a few Mildly CS formalisms and show that the Chinese number-name system (henceforth N(umeric) C(hinese)) is a non-TAL. In Section 5, we discuss additional grammar formalisms and show that NC is not a M(ultiple) C(omponent) TAL. We also investigate if NC can be characterized as a Mildly CSL. We discuss the linguistic relevance of our formal results in Section 6. Finally, Section 7 presents the conclusions of this study.

2. Zwicky (1963) and Objections Thereunto

Zwicky (1963) discusses some constructions of names for cardinal numbers that are not generated by a CFG. The one he labels (1) resembles the structure of very large number-names in English (and other NLs):

$$NT^n(,NT^{n-1})\ldots(,NT)(,N) \tag{1}$$

In this construction, N indicates a number between 1 and 999, T is an abbreviation for *thousand*, commas indicate an intonational pause, and everything within parentheses is optional. This construction could be characterized as follows:

(i) Given a system in English, for example, where *thousand* is used as the largest single word for a number, *million* would be represented as *thousand thousand*, (Amer.) *billion* as *thousand thousand thousand*, (Amer.) *trillion* as *thousand thousand thousand thousand, etc., ad infinitum.*

(ii) In a system like (i), larger clusters of *thousand* must precede smaller clusters of *thousand* in the same manner that *decillion* must precede *trillion*, which must precede *million*, which must precede *thousand* in the standard English number-name system using single-words for numbers of higher values.

Zwicky relates construction (1) to the formal language P:

$$P = \{x|x = b^n(ab^{n-1})\ldots(ab^2)(ab),\ \text{where } n = 1,2,3,\ldots\}$$

He proceeds, *inter alia*, to prove that P is non–context-free. A conclusion from his study is that the sublanguage of English encompassing the names for cardinal numbers is strictly context-sensitive.

Although Zwicky's mathematical argumentation is sound, room is left for some investigators to cast doubts on whether his claims bear in any significant way on NL. The empirical basis for Zwicky's argument rests largely on whether characteristics (i) and (ii) are indeed linguistically real. There has been much controversy over the status of these characteristics. For example, Merrifield (1968, p. 91) states the following:

> In working with a language isolate such as a system for naming numbers, several things should be kept in mind.
> In the first place, such a system differs from the larger grammar of which it is but a segment in not being indefinitely recursive. A grammar of a natural

> language accounts for an infinite number of utterances; a grammar of number names apparently does not. The latter is limited by the number of linguistic primitives of the sort 'billions,' 'trillions,' 'quadrillions,' etc., which it includes. And though a mathematician is presumably able to write down in mathematical notation an infinitely large set of numbers, when he attempts to give names to the members of the set in a natural language, he is limited by the number of primitives at his disposal.

Greenberg (1978, p. 253) expresses Merrifield's assertion as the generalization that, "every language has a numeral system of finite scope." Greenberg then proceeds to claim that the largest expressible natural number in American English is 10^{36-1} "assuming that, as in most dictionaries of AMERICAN ENGLISH, the lexical item with the highest numerical value is 'decillion'."[2]

Thus, Merrifield and Greenberg take the view that there is an upper limit on linguistically expressible number-names. Hence, by this view, characteristic (i) appears not to be linguistically warranted. Hurford (1975, p. 4) suggests otherwise:

> Now it can be argued that the class of number expressions in any given language is infinite. Intuitions of language users differ on the matter of whether the set of number expressions in their language is infinite. The crux of the matter is the question whether the names for very high numbers are in fact wellformed. In English, for example, the expression *two billion billion, five hundred and five* may be felt by some speakers to be quite wellformed, though of course unlikely to be observed, whereas other speakers may object that it is not wellformed.

Accordingly, characteristic (i) is linguistically warranted for at least some speakers. Hurford (*ibid.*) takes in fact this position:

> It will become obvious as we proceed that the particular systematic characteristics which are evident in natural language number-name systems tend to project the existence of infinite sets of number-names and a higher limit to the value of wellformed number-names can only be stated in a fairly *ad hoc* arbitrary manner.

Epstein (1978, p. 123) contests this claim by arguing:

> Contrary to what Hurford claims, there are a finite number of these [numerical expressions in English]. *Ten to the trillionth power*, for example, has no corresponding counting expression.

Hurford (1979, p. 42) responds:

> This is a misconception. It would be similarly wrong to assert that there is no single English sentence giving the full names, addresses, heights, weights, and IQ's of all UK citizens at midnight on March 1st, 1978. Such a sentence would be impossibly long to utter, but that is not a restriction which need be stated as part of English grammar, or indeed of general linguistic theory. If the highest-valued number word in your vocabulary is *trillion*, and you want to express higher numbers, you just string together enough *trillions* to get you there. Nobody, as a

2 Incidentally, Merriam-Webster's *Third New International Dictionary of the English Language* (in the Fifteenth Edition of *Encyclopedia Britannica*) has a table on p. 1549 labeled "DENOMINATIONS ABOVE ONE MILLION." This table contains entries up to 'centillion,' which happens to be 10^{303} in the American system and 10^{36} in the British one. The number 10^{36} is expressed in the American system as 'undecillion.'

> matter of performance could possibly utter enough *trillions* in sequence to make a counting expression expressing *ten to the trillionth power*, but no grammatical theory need concern itself with this fact.

While we fully agree with Hurford on this matter, the contrived nature of "strung-together" *trillions* or *thousands* or any other single-word number-name seems to lead some investigators into rejecting its empirical validity in English. For example, Brainerd (1971, p. 208) mentions the following:

> The collection of numerical expressions in most languages, as in English, are basically finite. Thus in English we must ultimately coin new *'illions'* if we are to transcend our [finite] system of number names. And where are these to come from when we have run out of Latin prototypes?[3] In some languages there are, however, purely linguistic devices which allow for an infinitude of numerical expressions. For example, in Chinese, *wan* is used for 10^4 and *wan wan* for 10^8. Presumably we can continue ad infinitum, *wan wan wan* 10^{12}, *wan wan wan wan* 10^{16}, etc.

Thus, while the linguistic status of strung-together number-names in English might be questionable, Brainerd suggests on the basis of empirical evidence that this is not so in Chinese. Characteristic (i) appears then to be linguistically warranted in at least one NL, currently the most widely natively spoken one.[4] In the next section, we take a look at some more data on number-names in Chinese. This will help us to support our conclusion regarding the linguistic reality of characteristic (i) as well as to determine the empirical status of characteristic (ii). Furthermore, the data will serve as a basis for the formal arguments to be presented in Sections 4 and 5. In Section 6, we shall return to discuss the linguistic status of characteristic (ii).

3. Data[5]

The number-name in Chinese for '10,000' is *wan*. The number-name for '100,000,000' is either *wan wan* or *yi*.[6] Contrary to Brainerd's (1971) presumption, some native speakers find longer number-names such as *wan wan wan* for 10^{12} or *wan wan wan wan* for 10^{16} awkward due to the existence of the numerically higher-valued single-word number-

3 Alexis Manaster-Ramer (p.c.) has provided an answer to this question: These are to come from indefinite iteration. In other words, once we have reached the limit of Latin prototypes, we can still form new number-names by stringing together Latin prototypes word-internally. In fact, such a number-word-formation process would be a morphological analogue of (i) at the single-word level.

4 We also find the following claims in the literature: (a) Menninger (1969, p. 129) reports: "The Gothic word for 'hundred'... is not *hund* but is represented by two full word forms: *taihun-taihund*, 'a ten of tens,' in which the grouping 'ten' is counted." Nevertheless, since Gothic is dead, our knowledge of it is limited to a finite nonproductive corpus of items. From this one example, we cannot conclude that the Gothic cardinal number-name system permitted an indefinite amount of concatenations of number names. (b) Brainerd (1967, p. 43) reports: (due to Gr. C. Moisil) "It has been suggested that by using the expressions *milioane de milioane, milioane de milioane de milioane*, etc. a name can be given for every natural number [in Rumanian]." Strung-together number-names in Rumanian sound less awkward due, most likely, to the phonetically realized separator *de* between the single-words. Yet, while such expressions sound fine, Rumanian has no direct analogue of Chinese's *wan wan*, in the sense of a commonly-used strung-together number-name. As will be shown in Section 3, *wan wan* is a perfectly common and natural name for 10^8 in Chinese.

5 For elaborations on Chinese names for cardinal numbers beyond what we present here, see, for example, Chao (1968, pp. 567–575) or Henne *et al.* (1977, pp. 239–242). For grammars generating Chinese number-names, see Brainerd (1966a) and Brainerd and Peng (1968). For a historical survey, see Needham (1959, pp. 1–90).

6 When used as the number-name for '100,000,000,' *yi* is usually pronounced in the fourth tone. If used as the number-name for '1,' it is pronounced in the first tone.

name *yi*. Thus, 10^{12} is more naturally expressed as *wan yi* and 10^{16}, as *yi yi*. Yet there exists a single-word number-name with a higher numerical value than *yi*. This single-word is *zhao*, meaning 10^{12}, i.e., Amer. 'trillion.' This number-name appears to be the numerically highest-valued single-word number-name in use in the modern language.[7] The most natural way of linguistically expressing a number exponentially greater than 'trillion'—as "natural" as one gets with such high numbers—is by stringing together instances of *zhao*. This is the same tool used for expressing 10^8 as *wan wan*, which is, as previously mentioned, a frequently used synonym for *yi*. Thus, the unbounded stringing-together of instances of *zhao* is simply an extended instantiation of a method frequently used in Chinese for expressing a more common lower-valued number.

Examples (a) and (b) below are well-formed Chinese number-names, while example (c) is not:[8]

(a)	*wu*	*zhao*	*zhao*	*wu*	*zhao*				
	five	trillion	trillion	five	trillion				
	(i.e. 5,000,000,000,000,005,000,000,000)								
(b)	*wu*	*zhao*	*zhao*	*zhao*	*zhao*	*zhao*	*wu*	*zhao*	*zhao*
	five	trillion	trillion	trillion	trillion	trillion	five	trillion	trillion
	zhao	*zhao*	*wu*	*zhao*	*zhao*	*zhao*	*wu*	*zhao*	*zhao*
	trillion	trillion	five	trillion	trillion	trillion	five	trillion	trillion
	wu zhao								
	five trillion								
(c)	**wu*	*zhao*	*zhao*	*wu*	*zhao*	*zhao*	*zhao*		
	five	trillion	trillion	five	trillion	trillion	trillion		

Similarly, example (d) is well-formed, while (e) is not. Both of these are examples of number-names containing adjacent 'columns' in which the number of instances of *zhao* in one of the 'columns' is neither the predecessor nor the successor of the number of instances of *zhao* in the other column:

(d)	*wu*	*zhao*	*zhao*	*zhao*	*zhao*	*wu*	*zhao*	*zhao*	
	five	trillion	trillion	trillion	trillion	five	trillion	trillion	
(e)	**wu*	*zhao*	*zhao*	*wu*	*zhao*	*zhao*	*wu*	*zhao*	*zhao*
	five	trillion	trillion	five	trillion	trillion	five	trillion	trillion
	zhao	*zhao*							
	trillion	trillion							

The well-formed number-names that we have seen all follow a pattern in which larger clusters of *zhao* precede, from left to right, smaller clusters of *zhao*, while the ill-formed number-names do not adhere to such a requirement. All the well-formed number-

7 Needham (1959, p. 87) (also reproduced in Brainerd (1966a, p. 42)), gives a list of single-word number-names up to 10^{44}. Yet all those beyond *zhao* are extremely archaic and are most likely not part of the vocabulary of a present-day native speaker.

8 In all of our examples, we omit possible instances of *ling*. This is a morpheme with numerous meanings—as is not uncommon in Chinese—including *'zero,'* 'remainder,' and, when used in between other morphemes in a number-name, we could say it serves as a type of conjunction. It is used in case of a gap of more than one decimal order within a number-name. Yet, if there is more than one such gap in a number-name, generally only one *ling* is used and this is in place of the rightmost gap. Number-names for multiples of '10,' as are all of our examples, generally, though not necessarily, lack such use of *ling*. In any case, its use would not affect in any significant way the formal argumentation to be presented in Sections 4 and 5. (On the semantic evolution of *ling*, see Needham (1959, pp. 16–17)).

names composed only of instances of *wu* and *zhao* form the set J:

$$J = \{wu\ zhao^{kl}\ wu\ zhao^{k2} \ldots wu\ zhao^{kn} \mid k1 > k2 > \ldots > kn > 0\}.$$

This clearly lends empirical support to characteristics (i) and (ii) of Section 2. Yet, whether characteristic (ii) is a consequence of a linguistic constraint or of some other type of constraint is a question we leave open till Section 6.

4. Numeric Chinese and Tree Adjoining Languages

In light of recent, and not so recent, definitions of languages lying between CFLs and CSLs, we are now capable of demonstrating that it is not only CFGs that fail to generate Numeric Chinese, but even more powerful grammars as well. We shall argue here that Numeric Chinese is not a TAL. The weak equivalence of TAGs, a modified version of Head Grammars and Linear Indexed Grammars has been shown in Vijayashanker (1988), while the weak equivalence of these three formalisms and Combinatory Categorial Grammars has been shown in Weir (1988) and Weir and Joshi (1988).

TAGs perform certain manipulative adjunction operations on tree structures. HGs are similar to CFGs except that they allow head-wrapping operations in addition to the usual concatenation operation used in CFGs. LIGs are a restricted type of IGs, which were introduced by Aho (1968) as an extension of CFGs. In addition to terminals and nonterminals, IGs also have indices, or flags, which can be used in rewrite operations. Their use may lead to the generation of non-CFLs. LIGs restrict the way in which indices may be used. Combinatory Categorial Grammars are an extension of classical Categorial Grammars. The latter were introduced into the linguistic literature by Bar-Hillel (1953), based largely on the work of Lesniewski (1929) and Ajdukiewicz (1935) in the context of philosophical logic. The classical Categorial Grammar formalism is characterized by the use of the combinatory operation of functional application, and is extensionally equivalent to the CF formalism. Combinatory Categorial Grammars are augmented with the combinatory operation of functional composition used in a restricted way.

In addition to generating all CFLs, these four types of grammars generate non-CFLs such as $\{a^n b^n c^n\}$, $\{a^n b^n c^n d^n\}$, and $\{WW|W \in (a+b)^*\}$. They exclude, however, languages such as $\{a^n b^n c^n d^n e^n\}$ and others exhibiting a dependency of more than four **columns**, and $\{WWW|W \in (a+b)^*\}$ and other languages exhibiting more than a single copy. Since these four formalisms have been proved to be weakly equivalent, i.e. they generate the exact same stringsets, we will limit our discussion below to TAGs.

We now proceed to prove that NC is not a TAL in the weak sense. We begin by considering the following regular language R:

$$R = \{wu\ zhao^+\ wu\ zhao^+\ wu\ zhao^+\ wu\ zhao^+\ wu\ zhao^+\}$$

Strings of R may be described as the concatenation of five columns, each column comprising one instance of *wu* followed by one or more instances of *zhao*. Let H = NC ∩ R. Clearly,

$$H = \{wu\ zhao^n\ wu\ zhao^m\ wu\ zhao^l\ wu\ zhao^k\ wu\ zhao^j \mid n > m > l > k > j > 0\}$$

Strings of H may be described as the concatenation of five columns, each column comprising one instance of *wu* followed by one or more instances of *zhao* and each

column having more instances of *zhao* than any column to its right. For notational convenience, we now define the homomorphism h such that:

$$\begin{aligned} h(wu) &= a \\ h(zhao) &= b \end{aligned}$$

Let L = h(H). Clearly,

$$L = \{ab^n ab^m ab^l ab^k ab^j \mid n > m > l > k > j > 0\}$$

Lemma 1
L is not a TAL.

Proof
We apply the pumping lemma for TALs and arrive at a contradiction. This pumping lemma is given in Vijayashanker (1988, pp. 96–101) as Theorem 4.7:

> **Pumping Lemma for TALs**
>
> If L is a TAL, then there is a constant n such that if $z \in L$ and $|z| \geq n$ then z may be written as $z = u_1v_1w_1v_2u_2v_3w_2v_4u_3$ with $|v_1w_1v_2v_3w_2v_4| \leq n$, $|v_1v_2v_3v_4| \geq 1$ such that for all $i \geq 0$, $u_1v_1^iw_1v_2^iu_2v_3^iw_2v_4^iu_3 \in L$.

It follows from this theorem that for any string in a TAL longer than a determined constant for that TAL, the string has at most four pumpable substrings. We now demonstrate that some long strings of L require more than four pumpable substrings in order to remain within L after pumping. Assume L is a TAL. Where n is the constant referred to by the lemma, corresponding to our L, consider the string $z = ab^{n+4}ab^{n+3}ab^{n+2}ab^{n+1}ab^n$ which is in L. Let us now number the columns of z 1–5 from left to right, where a column is an *a* followed by *b*s. Since all strings of L contain exactly five instances of *a*, $v_1v_2v_3v_4$ cannot contain *a*s, else these could be pumped, yielding strings outside of L. Thus, $v_1v_2v_3v_4$ must consist solely of *b*s. Since there are at most four pumpable substrings (v_1, v_2, v_3, and v_4) and z comprises five columns, at least one column of z will not contain a pumpable substring. We can now look at the following two exhaustive cases:

A. A column c with no pumpable substrings to the left of some column d with at least one pumpable substring.

B. A column c with no pumpable substrings to the right of some column d with at least one pumpable substring.

Case A. Let c be column 1 and let v_1, v_2, v_3, and v_4 be in columns 2, 3, 4 and 5, respectively. Let d be column 2. It suffices to pump up once, i.e. set $i = 2$, in order for column 1 to cease being longer than column 2, a necessary condition for all strings in L. As long as there is at least one pumpable substring in some column d to the right of c, any other instantiation of $v_1v_2v_3v_4$ will yield a result in which a column will not be longer than another column to its right when setting $i = 2$.

Case B. Let c be column 5 and let v_1, v_2, v_3, and v_4 be in columns 1, 2, 3, and 4, respectively. Let d be column 4. Here pumping up once or more, i.e. setting $i > 1$,

will not suffice, because the newly created strings remain in L, since they meet the well-formedness conditions of L: Each column contains more instances of *b* than any one of the columns to its right. What we need in this case is to resort to pumping down, i.e. setting $i = 0$. In such a case, column 5 will cease being shorter than column 4, a necessary well-formedness condition for all strings in L. As long as there is at least one pumpable substring in some column d to the left of c, any other instantiation of $v_1v_2v_3v_4$ will yield a result in which a column will not be longer than another column to its right when $i = 0$.

These two cases exhaust all possible pumpings of non-fixed positions. Both lead to contradictions. Hence, the assumption that L is a TAL cannot be true.[9] ■

Theorem 1
NC is not a TAL.

Proof
L = h(NC ∩ R). R is an RL. TALs are closed under intersection with regular languages (*cf.* Theorem 3.6 in Vijayashanker (1988 pp. 76–77)) and under arbitrary homomorphism (*cf.* Corollary 4.1 in *ibid.*, p. 94). By Lemma 1, L is not a TAL. Therefore, neither is NC. ■

5. NC, ILs, MCTALs and Mildly CSLs

After demonstrating in the previous section that NC is not a TAL, we are now faced with the following question: What type of non-TAL is it? Under the assumption that the only nonfinite phenomena in NC are strung-together number-names (of the type we have investigated), we would conjecture it is included in the class of Indexed Languages. The following IG(H) generates H, NC's proper subset responsible in our argument for NC's being a non-TAL: (our notation is a minor variant of the one used by Hopcroft and Ullman 1979, pp. 389–390).

IG(H)=({S,T,U,V,W,X,Z}, {*wu*, *zhao*},{f,g}, P, S) where P comprises:

S → Tg	W → Wf
T → Tf	W → Xf *wu* Z
T → Uf *wu* Z	X → Xf
U → Uf	X → *wu* Z
U → Vf *wu* Z	Zf → *zhao* Z
V → Vf	Zg → *zhao*
V → Wf *wu* Z	

However, H is not the only proper subset of NC having the characteristic of more than four columns. We could just as well have intersected NC with a different regular language containing more than five sequences of *wu zhao*$^+$, thus yielding a proper

9 Since our proof has made no use of the condition in Theorem 4.7 requiring the length of $v_1w_1v_2v_3w_2v_4$ to be less than or equal to the constant n, perhaps even some weaker version of the pumping lemma for Tree Adjoining Languages lacking that condition would suffice for our purposes. In fact, such a version, attributed to Vijayashanker, is mentioned in passing by Weir (1987, p. 5). Furthermore, the pumping lemma for Head Languages, Theorem 45 in Roach (1987, pp. 321–325), lacks a condition of this sort (though added subsequently on p. 336). We have nevertheless opted for Vijayashanker's (1988) Theorem 4.7 because of its succinctness and perspicuity.

subset of NC with a number of columns equal to the number of *wu zhao*$^+$ sequences in the intersected regular language. If this number were fixed, then we could simply define more nonterminals and add more productions to IG(H) yielding a language of, say, 6, 275, or 10^{9438} columns, all depending on the fixed number of columns we wish to have. But since the number of columns in our construction is indefinite, such an IG would not suffice. For a proper subset J of NC, with arbitrarily many columns as described in Section 3, we need the following simpler IG(J):

IG(J) = ({S,T,Z}, {*wu*, *zhao*}, {f, g}, P, S) where P comprises:

S → Tg
T → Tf
T → Tf *wuZ*
T → *wuZ*
Zf → *zhao Z*
Zg → *zhao*

Are there known formalisms that generate NC, but not the entire class of ILs (or some non-ILs)? We attempt to answer this question by first considering M(ultiple) C(omponent) TAGs and other similar formalisms. MCTAGs have been discussed in the literature; for example, in Joshi (1987, pp. 110–113), Joshi *et al.* (1989, pp. 13–15), Vijayashanker *et al.* (1987, pp. 106–107), and Weir (1987, pp. 30–33; 1988, pp. 31–39). These are grammars whose generative capacity exceeds that of TAGs, since adjunction in MCTAGs is performed simultaneously on sequences of trees rather than on single trees. Pollard (1984, pp. 210–215) has defined a formalism called Generalized CFG. This formalism is an extension of CFG wherein the compositional operations in the grammar's production set need not be limited to functions of concatenation or syncategorematic insertion of terminals. Kasami *et al.* (1988, pp. 9–11) have shown that Generalized CFGs generate all the recursively enumerable sets. In addition, they have defined a restricted Generalized CFG called M(ultiple) CFG, which is stronger than Head Grammar. (HG is a Generalized CFG whose compositional operations are restricted to those of CFG plus head-wrapping.) Independent of Kasami *et al.*'s research, Vijayashanker *et al.* (1987, pp. 108–111) and Weir (1988, pp. 90–110) have introduced a formalism called L(inear) C(ontext-)F(ree) R(ewriting) S(ystem)s, whose definition turns out to be the same as that of MCFG. The composition operations in LCFRSs, or MCFGs, are restricted to being *linear* and *nonerasing*. In other words, they lack functions that 'copy,' 'erase,' or 'restructure' unbounded components of their arguments, and as such are 'size' preserving. Weir (1988, pp. 101–106) has shown that the classes of stringlanguages generated by MCTAGs and LCFRSs are the same. Given the equivalence of MCTALs, LCFRLs, and MCFLs, we focus our discussion below on MCTAGs, yet refer to LCFRSs and MCFGs if necessary or convenient.

Manaster-Ramer (1987a, p. 233) has pointed out that "[the MCTAG formalism] can handle quintuple and higher counting dependencies, but it still cannot deal with ... unbounded counting dependencies." That is, this formalism generates languages with a fixed number of columns, and not ones with an arbitrary number of them. Thus, while MCTAGs generate H, they do not generate J. Accordingly, Numeric Chinese cannot be accounted for by any MCTAG. We now proceed to prove this claim. We begin by considering the regular language R′:

$$R' = \{(wu\ zhao^+)^+\}$$

Strings of R′ may be described as the concatenation of any arbitrary non-zero number of columns, each column comprising one instance of *wu* followed by one or more instances of *zhao*. Let J = NC ∩ R′. Clearly, as presented already in Section 3,

$$J = \{wu\ zhao^{kl} wu\ zhao^{k2} \ldots wu\ zhao^{kn} \mid k1 > k2 > \cdots > kn > 0\}.$$

Strings of J may be described as the concatenation of any arbitrary non-zero number of columns, each column comprising one instance of *wu* followed by one or more instances of *zhao*, and each column has more instances of *zhao* than any column to its right. For notational convenience, let K = h(J), where the homomorphism h is as defined previously in Section 4. Clearly,

$$K = \{ab^{k1}ab^{k2} \ldots ab^{kn} \mid k1 > k2 > \cdots > kn > 0\}.$$

Lemma 2
K is not a MCTAL.

Proof
We apply a pumping lemma for MCTALs arriving at a contradiction. A pumping lemma for Multiple CFLs is given in Kasami *et al.* (1988, pp. 18–20) as Lemma 3.4. In order to achieve maximum analogy and uniformity with Vijayashanker's (1988) pumping lemma for TALs, we rephrase Kasami *et al.*'s Lemma 3.4, making it applicable to MCTALs:[10]

Pumping Lemma for MCTALs

If L is a MCTAL, then there are constants n and m such that if $z \in L$ and $|z| \geq n$ then z may be written as $z = u_1v_1w_1s_1u_2v_2w_2s_2u_3 \ldots u_mv_mw_ms_mu_{m+1}$ with $\sum_{j=1}^{m} |v_js_j| \geq 1$ such that for all $i \geq 0$, $u_1v_1^iw_1s_1^iu_2v_2^iw_2s_2^iu_3 \ldots u_mv_m^iw_ms_m^iu_{m+1} \in L$.

10 In fact, Kasami *et al.*'s pumping lemma is weaker than ours. Their Lemma 3.4 (p. 18) is phrased as follows: For any ... mcfl L, if L is an infinite set then there exist some $u_j \in T^*$ [where T is a finite set of terminals] $(1 \leq j \leq m + 1)$ [for a fixed m associated with L], v_j, w_j, $s_j \in T^*$ $(1 \leq j \leq m)$ which satisfy the following conditions: (1) $\sum_{j=1}^{m} |v_js_j| > 0$, (2) for any non-negative integer i,

$$z_1 \stackrel{\Delta}{=} u_1v_1^iw_1s_1^iu_2v_2^iw_2s_2^iu_3 \ldots u_mv_m^iw_ms_m^iu_{m+1} \in L.$$

There is an essential quantificational difference between Lemma 3.4 and our lemma. Lemma 3.4 claims that every infinite Multiple CFL has some strings, i.e. *at least one and maybe more*, which satisfy the pumping conditions. Such a lemma suffices, for example, for proving the non-MCFness of a language with an indefinite amount of columns in which all columns are equal in length, since such a language does not contain a single string with a fixed number of pumpable substrings. [See Lemma 3.5 in Kasami *et al.* (1988, p. 20).] However, such a lemma is not strong enough to show that a language, such as K, with an indefinite amount of columns in which every column is longer than any column to its right, is not an MCFL. For example, the following string $z \in K$, satisfies the pumping conditions of Lemma 3.4:

$$z = ab^{1000000(n+2m)}ab^{n+2m-1}ab^{n+2m-2} \ldots ab^{n+2m-(2m-1)}ab^n$$

If v_1 is in the leftmost column of z and we only pump v_1, as allowed by the lemma, then we still remain in the language, for all $i \geq 0$. Our lemma, on the other hand, claims that the pumping conditions must be satisfied by *all* the strings of a MCTAL longer than a determined constant for that MCTAL. With our lemma, it is enough to find even one single string, longer than the constant, that does not satisfy the pumping conditions, in order to prove, as we do below, that the language containing that string is not a Multiple Component Tree Adjoining Language.

Interestingly, it is not necessary for us to prove our version of the pumping lemma: Kasami *et al.* have already done so! Their proof crucially depends on the following assumption (p. 19): "Let us

It follows from this lemma that, for any string in a MCTAL that is longer than a determined constant n associated with this MCTAL, the string has at most $2m$ pumpable substrings, where m is a fixed number associated with the particular MCTAL. We now demonstrate that some long strings of K require more than $2m$ pumpable substrings in order to remain within K after pumping. Assume K is a MCTAL. Where n and m are the constants for K corresponding to those referred to by the lemma, consider the string $z = ab^{n+2m}ab^{n+2m-1}ab^{n+2m-2} \ldots ab^{n+2m-(2m-1)}ab^n$ which is in K (since the number of columns in strings of K is not necessarily fixed). Let us now number the columns of z 1 thru $2m + 1$ from left to right. The pumpable substrings, i.e. $v_j s_j$ for all $j = 1$ to m, cannot contain any instances of a, since pumping as would yield new strings with columns longer than columns to their left, columns shorter than columns to their right, columns equal in length to other columns, or two adjacent as. None of these are in K. Thus, the pumpable substrings must consist solely of bs. The remainder of this proof is analogous to that of Lemma 1, *mutatis mutandis*. ■

Theorem 2
NC is not a MCTAL.

Proof
K = h(NC ∩ R'). R' is an RL. MCFLs are closed under intersection with regular languages and under substitution [*cf.* Theorem 3.9 in Kasami *et al.* (1988, p. 21)]. Homomorphism is simply a special case of substitution. MCFLs = MCTALs. By Lemma 2, K is not a MCTAL. Therefore, NC is not a MCTAL. ■

We may now begin to answer the question posed at the outset of this section: NC is not a MCTAL, but still appears to be an IL.[11] We fail, however, to find a well-studied and attractive formalism that would seem to generate NC without generating the entire class of ILs (or some non-ILs).[12] Now, we are faced with the following question: if NC

---

consider a derivation tree t of $z \in L$ such that $|z| \geq q^{|N|+1}$." A consequence of this assumption is that z satisfies the pumping conditions, hence there is at least one such string in any infinite Multiple Context-Free Language. Yet, their z was chosen arbitrarily from the set of strings in L longer than a determined constant for L (in their case $q^{|N|+1}$). Hence, ***all*** strings longer than that constant satisfy the pumping conditions. Thus, in fact, Kasami *et al.* proved implicitly something stronger than what they had claimed to prove.

11 Kac (1987, p. 451) mentions: "It nonetheless remains to be seen what sort of increase in power over that possessed by these classes of grammars [i.e. HG and TAG] is sufficient to handle *respectively* sentences." The issue is raised after noting that certain instances of the English *respectively* construction map on to the five-column language $\{b^h a^n b^m c^l d^k \mid 2 \leq h \leq n \leq m \leq l < k\}$ and to additional multicolumn languages without limit on the number of columns they contain. An indirect consequence of our last conclusion is that, given the close resemblance between Kac's five-column language and our L, as well as, of course, between any of his other multicolumn languages and our corresponding subsets of K, we can now suggest that the increase in power sought will have to be of the type that allows for the generation of non-MCTALs without allowing for the generation of the entire class of ILs (or some non-ILs).

12 Such a formalism might be of the type similar to an augmented version of Combinatory Categorial Grammar as proposed by Steedman (1985) and discussed in Weir (1988, pp. 128–131) (together with suggested extensions to Linear Indexed Grammars) and Weir and Joshi (1988, p. 284). There do exist, in fact, two formalisms known to the author, which are restricted types of IGs, that generate the phenomena under discussion:

a) If we make one simple change to the definition of our IG(J) so that Z is not a member of the set of *nonterminals*, but rather forms a singleton of *intermediates*, then what we end up with is an *R*-grammar. *R*-grammars are discussed in Aho (1968, p. 670). Only *intermediates* can consume indices and once an index is consumed, no new indices can be generated by the *intermediates*. Also, all productions involving an index consumption must be right linear in form (as are such rules in our IG(J)). Nevertheless, neither Aho (1968) nor Aho (1969), wherein these restrictions on IG are also discussed,

is not a MCTAL, then is it Mildly Context-Sensitive? To answer this, we must consider the properties that make a CSL mild. On this matter, Joshi (1985, p. 225) writes:

> I would like to propose that the three properties
>
> 1. limited cross-serial dependencies,
> 2. constant growth, and
> 3. polynomial parsing
>
> *roughly* characterize a class of grammars (and associated languages) that are only slightly more powerful than context-free grammars (context-free languages). I will call these *mildly context-sensitive grammars (languages).*

Accordingly, the entire class of MCTALs is Mildly CS, but not the entire class of ILs, which, *inter alia*, includes languages that are non-constant growth such as a^{2^n}. We now investigate if NC possesses the Mildly CS properties.

Polynomial Parsing. Since parsing is dependent on a grammar and we are not dealing here with a specific grammar but rather with a language, we can only consider polynomial recognition. Weir (1988, pp. 98–101) gives the following theorem:

> **Theorem 4.4.1** If L is a language generated by a grammar of some formalism that is a LCFRS, then L can be recognized in polynomial time on a turing machine.

By Weir's Theorem 4.4.1, strings in NC generated by a MCTAG are polynomially recognizable. We will now show that this is true for strings in NC not generated by a MCTAG. Such strings are precisely those that contain an indefinite amount of columns of strung-together number-names, and the set of these strings can be mapped to K via simple homomorphisms. The following transitions of a T(wo-) H(ead) F(inite)

---

shows explicitly and unequivocally that the class of languages generated by such restricted IGs is indeed *properly* included within the class of ILs. In other words, although not intuitively likely, it may still be the case that the stringlanguage classes are equivalent.

b) Fischer (1968) discusses Macro-like Grammars. One type of these is the O(utside) I(n) Grammars, which Fischer proves to be weakly equivalent to IGs. A particular restriction on the definition of OI Grammars yields Basic Grammars, whose class of object stringlanguages is properly contained in the class of ILs. This is because Basic Grammars generate only those languages that are generated both by OI Grammars and I(nside) O(ut) Grammars, i.e. their intersection, and there are OI languages (i.e. ILs) that are not IO languages. A restriction on the definition of Basic Grammars yields Linear Basic Grammars. Proper inclusion of the latter in the former is only conjectured. The following rather simple Linear Basic Grammar (and, of course, Basic Grammar) productions generate J (where the symbol 'λ' refers to the empty string and the symbol '|' is a disjunction of right-hand sides of rules having a common left-hand side):

$$S \rightarrow F(wu\ zhao, \lambda)$$

$$F(x, y) \rightarrow F(x\,zhao, y) \mid F(x\,zhao, xy) \mid xy$$

However, Fischer conjectures that the class of Linear Basic Languages does not include the class of CFLs. A formalism that does not generate all CFLs would, most likely though perhaps arguably, seem unattractive for our purposes. Furthermore, although the class of Basic Languages does include the class of CFLs, both the class of Basic Languages and the class of Linear Basic Languages are not closed under inverse homomorphism and, as such, do not each form a full A(bstract) F(amily of) L(anguages). As Savitch (1989, p. 255) comments, "... in some circles [being a full AFL] invests the class [of languages] with a certain respectability." Thus, again, a formalism whose class of object languages does not form a full AFL would, most likely though perhaps subject to counterarguments, seem unattractive for our purposes. The other language classes discussed in this paper each form a full AFL. (For more on Macro-like Grammars, see Fischer (1968).)

A(utomaton) efficiently accept exactly K in linear time (less than $2n$), *ipso facto*, polynomial time:

(q0,a,a,q1)

(q1,λ,b,q2)

(q2,λ,b,q2)

(q2,λ,#,q*f*)

(q2,λ,a,q3)

(q3,b,b,q4)

(q4,b,b,q4)

(q4,b,a,q5)

(q4,b,#,q*f*)

(q5,b,λ,q5)

(q5,a,λ,q3)

A THFA consists of a finite control, an input tape, and two read-only heads that move only left to right. It is started in state q0 with both heads on the tape's leftmost square. Transitions are of the form (q*i*, w, x, q*j*) where q*i* is the current state, w the string to be read by head 1, x the string to be read by head 2, and q*j* the state to enter. We use the symbol 'λ' to refer to the null string, i.e. "do not read," and '#' as a right-edge marker. An input string is accepted iff one of the heads (or both) falls off the right edge and enters the final state q*f*.[13] Our THFA(K) simultaneously uses its two heads in order to compare the number of *b*s in every two adjacent columns. It accepts a string iff the string has no column whose length is greater than or equal to the length of a column to its left. THFAs for acceptance of actual NC strings will be analogous to THFA(K).

C(onstant) G(rowth). Based on work by Aravind Joshi and by Bob Berwick, Weir (1988, p. 3) presents a definition of CG as follows:[14]

> L is **constant growth** if there is a constant c_0 and a finite set of constants C such that for all $w \in L$ where $|w| > c_0$ there is a $w' \in L$ such that $|w| = |w'| + c$ for some $c \in C$.

He also gives the following theorem: (*ibid.*, pp. 96–97)

> **Theorem 4.3.1** If L is a language generated by a grammar of some formalism that is a LCFRS, then L is a semilinear language.

CG is a consequence of semilinearity. By Weir's Theorem 4.3.1, the set of strings in NC generated by a MCTAG is CG. We show now that the set of strings in NC not generated by a MCTAG is also CG, according to the Joshi–Berwick definition. Such

13 Our definition of a Two-Head Finite Automaton differs slightly from that given in Lewis and Papadimitriou (1981, pp. 306–307) wherein acceptance requires both heads to simultaneously move off the right end of the tape while entering a designated final state.

14 A similar definition is given in Berwick (1984, p. 198) and in Berwick and Weinberg (1984, p. 113).

strings are precisely those that contain an indefinite amount of columns of strung-together number-names. To these, we can always add one more instance of the strung-together element, whether *zhao*, *yi*, *wan*, or what have you, immediately to the left of the leftmost instance of that element, yielding a wellformed Chinese number-name. Thus, the set of strings in NC not generated by a MCTAG can "grow" by a constant of 1. Hence, by the Joshi–Berwick definition of CG, NC is CG.

However, according to Alexis Manaster-Ramer (p.c.), the Joshi–Berwick definition of Constant Growth is flawed. While the Joshi–Berwick definition excludes languages such as $\{a^n \mid n \text{ is prime}\}$ from being CG, it nonetheless includes the language $\{b^*a^n \mid n \text{ is prime}\}$, since strings in the latter can "grow" by any constant due to b^*. Yet, these two languages are rather similar and one would not expect them to differ in terms of CG. To avoid this state of affairs, Manaster-Ramer proposes the following embellished definition of Constant Growth:

> A language L over $\{a_1, \ldots, a_n\}$ is Constant Growth iff there is a set C of n-tuples of constants $\{(c_{11}, \ldots, c_{n1}), \ldots, (c_{1m}, \ldots, c_{nm})\}$, and for any string in L the number of occurrences of the different letters can be increased by the constants of such an n-tuple, respectively. Moreover, for every Constant Growth language there is a constant k such that any string containing at least k occurrences of a letter a_i can be increased by at least one such n-tuple which has a non-zero value for the corresponding c_i.

Thus, if L is infinite and there is, in particular, no upper bound on the number of a_is in its sentences, then there is a way of obtaining longer strings by means of increasing the number of occurrences thereof (possibly, but not necessarily, in conjunction with that of some other letter or letters). According to this definition, $\{b^*a^n \mid n \text{ is prime}\}$ is not CG, since given a string with enough *a*s in it, it no longer suffices to increase the number of *b*s, as is possible under the Joshi–Berwick definition. Likewise, under Manaster-Ramer's definition, Numeric Chinese is not CG: Let c be the greatest of the constants by which the number of *zhao*s may be increased. Let $p = \max(k+1, c+1)$. Now consider the following string z in NC:

$$z = wu\ zhao^p wu\ zhao^{p-1} \ldots wu\ zhao$$

By the embellished definition, if NC is CG then it must have some string with a greater number of *wu*s and of *zhao*s than in z. Now in order to increase the number of *wu*s, one must at least create a new column of *zhao*s whose length must be at least $p+1$. But that would mean an increase by more than c, which is the greatest of the constants. There is, thus, no way of increasing the number of *zhao*s while both remaining in NC and satisfying Manaster-Ramer's definition of CG. Hence, NC is not CG under the embellished definition, but is under the Joshi–Berwick definition.

However, as pointed out by David Weir (p.c.), Manaster-Ramer's definition is also flawed. While it excludes both $\{a^n \mid n \text{ is prime}\}$ and $\{b^*a^n \mid n \text{ is prime}\}$ from being Constant Growth, it nonetheless includes $\{a^*b^*a^n \mid n \text{ is prime}\}$, since in this last language, both the *a*s as well as the *b*s, i.e. all the members of its alphabet, can "grow" by any constant due to a^*b^*. But here again, the three of these languages are rather similar and one would not expect them to differ in terms of CG. All this suggests that perhaps these types of definitions will not lead us to capture the intuitive notion of CG in its entirety. While Manaster-Ramer's definition attempts to be as analogous as possible to some definitions of semilinearity, it still does not model all and only that which CG is intuitively meant to include.[15]

15 An anonymous referee has aptly indicated that the main intuition behind CG is the linear growth of

Limited Cross-Serial Dependencies. This property rests largely on how a grammar handles some particular dependency. Moreover, what is considered 'limited' may be open to more than one interpretation. Joshi *et al.* (1989, p. 3) suggest that the cross-serial dependencies in Dutch subordinate clauses, homomorphic to the single-copy language $\{WW \mid W \in (a + b)^*\}$, could be considered an instance of a limited cross-serial dependency, but not the language MIX, which consists of an equal number of *a*s, *b*s, and *c*s in any linear order. The dependencies in NC are of multiple columns correlated in length. These are considered Mildly CS, as MCTAGs capture them. What MCTAGs fail to do is generate a language with such dependencies, such as NC, that also places no fixed limits on the number of columns in its strings. We ask then whether a dependency that leads to an indefinite number of columns correlated in length is to be considered "limited" or not. Since, as mentioned earlier, what is "limited" may be open to numerous interpretations, we leave this question unanswered for the time being. Thus, a grammar for NC will have the Mildly CS property of limited dependencies, in case a dependency between an indefinite number of columns correlated in length is to be considered limited.

We conclude then that it is not entirely clear whether NC is Mildly CS, although intuitively it appears most likely not to be so. While NC is polynomially recognizable, we can say nothing about polynomial-time parsing for this language, as this would require reference to a *particular* grammar generating the language. It is constant-growth under one known definition, but not so under a somewhat better embellished one. Last, we cannot determine whether its dependencies of indefinite columns of correlated length are limited or not, lacking a straightforward and lucid definition for "limited." However, in contrast to what we can conclude regarding NC and the vague notion of mild context-sensitivity, we do at least know for certain that NC is not a MCTAL.

6. The Linguistic Relevance of Our Formal Results

In the previous sections, we have shown that Numeric Chinese can be generated neither by a single- nor a multiple-component TAG. What are the implications of this for its proper superset Chinese, a natural language? These are not immediately obvious. After discussing Zwicky's (1963) argument, which, albeit weaker, resembles our own due to the similarity between his language P and our language K, in a study which surveyed and refuted earlier arguments calling for the non–context-freeness of NL, Pullum and Gazdar (1982, p. 502) mention:

> The interest of this argument [i.e., Zwicky (1963)] in the context of the study of natural languages is, however, greatly lessened by the fact that it deals with the internal structure of elements of a representational system for mathematics. We would maintain that knowledge of how to construct such number names (which, of course, has to be explicitly taught to children who speak English perfectly well) is knowledge of mathematics rather than of language.

However, simply "maintaining," with no further argumentation, that we are dealing here with "knowledge of mathematics rather than of language," is far from clear. More evidence is needed in order to sustain this claim. We intend to show below that, in any case, maintaining this position resolves little.

'structures' (and not necessarily 'strings') and, therefore, any redefinition of this property would have to reflect the structure-based intuition.

Our formal argumentation, as well as that of Zwicky's, relies crucially on characteristics (i) and (ii) discussed in Section 2. These characteristics have been shown in Section 3 to hold for the Chinese number-name system. The issue now is whether they hold because of some linguistic constraint or because of some other constraint, perhaps mathematical in nature.

Let us assume, for the sake of argument, that these characteristics, particularly (ii), are mathematical in nature. In other words, the reason that larger clusters of *zhao* precede smaller ones is due to some constraint within the mathematical component of the human cognitive faculty. This would make little sense, however, since addition is commutative. In other words, in purely arithmetical terms *wu zhao zhao wu zhao* is equivalent in value to *wu zhao wu zhao zhao*. But while the former is empirically acceptable in the language, the latter is not. Thus, the linear order constraint that larger clusters of *zhao* must precede smaller ones cannot be based on pure mathematics.

If the nature of the constraint is mathematical in any way, it must be one that relates to the interface between the cognitive linguistic component and the cognitive mathematical one.[16] Since, for purposes of cognitive numerical computing, number-names must ultimately be translated somehow into some encoding of their particular arithmetic values, an NL-math interface is necessary. Let us assume that the linear order constraint applies in the interface. Two possibilities then exist: the constraint is either string-based or value-based. We explore these below.

String-based. What we mean by this is the following: The syntax of some NL, in our case Chinese, generates cardinal number-names. These are passed to the cognitive mathematical component for further computation, via the NL-math interface. In this interface, a test is performed on strings of type (i) to see that they adhere to characteristic (ii). Yet, given our result from Section 5, an automaton with the power of, or weaker than, a MCTAG will not be able to perform such a test, since it cannot discriminate between these strings and those of type (i) that do not adhere to characteristic (ii) (hence cannot accept J). Our results, then, are directly translatable to apply not to the "pure" syntax of Chinese, but rather to the interface between Chinese and mathematics. We will have said then something substantially interesting about the interface between NL and mathematics, namely, that this interface must have tools that are weakly more powerful than a MCTAG.

Value-based. Here again, the syntax generates cardinal number-names that are then passed on to the cognitive math component via the NL-math interface. However, in this case the interface performs a test not on the basis of substring length, but rather on the basis of the numeric content of each column. In other words, when encountered with a strung-together number-name, the interface calculates, say right-to-left, the numeric value of each column and compares this value with the numeric value of its adjacent left column. The numeric value of columns must increase from right to left. However, in order to calculate the numeric value of a column, the interface must have the capacity to compute expressions such as a trillion raised to the power of n and save the result for further comparison. If a machine performs this task, as most Turing-like machines do, by means of strings whose length is in direct proportion to the numeric value they encode, then this case is essentially equivalent to a string-based test, as discussed earlier. Yet, if the length of the encodings is not proportional to their value,

16 In this paper, we do not use "interface" to refer necessarily to some 'psychologically (or neurologically) real' processing model, but rather to a model emulating cognitive competence (in the same way that the syntax-semantics interface and syntax-phonology interface are understood).

then we cannot conclude that a machine with more power than a MCTAG must be necessary for the task. Perhaps transducers weaker than, or incomparable with, MCTAGs would do. Yet, are we to expect the interface to be capable of doing computations like these? If it can compute x^y then it can, most likely, deal with many sophisticated computations. But if so, then how does it differ in any substantial way from the "pure" cognitive mathematical component (assumed to be Turing-equivalent)? In other words, why would the interface need the power to perform multiple multiplications if the math component does this anyway? We would expect the interface to be far weaker, if possible. It is indeed possible for it to be weaker, if we assume the test is string-based. But then, our result regarding non-MCTALs holds again.

Thus, if the linear order constraint applies in the NL-math interface, then this interface will either have to be more powerful than a MCTAG in terms of string recognition, or powerful enough to perform calculations that are complex in terms of computing capacity. Given the lack of parsimony in having an NL-math interface that can perform the same tasks that the cognitive mathematical component performs, we conclude that, if the test applies in the interface, this test must be string-based, forcing the interface to be more powerful than, or at least incomparable with, a MCTAG.

Another possibility exists: that the constraint is purely syntactic in nature. If so, we encounter the problem created by what Manaster-Ramer (1988, p. 102) refers to as Ziff's Law: while, for instance, *wu zhao wu zhao zhao* is not a well-formed number-name, nothing blocks it from being some other well-formed proper name, such as the title of a book, for example. Thus the grammar of Chinese can generate a string such as *wu zhao wu zhao zhao*, albeit not as a number-name but still as some acceptable proper name that can function as a noun-phrase in a sentence. If so, the intersection of R or R′ and Chinese is not H or J, respectively, but rather R or R′, respectively. Since L and K are not homomorphisms of R and R′, respectively, an intersection-based proof would fail to say much about the weak generative capacity of the whole of Chinese. That is, in the weak sense, we have not shown that Chinese is beyond the generative power of a TAG or a MCTAG, although we have shown this to be true for its proper subset Numeric Chinese.

The consequences of Ziff's Law are avoided, however, by invoking considerations of *classificatory capacity*. This notion was introduced in Manaster-Ramer (1987a, p. 238) and preliminarily defined there as "the measure of a formalism's ability [to] classify a set of strings (and substrings) and specify which ones are like which other ones." In Radzinski (1990a, p. 122 and 1990b, pp. 85–86), we have further refined this definition to apply to CFGs:

> Let some nonterminal in the grammar ultimately be rewritten only as a string belonging to some particular construction. For example, if passive sentences are to be considered a construction in some NL $\mathcal{L}$, then let the CFG G($\mathcal{L}$) include a member PASS in its set of nonterminal symbols which yields in one or more steps all and only passive sentences of $\mathcal{L}$. We say then that G($\mathcal{L}$) *classifies* passives.

According to the definitions of Linear Context-Free Rewriting Systems given in Weir (1988), (including CFGs, Tree Adjoining Grammars, Head Grammars, Linear Indexed Grammars, Combinatory Categorial Grammars, and Multiple Component TAGs) any LCFRS contains a set of nonterminals. We can thus extend our definition to apply to all LCFRSs, in addition to CFGs.[17] What we would wish to test then is whether any

17 Also, as mentioned in Radzinski (1990a, p. 122 and 1990b, p. 94), there no doubt exist other means for defining classification within CFGs. This point applies to all other LCFRSs as well. Rather than using a

LCFRS for Chinese can classify NC, whose strings clearly constitute a construction in Chinese separate from other noun-phrases. Such a test would fail, since, given our formal results, an arbitrary MCTAG, or other LCFRS, G(Chinese) could not include a nonterminal yielding all Chinese number-names in one or more steps. Hence, if the linear order constraint is purely syntactic in nature, then Chinese is a non-MCTAL on grounds of classificatory capacity.

No matter which position one takes with respect to where the linear order constraint applies, interesting conclusions follow: either Chinese is not a Multiple Component Tree Adjoining Language by classificatory capacity considerations or the interface-language between Chinese and cognitive math, something we know little of, is not an MCTAL when viewed as a formal language. In his "Topic ... Comment" column, Pullum (1986, p. 410) once wondered:

> ... neither the mathematics nor the facts [discussed in Zwicky (1963)] are in dispute, yet the issue still seems hard to resolve. ... We have to be taught [the constraint] in math classes at school, and we do not acquire it with our language *per se*. On odd dates I still think this is right, but on even dates I think the argument has been unjustly overlooked. Zwicky thinks we [i.e. Pullum and Gazdar] were correct to dismiss it, but maybe he is wrong and it was the first valid argument that English is non-CF. The problem here is that we are not entirely sure what is a fact about a language and what is a fact about the culture associated with it.

Yet it is hard to believe that the constraint is explicitly taught in math classes at school, since children rarely deal with numbers with such values. Also, there is no reason to assume that *wan wan* is taught in school, given its natural and common use by Chinese native speakers, regardless of their educational level.[18] Moreover, the phenomenon is completely irrelevant to the culture associated with Chinese or with any other natural language exhibiting a similar behavior. It is a constraint related either to the "pure" syntax of Chinese or to the component interfacing between this syntax and the cognitive faculty for arithmetic reasoning. Both of these are hardly culture-based.[19]

We end this section by presenting, for purposes of contrast, two mathematics-based formal arguments that are quite irrelevant to the study of natural language, although their proponents may have laid claims, or at least hinted, otherwise. The first of these is an argument against the context-freeness of English presented in Elster (1978, pp. 42–44) and refuted by Pullum and Gazdar (1982, pp. 479–481). Elster bases

single nonterminal as a basis, classification can be achieved via a set of nonterminals or via some production rule that applies in all and only the derivations of a particular construction. These definitions, however, would be merely similar alternative formalizations of the same intuitive notion.

18 An anonymous referee has indicated that while this may be true for *wan wan*, forms such as *zhao* are surely taught (originally) in school. Yet, we believe that even if the single-word number-name *zhao* is learned explicitly only via formal education, its use within a strung-together number-name and the linear order pattern exhibited by such a long number-name is not. Rather, the constraints for forming strung-together number-names using *zhao* are the same as those for forming long number-names using *wan*. As mentioned in Section 3, the use of *zhao* instead of *wan* in long number-names results in better acceptability, as the former appears to be the highest valued number denoted by a single-word in modern Chinese.

19 It may very well be the case that the constraint could be shown to be a consequence of some other independent cognitive strategy, such as a version of Hurford's (1975) *Packing Strategy*, for example. This strategy for number-names suggests roughly that arithmetically higher-valued chunks belong higher up in the tree than do the lower-valued ones, analogous to the way one packs a suitcase of books: first the big fat heavy ones and then the lighter ones. However, formal claims (with mathematical rigor) are made precisely in order to get a better understanding and give a better description of what strategies are meant to account for informally.

his argument on sentences like the following:

B_1: The first two million numbers in the decimal expansion of π are $a_1a_2 \ldots a_{2000000}$.

B_2: The first two million million numbers in the decimal expansion of π are $a_1a_2 \ldots a_{2000000000000}$.[20]

$\vdots$

B_k: The first two (million)^k numbers in the decimal expansion of π are $a_1a_2 \ldots a_{2 \cdot 10^{6k}}$.

$\vdots$

According to the pumping lemma for CFLs, any sufficiently long sentence of a CFL can be extended by indefinite repetition of, at most, two subparts without violation of grammaticality. The only possible pumpable substrings in sentences like B_k are within the cluster of *millions* and $a_1 \ldots a_n$. If we pump the substrings 'million^q' and '$a_r \ldots a_t$' up once, then we end up with a sentence like *C*:

C: The [first] two million^{k+q}...numbers in the decimal expansion of p are $a_1 \ldots a_t a_r \ldots a_t \ldots a_{2 \cdot 10^{6k}}$

Elster claims (p. 44) that *C* is not a grammatical sentence in English because:

> ...the number 'two million^{k+q}' must be the same as the number '$2.10^{6k} + t - r + 1$', i.e. the same as the number of numbers in the decimal expansion. Note that this is a requirement not of mathematics, but of linguistics, just as the lack of grammaticality of the sentence,
>
> *D*: the two largest animals in the zoo are a mouse,
>
> is a matter of linguistics, and not of mathematics ...

But, according to Elster, in order for *C* to be "grammatical," the length of its decimal expansion must be longer: $2 \cdot 10^{6(k+q)}$. Thus, since there is a sufficiently long sentence in English that lacks, at most, two pumpable substrings, English is not CF.

As Pullum and Gazdar (1982, pp. 480–481) claim, Elster is wrong since he is assuming that English grammar requires, as seen from the ungrammaticality of *D* versus the grammaticality of the *B* sentences above, that the number of entities listed in a predicate correspond to the number named in the subject. Yet if *D* is ungrammatical, it is not because of such a requirement, but rather because of the number (i.e., singular/plural) disagreement it exhibits. If Elster were right, then the sentence "The two largest animals in the zoo are Mickey, Minnie, and Donald" would be ungrammatical. Yet this sentence is clearly grammatical, albeit infelicitous. Elster is confusing

20 The reproduction of this sentence in Pullum and Gazdar (1982, p. 479) lacks one instance of *million*. Thus, Elster's argument becomes largely incomprehensible if read only there. The version reproduced in Savitch *et al.* (1987, p. 149) contains the same typographical error. Also, notice that Elster's acceptance of sentences of this sort with strung-together *millions*, indicates that he too is of the opinion that strung-together number-names are empirically attested in English.

grammaticality with arithmetical felicity. His argument, thus, bears little on natural language.

The second argument, this time not discussed by Pullum and Gazdar (1982), is one presented in Brainerd (1966b, pp. 119–124). Brainerd discusses "verbal expressions," i.e. number-names, for rational numbers of the following type:

zero point three six five four six five three six five three etc.

three point two seven seven two seven seven two seven seven etc.

These number-names for repeated decimals contain subparts of indefinite length that are repeated *ad infinitum*. However, given the periodicity of the repetition, the number, and hence its name, may be expressed by a convention requiring only a fixed amount of repetitions. A convention using merely one repetition suffices to argue that the set comprising the names for repeated decimals constitutes a non-CFL (two repetitions: a non-TAL). As Brainerd claims on p. 122:

> ...it is easy to show that the language L(R) [comprising the numeral names for non-negative rational numbers in English] possesses no context-free grammar.

Indeed, such a language can be shown to be a non-CFL, via the strong pumping lemma for Context-Free Languages. Brainerd later writes on p. 124:

> It is perhaps not out of place to observe that if repeated decimals (in their English verbal form) are a part of the natural language, then repeated decimals constitute an example of a duplication-structure of arbitrary length in English.

Thus, Brainerd suggests, albeit equivocally, that English, a natural language, is affected by the non–context-freeness of its sub-language consisting of the names for rational numbers. Yet, this could hardly have any basis whatsoever. On grounds of weak generative capacity, claiming that English is non-CF because it has "a duplication-structure of arbitrary length," is as absurd as claiming that the regular language $\{a^*b^*c^*\}$ is a non-CFL because it is a proper superset of the non-CFL $\{a^nb^nc^n\}$. Although pumping a string in $L(R)$, longer than some constant, in accordance with the conditions set forth by the strong pumping lemma for CFLs, yields a string outside of $L(R)$, it nevertheless yields one that is a well-formed number-name, albeit corresponding perhaps to an irrational number. However, considerations of classificatory capacity will not help here, since there is absolutely no linguistically based reason to assume that the set of names for rational numbers constitutes a construction distinct from names for other types of numbers. Thus, as with Elster's argument, Brainerd's has little bearing on natural language. Contrary to these two arguments, our claims, as we have already seen and discussed, *definitely* do bear either on the syntax of a natural language or on the interface between that natural language and the cognitive mathematical faculty.

7. Conclusions

We have shown that, when viewed as a formal language, the number-name system of Chinese is neither a single- nor a multiple-component Tree Adjoining Language, due to its strung-together number-names of indefinite length. As a consequence, it cannot be generated by any Linear Context-Free Rewriting System, including Context-Free Grammars, Head Grammars, Linear Indexed Grammars, or Combinatory Categorial Grammars. It appears also not to be Mildly Context-Sensitive at all, notwithstanding its

recognition in linear time. Our formal results relate directly either to the syntax proper of Chinese or to the interface between that natural language, the most widely natively spoken one, and the mathematical component of the human cognitive endowment. Similar results hold, most likely, for other natural languages, perhaps even for English. Consequently, it may be the case that Zwicky (1963) did expound after all, at least in spirit if not in letter, the first valid argument that English is non–context-free. On a practical side, then, our study may help Geoff Pullum untangle himself from his odd/even dates dilemma. On a more theoretical side, we have discovered something quite interesting about the nature of human language and its relationship to numeral systems.

Acknowledgments

The author wishes to thank James Allen, David Blanc, Lindsey Eck, Joyce Friedman, Gerald Gazdar, David Gil, Doron Gill, Jim Hurford, Susumu Kuno, Tsippy Lotan, Alexis Manaster-Ramer, John O'Neil, Maya Radzinski, Walt Savitch, Mark Steedman, K. Vijayashanker, David Weir, Arnold Zwicky, and anonymous referees for their many comments and suggestions; Fu Tan and Yuli Zhou for supplying native judgments of Chinese; Baruch Tercatin for supplying native judgments of Rumanian; and Raffaella Zanuttini for her assistance in the search of valuable sources of information essential for this study. Responsibility for errors lies solely with the author. This paper is a modified version of Chapter 4 of Radzinski (1990b).

References

Aho, A. (1969). "Nested Stack Automata," *Journal of the ACM* 16, 383–406.

Aho, A. (1968). "Indexed Grammars — An Extension of Context-Free Grammars," *Journal of the ACM* 15, 647–671.

Ajdukiewicz, K. (1935). "Die syntaktische Konnexität," *Studia Philosophica* 1, 1–27.

Bar-Hillel, Y. (1953). "A Quasi-Arithmetical Notation for Syntactic Description," *Language* 29, 47–58 [reprinted in Y. Bar-Hillel (1964). *Language and Information: Selected Essays on Their Theory and Application*, 61–74, Addison-Wesley, Reading, MA.]

Berwick, R. (1984). "Strong Generative Capacity, Weak Generative Capacity, and Modern Linguistic Theories," *Computational Linguistics* 10, 189–202.

Berwick, R.; and Weinberg, A. (1984). *The Grammatical Basis of Linguistic Performance: Language Use and Acquisition*, The MIT Press, Cambridge, MA.

Brainerd, B. (1971). *Introduction to the Mathematics of Language Study*, Elsevier, New York.

Brainerd, B. (1967). "A Transformational-Generative Grammar for Rumanian Numerical Expressions," *Cahiers de Linguistique Theorique et Appliquee* 4, 35–45 [reprinted in Brandt Corstius (1968), 41–52].

Brainerd, B. (1966a). "Two Grammars for Chinese Number Names," *Canadian Journal of Linguistics* 12, 33–51.

Brainerd, B. (1966b). "Grammars for Number Names," *Foundations of Language* 2, 109–133 [revised version appears as "On the Syntax of Certain Classes of Numerical Expressions," in Brandt Corstius (1968), 9–40].

Brainerd, B., and Peng, F. (1968). "A Syntactic Comparison of Chinese and Japanese Numerical Expressions," in Brandt Corstius (1968), 53–81.

Brandt Corstius, H., ed. (1968). *Grammars for Number Names (Foundations of Language* Supplementary Series Volume 7), Reidel, Dordrecht.

Chao, Y. (1968). *A Grammar of Spoken Chinese*, University of California Press, Berkeley, CA.

Culy, C. (1985). "The Complexity of the Vocabulary of Bambara," *Linguistics and Philosophy* 8, 345–351 [Reprinted in Savitch *et al.* (1987), 349–357].

Elster, J. (1978). *Logic and Society: Contradictions and Possible Worlds*, John Wiley and Sons, New York.

Epstein, S. (1978). "Review of Hurford (1975)," *Journal of Linguistics* 14, 123–124.

Fischer, M. (1968). *Grammars with Macro-like Productions*, Doctoral Dissertation, Harvard University [appeared same year in *Mathematical Linguistics and Automatic Translation*, Harvard University Computation Laboratory Report NSF-22].

Gazdar, G. (1988). "Applicability of Indexed Grammars to Natural Languages," In U. Reyle and C. Rohrer, eds.: *Natural Language Parsing and Linguistic Theories*, 69–94, Reidel, Dordrecht.

Greenberg, J. (1978). "Generalizations about Numeral Systems," in J. Greenberg, ed.: *Universals of Human Language: Volume 3 — Word Structure*, 249–295, Stanford University Press, Stanford, CA.

Henne, H.; Rongen, O.; and Hansen, L. (1977). *A Handbook on Chinese Language Structure*, Universitetsforlaget, Oslo.

Hopcroft, J., and Ullman, J. (1979). *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Reading, MA.

Hurford, J. (1979). "Numerals and the Homogeneity of Description and Explanation," *Lingua* 48, 35–42.

Hurford, J. (1975). *The Linguistic Theory of Numerals*, Cambridge University Press, Cambridge, England.

Huybregts, R. (1984). "The Weak Inadequacy of Context-Free Phrase Structure Grammars," in G. de Haan, M. Trommelen, and W. Zonneveld, eds.: *Van Periferie naar Kern*, 81–99, Foris, Dordrecht.

Joshi, A. (1987). "An Introduction to Tree Adjoining Grammars," in Manaster-Ramer (1987b), 87–114.

Joshi, A. (1985). "Tree Adjoining Grammars: How Much Context-Sensitivity Is Required to Provide Reasonable Structural Descriptions?", in D. Dowty, L. Karttunen, and A. Zwicky, eds.: *Natural Language Parsing: Psychological, Computational and Theoretical Perspectives*, 206–250, Cambridge University Press, Cambridge, England.

Joshi, A.; Vijayashanker, K.; and Weir, D. (1989). "The Convergence of Mildly Context-Sensitive Grammar Formalisms," University of Pennsylvania Technical Report No. MS-CIS-89-14 LINC LAB 144 [to appear in S. Shieber and T. Wasow (eds.): *The Processing of Linguistic Structure*, The MIT Press, Cambridge, MA].

Kac, M. (1987). "Surface Transitivity, *Respectively* Coordination and Context-Freeness," *Natural Language and Linguistic Theory* 5, 441–452.

Kasami, T.; Seki, H.; and Fujii, M. (1988). "Generalized Context-Free Grammars, Multiple Context-Free Grammars and Head Grammars," Osaka University Technical Report, Toyonaka, Osaka.

Lesniewski, St. (1929). "Grundzüge eines neuen Systems der Grundlagen der Mathematik," *Fundamenta Mathematicae* 14, 1–81.

Lewis, H., and Papadimitriou, C. (1981). *Elements of the Theory of Computation*, Prentice-Hall, Englewood Cliffs, NJ.

Manaster-Ramer, A. (1988). "Review of Savitch *et al.* (1987)," *Computational Linguistics* 14, 98–103.

Manaster-Ramer, A. (1987a). "Dutch as a Formal Language," *Linguistics and Philosophy* 10, 221–246.

Manaster-Ramer, A., ed. (1987b). *Mathematics of Language*, John Benjamins, Amsterdam.

Menninger, K. (1969). *Number Words and Number Symbols: A Cultural History of Numbers*, The MIT Press, Cambridge, MA.

Merrifield, W. (1968). "Number Names in Four Languages of Mexico," in Brandt Corstius (1968), 91–102.

Needham, J. (with the collaboration of Wang Ling) (1959). *Science and Civilisation in China: Vol. 3 — Mathematics and the Sciences of the Heavens and the Earth*, Cambridge University Press, Cambridge, England.

Pollard, C. (1984). *Generalized Phrase Structure Grammars, Head Grammars, and Natural Language*, Doctoral Dissertation, Stanford University [to be published by Cambridge University Press, Cambridge, England].

Pullum, G. (1986). "Footloose and Context-Free," *Natural Language and Linguistic Theory* 4, 409–414.

Pullum, G., and Gazdar, G. (1982). "Natural Languages and Context-Free Languages," *Linguistics and Philosophy* 4, 471–504 [reprinted in Savitch *et al.* (1987), 138–182].

Radzinski, D. (1990a). "Unbounded Syntactic Copying in Mandarin Chinese," *Linguistics and Philosophy* 13, 113–127.

Radzinski, D. (1990b). *Mathematics of Unbounded Duplicative and Columnar Constructions in Chinese*, Doctoral Dissertation, Harvard University, Cambridge, MA.

Roach, K. (1987). "Formal Properties of Head Grammars," in Manaster-Ramer (1987b), 293–347.

Savitch, W. (1989). "A Formal Model for Context-Free Languages Augmented with Reduplication," *Computational Linguistics* 15, 250–261.

Savitch, W.; Bach, E.; Marsh, W.; and Safran-Naveh, G., eds. (1987). *The Formal Complexity of Natural Language*, Reidel, Dordrecht.

Shieber, S. (1985). "Evidence against the Context-Freeness of Natural Language," *Linguistics and Philosophy* 8, 333–343 [reprinted in Savitch *et al.* (1987), 320–334].

Steedman, M. (1987). "Combinatory

Grammars and Parasitic Gaps," *Natural Language and Linguistic Theory* 5, 403–439.
Steedman, M. (1985). "Dependency and Coordination in the Grammar of Dutch and English," *Language* 61, 523–568.
Vijayashanker, K. (1988). *A Study of Tree Adjoining Grammars*, Doctoral Dissertation, University of Pennsylvania, Technical Report No. MS-CIS-88-03 LINC LAB 95.
Vijayashanker, K.; Weir, D.; and Joshi, A. (1987). "Characterizing Structural Descriptions Produced by Various Grammatical Formalisms," *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, 104–111.
Weir, D. (1988). *Characterizing Mildly Context-Sensitive Grammar Formalisms*, Doctoral Dissertation, University of Pennsylvania, Technical Report No. MS-CIS-88-74 LINC LAB 132.
Weir, D. (1987). "From Context-Free Grammars to Tree Adjoining Grammars and Beyond," Dissertation Proposal, University of Pennsylvania, Technical Report No. MS-CIS-87-42 LINC LAB 65.
Weir, D., and Joshi, A. (1988). "Combinatory Categorial Grammars: Generative Power and Relationship to Linear Context-Free Rewriting Systems," *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, 278–285.
Zwicky, A. (1963). "Some Languages That Are Not Context-Free," *Quarterly Progress Report of the Research Laboratory of Electronics*, MIT 70, 290–293.