ATOMIZATION IN GRAMMAR SHARING

Megumi Kameyama
Microelectronics and Computer Technology Cooperation (MCC)
3500 West Balcones Center Drive, Austin, Texas 78759
megumi@mcc.com

ABSTRACT

We describe a prototype SHARED GRAMMAR for the syntax of simple nominal expressions in Arabic, English, French, German, and Japanese implemented at MCC. In this grammar, a complex inheritance lattice of shared grammatical templates provides parts that each language can put together to form language-specific grammatical templates. We conclude that grammar sharing is not only possible but also desirable. It forces us to reveal cross-linguistically invariant grammatical primitives that may otherwise remain conflated with other primitives if we deal only with a single language or language type. We call this the process of GRAMMATICAL ATOMIZATION. The specific implementation reported here uses categorial unification grammar. The topics include the mono-level nominal category N, the functional distinction between ARGUMENT and NON-ARGUMENT of nominals, grammatical agreement, and word order types.

Is grammar sharing possible?

The multilingual project of MCC attempts to build a grammatical system hierarchically shared by multiple languages (Slocum & Justus 1985). GRAMMAR SHARING as proposed should have an advantage over a system with separate grammars for different languages: It should reduce the size of a multilingual rule base, and facilitate the addition of new languages. Before presenting evidence for such advantages, however, there is the basic question to be answered: Is grammar sharing at all possible? Although it is well known that languages possess similarities based on genetic, typological, or areal grounds, the question remains whether and how these similarities translate into computational techniques.

In this paper, we will describe a prototype shared grammar for simple nominal expressions in Arabic, English, French, German, and Japanese.[1] We conclude that grammar sharing is not only possible but also desirable. It forces us to reveal cross-linguistically invariant grammatical primitives that may otherwise remain conflated with other primitives if we deal only with a single language or language type. We call this the process of GRAMMATICAL ATOMIZATION[2] forced by grammar sharing. Each language or language type is then characterized by particular combinations of such primitives, often providing new insights with which to account for certain linguistic problems. Before we go into more detail, the following is our view of what general components and mechanisms constitute a shared grammatical system.

Basic mechanisms in a shared grammar: The process of building a shared grammar, in our view, requires (i) linguistic description of a set of languages in a common theoretical framework, (ii) a mechanism for EXTRACTING a common grammatical assertion from two or more assertions, and (iii) a mechanism for MERGING grammatical assertions. The linguistic description should define certain string-combination operations (defined on string TYPES) associated with information structures. Then what we do is identify sharable packages of common string-types and information structures among independently motivated language-specific grammatical assertaions. These packages are then put into the shared part of the grammar, and the remaining language-specifics are potential sources for more sharing. This extraction is essential in what we call ATOMIZATION, which is basically "breaking up of grammatical assertions into smaller independent parts" (i.e. decomposition). If we assume that all grammatical assertions are expressed in terms of FEATURE STRUCTURES (Shieber 1986), the atomization process would be defined around the notion of GENERALIZATION (i.e. reverse of UNIFICATION) as follows:

> ***basic atomization:*** Given two feature structures, Xa for category X in language A and Xb for category X in language B, the shared structure Xα for category X is the GENERALIZATION of Xa and Xb (i.e., the most specific feature structure in common with both Xa and Xb). Xα is separated out of either Xa or Xb, and placed into the shared space. Consequently, a partial ordering is established wherein Xα SUBSUMES Xa and Xb, respectively.

There is an underlying assumption that two language-specific definitions of a common grammatical category share something in common no matter how small it is. This means that the linguistic descriptive basis is questionable if the content of Xα above is null. Conversely, if clearly common information structures appear under language-specific definitions of distinct grammatical categories, we may suspect a basis for a new common grammatical category.

Once the shared and language-specific parts are separated out, a mechanism for merging them is necessary for successfully incorporating the shared assertion into the language-specific assertion. UNIFICATION by INHERITANCE is such a merging mechanism that we employ in our system (see below). The shared space is a complex inheritance lattice that provides various predefined grammatical assertions that can be freely merged to create language-specific ones.

---

[1] Preliminary investigations have also been made on Spanish, Russian, and Chinese.

[2] The verb *atomize* means "to separate or be separated into free atoms" (The Collins English Dictionary, 2nd edition, 1986).

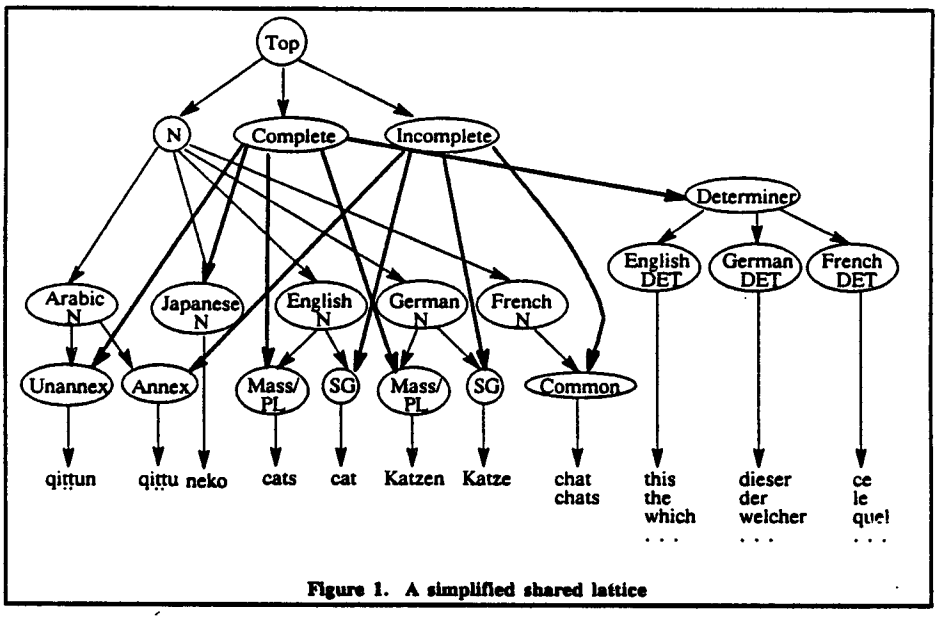


Figure 1. A simplified shared lattice

Shared inheritance lattice: Let us now take a look at a grossly simplified shared inheritance lattice that results from the process described above. See Figure 1. There is a universal notion N(ominal) in all five languages under consideration. This common notion is part of the N definition of each language by inheritance. There are some nominals that are 'complete' in the sense that they can be used as subjects or objects (e.g. *I saw cats/the cat.*). Some others are 'incomplete' in that they cannot be used as such (e.g. *I saw *cat.*). General notions Complete and Incomplete are thereby defined for characterizing relevant nominal classes of each language (see the discussion on ARG vs. NON-ARG below). Since Determiners in English, German, and French make such incomplete nominals complete, the Determiner definition inherits (i.e. includes) the definition of Complete. Lexical items in these languages are defined by multiply inheriting relevant assertions:

In what follows, we will first describe the specific linguistic and computational approaches that we employed to build our first shared grammar. We will then discuss the grammatical primitives for characterizing general nominals, adnominal modifiers, agreement, and word order types, illustrating solutions to specific cross-linguistic problems. We will end with prospects for further work.

Framework

Grammatical framework: We use a categorial unification grammar (CUG) (Wittenburg 1986a; Karttunen 1986; Uzkoreit 1986b). The one described here is a non-directional categorial system (e.g. Montague 1974; Schmerling 1983; van Benthem 1986:Ch.7) with a non-directed functional application rule as the only reduction rule (i.e., a functor X|Y may combine with adjacent Y in either direction to build X). Non-directionality allows for desired flexibility in the shared part of the grammar. A separate component constrains the linear order of elements in each language (see Aristar 1988 for motivation).

Unification and template inheritance: CUG's lexical orientation and unification are employed. In the LEXICON of each language, lexical items are defined to be the unification of language-specific GRAMMATICAL TEMPLATES (Shieber 1984, 1986; Flickenger et al. 1985; Pollard & Sag 1987). These language-specific templates, prefixed with AR(abic), EN(glish), FR(ench), GE(rman), and JA(panese), are feature structures composed by multiple inheritance from shared grammatical templates prefixed with SG (for "Shared Grammar"). SG-templates are themselves composed by multiple inheritance in a complex INHERITANCE LATTICE, whose bottom-end feeds into language-specific templates. The CUG parser (MCC's Astro, Wittenburg 1986b) applies reduction rules to the feature structures of words in the input string.[3] Arabic and Japanese strings are currently represented in Roman letters (augmented for Arabic) with spaces between 'words'.[4]

[3]The parser is linked to an independently developed morphology analyzer (Slocum 1988). This enables each word to undergo a morphological analysis including a dictionary look-up of the root morpheme, and to output a list (or alternative lists) of grammatical template names that, when their contents are unified, produce a single feature structure (or more than one if the word is ambiguous) for that particular token word.

[4]If we were to process Japanese texts directly, the system would have to perform morphological and syntactic analyses simultaneously since there is no explicit word boundaries. (This is one of the strong motivations for our recent movement toward building a new CUG-based morphology system.)

Present linguistic coverage

Simple nominals: The present linguistic coverage is the syntax of SIMPLE NOMINALS: nouns and nominal expressions with lexical or phrasal modifiers such as attributive adjectives (e.g. *long*), demonstratives (e.g. *this*), articles (e.g. *the*), quantifiers (e.g. *all*), numerals (e.g. *three*), genitives (e.g. *of the Sun*), and pp-modifiers (e.g. *in the ocean*). Complex nominals including conjunctions, derived nominals, gerunds, nominal compounds, and relative clause modification have not been handled yet.

Data analysis: We first analyzed a data chart of simple nominals in each language. The chart focused on the syntactic well-formedness of nominal expressions, in particular, the order and dispensability of elements when the nominal expression acts as an argument (e.g. subject, object) to a verb or an adposition (i.e. preposition or postposition).

Shared templates overview

By design, the SG-LATTICE captures shared grammatical features in the given set of languages, whether they are due to universal, typological, genetic, or areal bases. As our research proceeded, we observed an atomization process whereby more and more grammatical properties were distinguished. This was because certain grammatical characterizations that seemed most natural for some language(s) were only partially relevant to others, which forced us to break them down into smaller parts so that other languages can use only the relevant parts.

Modules in the SG-lattice: As the shared templates underwent atomization, we created sublattices corresponding to independent grammatical modules so that a grammar writer can make a language-specific combination of shared templates by consciously selecting one or more from each group. The existing subgroups are: (i) categorial grammar categories (the theory-dependent aspect of the shared grammar), (ii) common syntactic categories (theory-independent linguistic notions), (iii) grammatical agreement (to handle grammatical agreement within nominals), (iv) reference types (semantic features of the nominals, e.g. definite, indefinite, specific), (v) determiner types (to handle co-occurrence and order restrictions among determiners), and (vi) attributive modifier types (to handle order restrictions among attributive modifiers). We will focus on (i)-(iii) in this paper.

Kinds of SG-templates: SG-templates as they exist fall under the following types. The most general distinction can be made between ATOMIC and COMPOSITE templates. Atomic templates inherit from no other template. They result from the atomization process, and are primitive parts that a grammar writer can put together to create more complex templates. A composite template inherits from at least one other, to which a partial structure defined for itself may be added. We may also distinguish between UTILITY and SUBSTANTIVE templates. Utility templates contribute integral parts of categorial grammar categories such as how many arguments they need to combine with—none for a BASIC CATEGORY, and one or more for a FUNCTOR CATEGORY. Substantive templates supply grammatical categories and features expressed in terms of various linguistic notions. Specific examples are discussed below.

Highlights of shared grammatical atoms

The basic graph structure

Each word must be associated with a complete CUG feature structure. The current implementation uses a matrix notation for ACYCLIC DIRECTED GRAPH. See Figure 2:

```
[result: [cat:      [ ]    <- the syntactic type of α
         index:     [ ]    <- relative linear position of α
         agr:       [ ]    <- grammatical agreement features of α
                              (optional)
         feats:     [ ]    <- pragmatic agreement features of α
         type:      [ ]    <- the functional type of α  (see below)
         elements:  [ ]    <- elements within α
         order:     [ ]    <- order of elements  (see below)
 arguments:  [ ]]          <- arguments sought  (see below)
```

Figure2. The notation for a word whose resulting structure is α

A category is either SATURATED (looking for no argument) or UNSATURATED (needing to combine with one or more arguments). It is saturated when the value of ARGUMENTS is 'closed' with symbol #. An unsaturated category may seek one or more arguments, each of which is either unspecified ([ ]) or typed (e.g. [cat: N]). Overall saturation is sought in parsing. The parser assigns index numbers to words in the input string from left to right, and coindexes corresponding substructures under ELEMENTS. The ELEMENTS component currently has A for the word for which this structure is defined, B for the first argument, and C for the second argument. These labels simply flag PATHS for accessing particular elements. There can be any number of order-relevant labels corresponding to an element. These labels, with coindices with respective elements, are in the ORDER component, which is subject to the Word Order Constraint (discussed later). TYPE is the slot for assigning the pseudo-functional category ARG or NON-ARG that we found significant in the present cross-linguistic treatment of nominals (see below). AGR(eement) and FEATS subgraphs contain grammatical and pragmatic agreement features, respectively (discussed later).

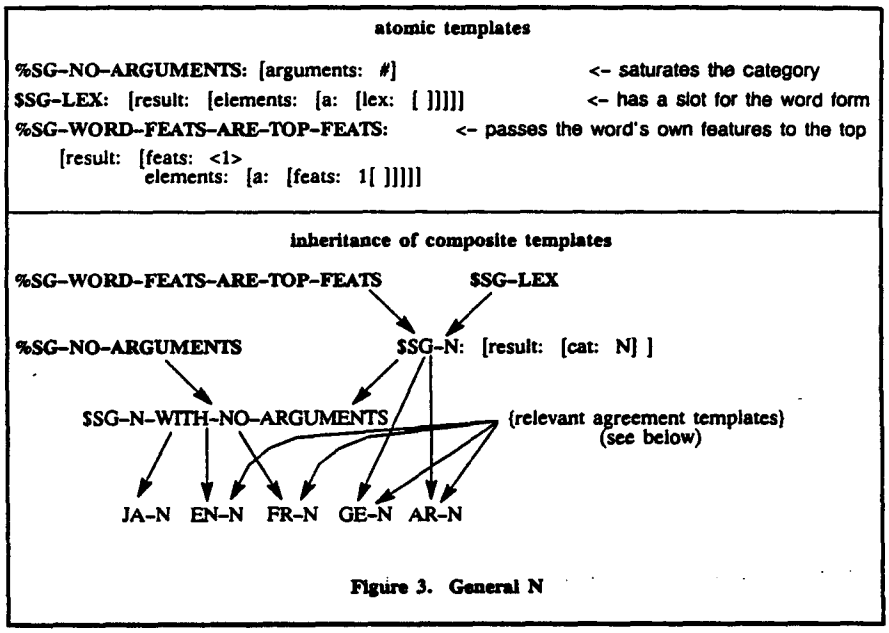


Figure 3. General N

A few more remarks about the notation follow. A value can be either atomic (e.g N), a disjunction of atomic values enclosed in curly brackets (e.g. {N P}), or a complex feature structure. It can also be unspecified ([ ]). The identity of two or more values is forced by reentrant structures indicated by coindexing (e.g. 1[ ] and <1>). Such coreferring value slots automatically point to a single data structure entered through any one of the slots.

Universal mono-level category N

Category N: We posit the universal category N for nominals. Nominals here are those that realize ARGUMENTS such as subjects and objects. Nominals are more commonly labeled NP, a phrase typically built around N or CN (common noun), as in phrase structure NP->DET N as well as in the categorial grammar characterization of DET as a functor NP/CN (i.e. combines with CN and builds NP) (e.g. Ades & Steedman 1982; Wittenburg 1986a). This BI-LEVEL view of nominals is motivated by facts in western European languages. In English, for instance, while *cat* or *white cat* cannot fill a subject position, *a cat* and *this cat* can. In contrast, while *he* can be a subject, it cannot be modified as *this he* or *strange he*. This motivates the following category-assignments with a constraint that only NPs can be arguments: *cat* is CN, *he* is NP, *a* and *this* are NP/CN, and *white* and *strange* are CN/CN. This, however, requires that plurals and mass nouns be CN and NP at the same time since *cats, gold, white cats, white gold, these cats,* and *this gold* can all be arguments. The count/mass distinction is also often blurred since a singular count noun like *cat* may be used as a mass noun referring to the meat of the cat, and a mass noun like *gold* may be used as a singular count noun referring to a UNIT of gold or a KIND of gold (see e.g. Bach 1986). The boundary between NP and CN is at best FUZZY.

When we turn to other languages, the basis for the bi-level view vanishes. In Japanese, for instance, *neko* 'cat' can be an argument on its own, and pronoun *kare* 'he' can be modified as in *ano kare* 'that he' and *okasina kare* 'strange he'. In short, there is no basic syntactic difference among count nouns, pronouns, and mass nouns (and no singular/plural distinction on a 'count' noun). All of them behave like plural and mass nouns in English. This supports a mono-level view of nominals, which we intend to capture with category N. Figure 3 shows the SG-templates relevant to the most general characterization of N in each language. SG-templates in the following illustrations are marked as follows: atomic templates SG-x (boldface), utility templates %SG-x, and substantive templates $SG-x.

At the most general level, the basic nominals in German (GE-N) and Arabic (AR-N) must be unsaturated because genitive-inflected Ns may take arguments. The basic nominals in Japanese (JA-N), English (EN-N), and French (FR-N), on the other hand, are basic categories that are saturated.[5] In addition, all but JA-N inherit relevant AGR(eement) templates (see below). Crucially, note that what looks like a reasonable characterization of N in each language actually consists of a particular selection from the common set of primitives.

ARGUMENT and NON-ARGUMENT: We posit a pseudo-functional level of description in terms of ARG(ument) and NON-ARG for category N instead of the category-level distinction between NP and CN. ARG may function as an argument alone, and NON-ARG cannot.

[5]Note that English possessive marker *'s* is not treated as an inflection here.

NON-ARG becomes ARG only by being combined with a certain modifier or by undergoing a semantic change (e.g massifying). In this view, the ARG/NON-ARG distinction is 'grounded on a complex interaction of morphology, semantics, and syntax.

In English and German, singular count nouns (e.g. *tree*, *Baum*) are NON-ARG while plurals, mass (singular) nouns, proper names, and pronouns are ARG. The NON-ARG nouns become 'complete' ARG nominals either by being modified with determiners or by changing into mass nouns (typically changing an object reference into a property/substance reference, e.g., *I used apple in my pie.*).[6] In French, all forms of common nouns (i.e. singular, plural, and mass) are NON-ARG, in need of determiners to become ARG (e.g., *J'ai vu *arbres/les arbres* 'I saw trees'; **Amour/L'amour est delicat* 'Love is delicate').

In Japanese, there are few NON-ARG nouns (e.g., *kata* 'person' (HONORIFIC)), which can become ARG with any modifier such as a relative clause or an adjective (e.g. *himana kata* 'free person (HON.)'.[7] In Arabic, the morphological distinction of nouns between ANNEXED vs. UNANNEXED corresponds to NON-ARG and ARG statuses, respectively.[8] For instance, the unannexed form *qiṭṭa:ni* CAT-DUAL_NOM-UNANNEX 'two cats' may occur as subject alone whereas the annexed form *qiṭṭa:* CAT-DUAL_NOM cannot. The latter must be modified with a noun-based modifier such as a genitive phrase, and this modifier must be unannexed (e.g. with *rajulin* MAN-GEN-UNANNEX, *qiṭṭa: rajulin* 'man's two cats'). These facts in Japanese and Arabic show that the proposed functional distinction for nominals is motivated independently from the syntactic role of determiners since neither language has modifiers of category DET that we find in English, French, and German (more discussed later).

We realize that the ARG/NON-ARG distinction itself is not a final solution until fine-grained syntactic-semantic interdependence is fleshed out. For now, we simply posit pseudo-functional types ARG and NON-ARG, which are either changed or passed up within the nominal structure:[9]

$SG-ARG: [result: [type: arg]]
$SG-NON-ARG:[result: [type: non-arg]]

Category N|N: Adnominal modifiers (N-MODs) are now universally N|N (i.e. a functor that combines with N and builds N). This includes both determiners and attributive modifiers. **Figure 4** shows the SG-templates for the basic N-MOD. Different kinds of N-MOD must then distinguish whether it takes one or two arguments and whether the resulting nominal with modification is ARG or NON-ARG. Each distinction is briefly illustrated below.

Two kinds of genitive: Genitive N-MOD functors may take different numbers of arguments cross-linguistically. An inflected genitive nominal (e.g. GE: *Marias*, AR: *rajulin* 'man's') takes one, while a genitive adposition (e.g. EN: *of*) takes two. The former is captured with SG-INFLECTIONAL-GENITIVE-CASE-MOD, and the latter, with SG-PARTICLE-GENITIVE-CASE-MOD. See **Figure 5**.

Non-universal determiner category: In the present approach, DET(erminer) is a modifier type (including articles, demonstratives, quantifiers, numerals, and possessives) such that at least one of its members is needed for making an ARG nominal out of a NON-ARG. The fact that a nominal with a determiner is always ARG translates into SG-DET inheriting from SG-ARG among others. DET is present in English, German, and French, but not in Japanese or Arabic (or Russian or Chinese). Demonstratives, quantifiers, numerals, and possessives in the latter languages do not share the syntactic function of DET. We suspect that the presence of DET is an areal property of western European languages.

The sublattice in **Figure 6** highlights two aspects of DET. One is the difference between DET and ADJ(ective) in English, German, and French with respect to the ARG status of the resulting nominal. DET always builds ARG cancelling whatever the type of the incoming nominal whereas ADJ passes the type of the incoming nominal to the top. The other is the place of demonstratives in relation to DET. Every language has demonstratives encoding two or three degrees of speaker proximity (e.g. JAPANESE: *kono* (close to the speaker), *sono* (close to the addressee),

[6]In implementation, this latter process may be triggered by a unary rule COUNT->MASS.

[7]They are assigned a NON-ARG category MN (for 'modified noun') separate from the ARG category N. Any modifier changes it into ARG.

[8]ANNEXED here means 'needing to be annexed to a noun-based modifier', and UNANNEXED means 'completed'. These are also called NONNUNATED and NUNATED forms, respectively, in Semitic linguistics (Aristar, personal communication).

[9]An intriging direction is shown in Krifka's (1987) categorial grammar treatment. He assigns the singular count noun in English (i.e. our NON-ARG) an unsaturated nominal category looking for its numerical value both in syntax and semantics. The significance of determiners is here as suppliers of numerical values. How this approach can be extended to cover the NON-ARG nominals in Arabic and Japanese (which are not in need of numerical values per se) remains to be seen. Although it makes sense to see NON-ARG as a functor looking for more semantic determination, implementing it would require a reduction rule for TWO FUNCTORS LOOKING FOR EACH OTHER. The current system would cause an infinite regression with such a rule.

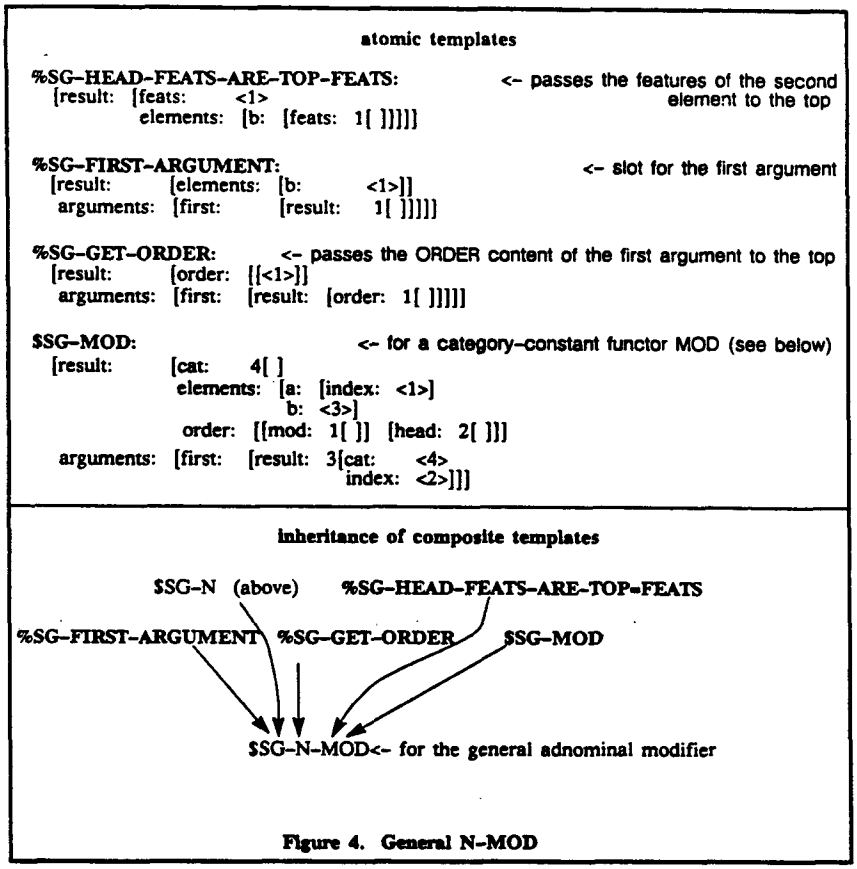


Figure 4. General N-MOD

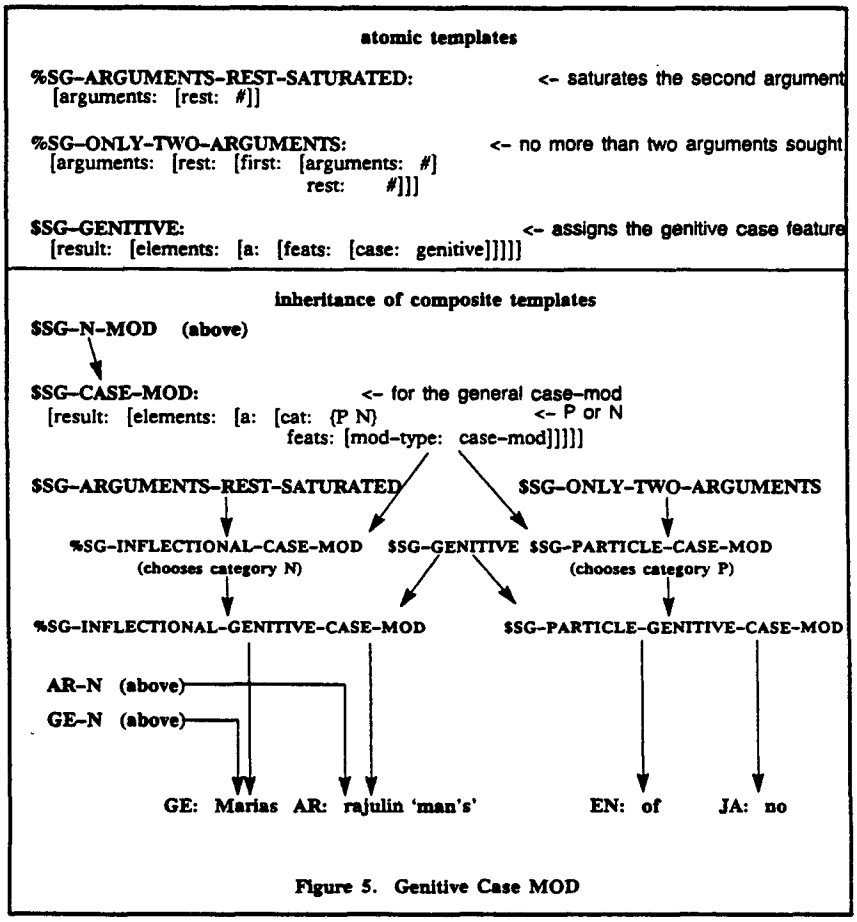


Figure 5. Genitive Case MOD

and *ano* (away from either)), but they belong to the class of determiners only if the language has DET.

Grammatical agreement (AGR)

Two kinds of features are distinguished, linguistic features relevant to GRAMMATICAL AGREEMENT (e.g. French grammatical gender *une/*un table* 'a table' *f.*), and referent features relevant to PRAGMATIC AGREEMENT (e.g. using *she* to refer to a female person; using appropriate numeral classifiers for counting objects in Japanese). The former is under attribute AGR, and the latter is under FEATS. The N-internal grammatical agreement (AGR) requires that certain features of the HEAD Nominal must agree with those of MOD. For instance, English has number agreement (e.g. *this book, *those book, *this books*). Among the five languages under consideration, all but Japanese have AGR.

Although there is cross-linguistic variation in AGR features, it is not random (Moravcsik 1978). Table 1 sums up the N-internal AGR features in the four languages. All AGR features go under attribute AGR so that its presence simply corresponds to the presence of grammatical agreement in a language. EN-N, for instance, inherits the shared template for number agreement, and FR-N inherits those for number and gender agreements. See below:

```
$SG-NBR-AGR:
[result: [agr: [nbr: <1>]
      elements: [a: [feats: [nbr: 1[]]]]]]
$SG-GDR-AGR:
[result: [agr: [gdr: <1>]
      elements: [a: [feats: [gdr: 1[]]]]]]
```

Separating AGR and FEATS enables us to create SG-templates that impose the most general agreement constraint regardless of the precise content of agreement features. Three agreement templates produce the combined effect of N-internal agreement constraint, SG-AGR, SG-AGR-ARGUMENTS, and the composite of the two, SG-AGR-WITH-ARGUMENTS. See Figure 7.

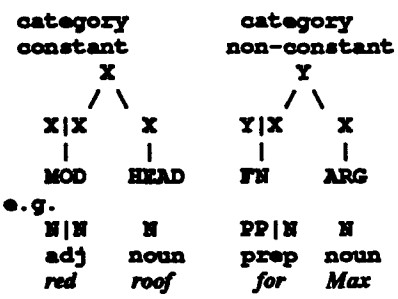
The reentrancies impose the strict identity of AGR features: (i) $SG-AGR—between the topmost structure and the element that the graph is defined for, (ii) $SG-AGR-ARGUMENTS—between the topmost structure and the first argument, and (iii) $SG-AGR-WITH-ARGUMENTS—among all the three. (i) goes into ALL NOMINALS, passing the nominal's AGR features to the top level. This is because the AGR features must always be available at the top level of a nominal so that they can be used when the nominal is further modified. (ii) goes into ALL ADNOMINAL MODIFIERS, passing the head nominal's AGR features to the top level. (iii) goes into ONLY THOSE ADNOMINAL MODIFIERS SUBJECT TO THE AGR CONSTRAINT, for instance, demonstratives (e.g. *these*) but not attributive adjectives (e.g. *small*) in English, and both demonstratives and adjectives in French (see this difference in the above inheritance).

This is an example where a better language-specific treatment is obtained from the grammar-sharing perspective. If only English is handled, one may simply force the identity of NBR features amidst all kinds of other features, but in the light of cross-linguistic variation and invariants, it lends itself naturally to separating out two kinds of features that correspond to different semantic interpretation processes.

Category constancy and word order typology

In connecting word order typology and categorial grammar, we have benefited from work of Greenberg (1966), Lehmann (1973), Vennemann (1974, 1976, 1981), Keenan (1979), Flynn (1982), and Hawkins (1984). Among these, we have a first-cut implementation of Vennemann's (1981) and Flynn's (1982) view that the functor types based on CATEGORY CONSTANCY have a significant relation to the default word order of a language. A functor is CATEGORY-CONSTANT if it builds the same category as its argument(s). It is CATEGORY-NON-CONSTANT if it builds a different category from its argument(s). These notions are also called ENDOTYPIC and EXOTYPIC, respectively, by Bar-Hillel (1953), and are crucially used in Flynn's high-level word order convention statements. The definitions of the notions MOD (modifier), HEAD (head), FN (function), and ARG (argument) follow:

- MOD is a category-constant functor (X|X) that combines with HEAD (X). (see above for SG-MOD)
- FN is a category-non-constant functor (Y|X) that combines with ARG (X).



There is cross-linguistic evidence that MOD-HEAD and FN-ARG orders tend to go in opposite directions. This amounts to two basic word order types in languages:

```
ORDER TYPE 1: ARG < FN
              MOD < HEAD
ORDER TYPE 2: FN < ARG
              HEAD < MOD
(where < reads as 'precedes')
```

The N-level default word order in a language is determined as follows: Every language has ADPOSITIONS (prepositions and postpositions), universally a category-non-constant functor PP|N. A postpositional language (i.e. a language that uses only or predominantly postpositions) then belongs to TYPE 1 (ARG < FN), and a prepositional language belongs to TYPE 2 (FN < ARG). In the present case, EN, GE, FR, and AR are prepositional while JA is postpositional.

The default MOD order is most faithfully observed in

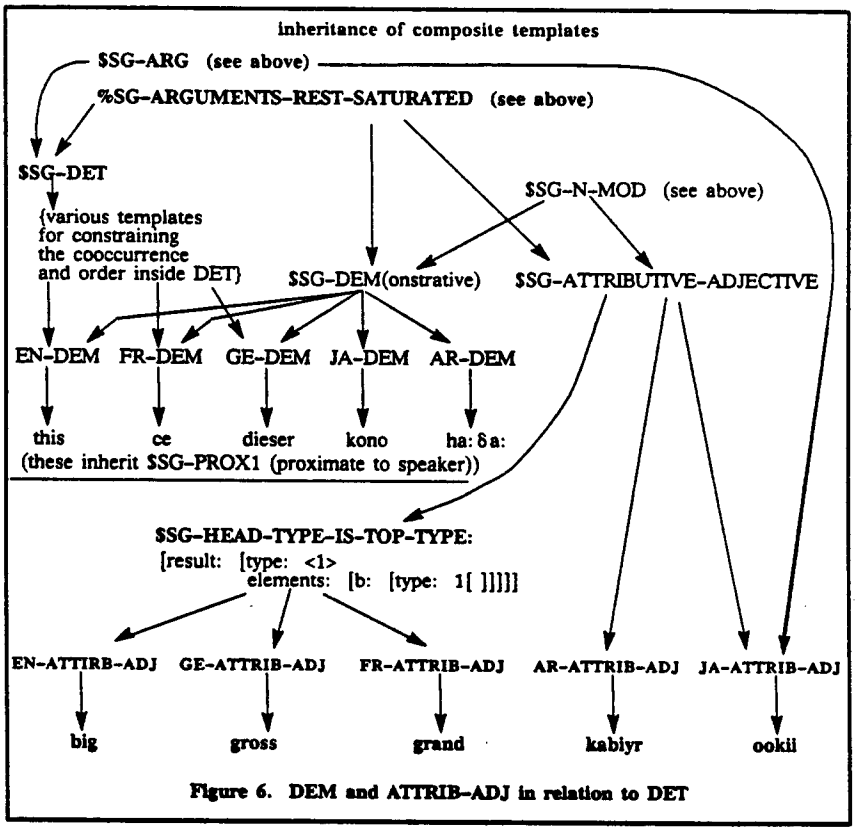


Figure 6. DEM and ATTRIB-ADJ in relation to DET

	NUMBER:	GENDER:	CASE:	DEFINITE:	ANNEXED
ARABIC:	SG DU PL3	M F	NOM ACC GEN	+ -	+ -
GERMAN:	SG PL	M F N	NOM ACC GEN DAT		
FRENCH:	SG PL	M F			
ENGLISH:	SG PL				

Table 1. N-internal Agreement Features

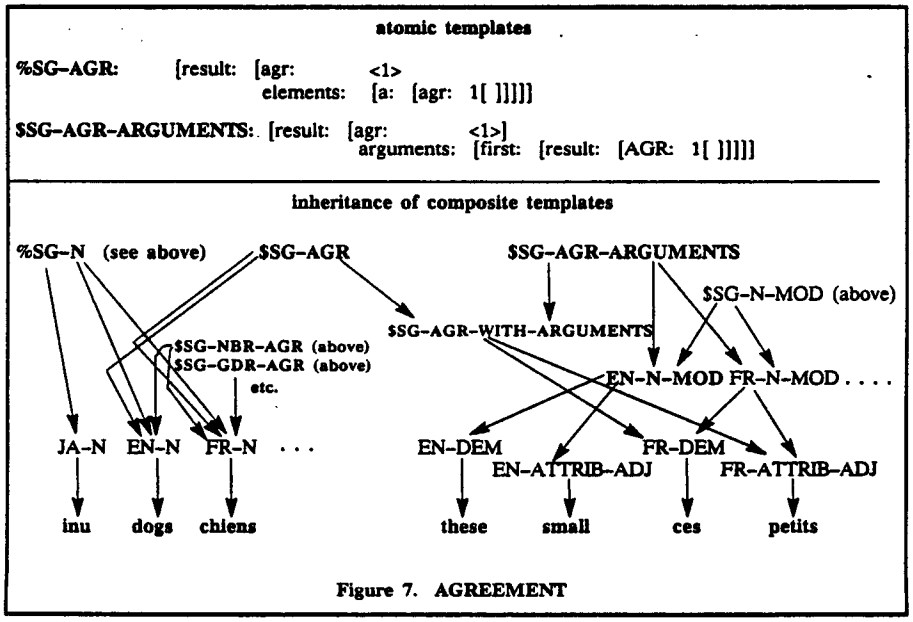


Figure 7. AGREEMENT

Arabic (HEAD < MOD) and Japanese (MOD < HEAD), with few exceptions. The three European languages, however, observe the default order only with 'heavier' (i.e. phrasal or clausal) modifiers, namely, genitives, pp-modifiers, and relative clauses. Lexical modifiers, including numerals, demonstratives, and adjectives (more or less), go in the opposite ordering. The exceptionally ordered MODs of the five languages revealed an implicational chain among modifiers: Numerals < Demonstratives < Adjectives < Genitives < Relative_clauses. Exceptional order was found with those MODs starting from the left-end of this hierarchy: JA: marked use of Numerals, AR: unmarked use of Numerals and Demonstratives, FR: Numerals, Demonstratives, and marked used of Adjectives, EN&GE: Numerals, Demonstratives, and Adjectives. The generalization is that a non-default order for a modifier type x implies the non-default order for other types located to the LEFT of x in the given chain. What we found supports the general implicational hierarchy that Hawkins (1984) found in his cross-linguistic study. We can still maintain, therefore, that there is such a thing as the default ordering, with a qualification that it may be overridden by non-random subclasses. In our current implementation, we simply assign another category MOD2 on those 'exceptional' modifiers in order to free them from the general order constraint on MOD, which we hope to improve in the future.[10]

Potential problems and solutions

There are two potential problems in an effort to develop a shared grammar as described here. One is the need for serious cooperation among the developers. A small change in shared templates can always affect language-specific templates that someone else is working on. The other problem is the sheer complexity of the inheritance lattice. Both problems can be most effectively reduced by a sophisticated editing tool.

[10]We envision using a data structure of type inheritance lattice defined for each language to express word order constraints in order to handle non-default ordering. The basic idea is that an order constraint stated on a descendant (e.g. DEM < head) overrides that stated on its ancestors (e.g. head < MOD). This differs from GPSG's LP rules (Gazdar & Pullum 1981; Gazdar et al. 1985; Uzkoreit 1986) in that the order constraints apply to items located anywhere in the derivational tree structrue, not limited to sister constituents, and the pieces of an item can be scattered in the tree. It is in spirit similar to LFG's functional precedence constraints (Kaplan 1988; Kameyama forthcoming).

Conclusions and future prospects

We have shown a specific implementation of grammar sharing using graph unification by inheritance. Although the case discussed covers only simple nominals in five languages, we believe that the fundamental process that we call GRAMMATICAL ATOMIZATION will remain crucial in developing a shared grammar of any structural complexity and linguistic coverage. The specific merits of this process is that (a) it tends to prevent the grammar writer from implementing treatments that work only for a language or a language type, and that (b) it provides insights as to how certain conflated properties in a language actually consist of smaller independent parts. In the end, when a prototype shared grammar attains a reasonable scale, we hope to verify the prediction that it will facilitate adding coverage for new languages.

The purpose of this work at MCC was to demonstrate the feasibility of a shared syntactic rule base for dissimilar languages. We only assumed that languages are used to convey information contents that can be represented in a common knowledge base. As the next step, therefore, we have chosen to connect syntax with 'deeper' levels of information processing (i.e. semantics, discourse, and knowledge base) rather than continuing to increase the syntactic coverage alone. Our current effort is on developing a blackboard-like system for controlling various knowledge sources (i.e. morphology, syntax, semantics, discourse, and a commonsense knowledge base (MCC's CYC, Lenat and Feigenbaum 1987)). In the future, we hope to see a shared grammar integrated in a full-blown interface tool for man-machine communication.

Acknowledgments

This shared grammar work is a collaborative effort of a team at MCC. I am especially indebted to my fellow linguists, Anthony Aristar and Carol Justus, for their insights into multilingual facts and numerous discussions. I would also like to thank Rich Cohen, Martha Morgan, Elaine Rich, Jonathan Slocum, Krystyna Wachowicz, and Kent Wittenburg for valuable comments and discussions at various phases of the work. Thanks also go to Al Mendall and Michael O'Leary for implementing the interface tool, and to anonymous ACL reviewers for helpful comments. I am responsible, however, for this particular exposition of the work and remaining shortcomings.

References

Ades, Anthony and Mark Steedman. 1982. On the order of words. Lingusitics and Philosophy, 4, 517-558.

Aristar, Anthony. 1988. Word-order constraints in a multilingual categorial grammar. To appear in the Proceedings for the 12th International Conference on Computational Linguistics, Budapest.

Bach, Emmon. 1986. The algebra of events. Linguistics and Philosophy, 9, 5-16.

Bar-Hillel, Y. 1953. A quasi-arithmetical notation for

syntactic description. Language, 29(1), 47-58.

van Benthem, Johan. 1986. Categorial grammar. Essays in Logical Semantics (Chapter 7). Dordrecht: Reidel, 123-150.

Flickenger, Daniel, Carl Pollard, and Thomas Wasow. 1985. Structure-sharing in lexical representation. The Proceedings for the 24th Annual Meeting of the Association for Computational Linguistics.

Flynn, Michael. 1982. A categorial theory of structure building. In G. Gazdar, G. Pullum, and E. Klein (eds), Order, Concord, and Constituency. Dordrecht: Foris.

Gazdar, Gerald and Geoffrey K. Pullum. 1981. Subcategorization, constituent order, and the notion 'head'. In Moortgat, M., H. v.d. Hulst, and T. Hoekstra (eds), The Scope of Lexical Rules. Dordrecht, Holland: Foris, 107-123.

____; Ewen Klein; Geoffrey K. Pullum; and Ivan A. Sag. 1985. Generalized Phrase Structure Grammar. Oxford, England: Blackwell Publishing and Cambridge, Mass.: Harvard University Press.

Greenberg, Joseph. 1966. Some universals of grammar with particular reference to the order of meaningful elements. In J. Greenberg (ed.), Universals of Language (2nd edition). Cambridge, Mass.: The MIT Press, 73-113.

Hawkins, John. 1984. Modifier-head or function-argument relations in phrase structure? The evidence of some word order universals. Lingua, 63, 107-138.

Kameyama, Megumi. forthcoming. Functional precedence conditions on overt and zero pronominals. Manuscript.

Kaplan, Ronald M. 1988. Three seductions of computational psycholinguistics. In Whitelock, Peter; Harold Somers, Paul Bennett, Rod Johnson, and Mary McGee Wood (eds), Linguistic Theory and Computer Applications. Academic Press.

Karttunen, Lauri. 1986. Radical lexicalism. Paper presented at the Workshop on Alternative Conceptions of Phrase Structure at the Summer Linguistic Institute, New York. [To appear in Kroch, Anthony et al. (eds), Alternative Conceptions of Phrase Structure.]

Keenan, Edward. 1979. On surface form and logical form. Studies in the Linguistic Sciences (special issue), 8(2).

Krifka, Manfred. 1987. Nominal reference and temporal constitution: towards a semantics of quantity. In J. Groenendijk, M. Stokhof, and F. Veltman (eds), Proceedings of the Sixth Amsterdam Colloquium, University of Amsterdam, Institute for Language, Logic, and Information, 153-173.

Lehmann, Winfred P. 1973. A structural principle of language and its implications. Language, 49, 47-66.

Lenat, Douglas B. and Edward A. Feigenbaum. 1987. On the thresholds of knowledge. Paper presented at the Workshop on Foundations of AI, MIT, June. Also in the Proceedings for the International Joint Conference on Artificial Intelligence, Milan.

Montague, Richard. 1974. The proper treatment of quantification in English. In Rich Thomason (ed.), Formal Philosophy: Selected Papers of Richard Montague. New Haven: Yale, 247-279.

Moravcsik, Edith. 1978. Agreement. In J. H. Greenberg et al. (eds), Universals of Human Language, Vol. 3. Stanford: Stanford University Press.

Pollard, Carl and Ivan Sag. 1987. Head-driven Phrase Structure Grammar. The course material for the Linguistic Institute at Stanford University.

Schmerling, Susan. 1983. Two theories of syntactic categories. Linguistics and Philosophy, 6, 393-421.

Shieber, Stuart. 1984. The design of a computer language for linguistic information. The Proceedings for the 10th International Conference on Computational Linguistics, 362-366.

____. 1986. An Introduction to Unification-based Approaches to Grammar. CSLI Lecture Notes 4. Stanford: CSLI. (available from the University of Chicago Press)

Slocum, Jonathan. 1988. Morphological processing in the Nabu system. In the Proceedings for the 2nd Conference on Applied Natural Language Processing. ACL.

____ and Carol Justus. 1985. Transprtability to other languages: the natural language processing project in the AI program at MCC. ACM Transactions on Office Information Systems, 3(2), 204-230.

Uzkoreit, Hans. 1986a. Constraints on order. Stanford, CA: CSLI Report No. CSLI-86-46.

____. 1986b. Categorial unification grammars. The Proceedings for the 11th International Conference on Computational Linguistics, 187-194.

Vennemann, Theo. 1974. Topics, subjects and word order: From SXV to SVX via TVX. In J. M. Anderson and C. Jones (eds), Historical Linguistics, I. Amsterdam: North-Holland, 339-376.

____. 1976. Categorial grammar and the order of meaningful elements. In A. Juilland (ed.), Linguistic studies offered to Joseph Greenberg on the occasion of his sixtieth birthday. California: Saratoga, 615-634.

____. 1981. Typology, universals and change of language. Paper presented at the International Conference on Historical Syntax, Poznan.

____ and Ray Harlow. 1977. Categorial grammar and consistent basic VX serialization. Theoretical linguistics, 4(3), 227-254.

Wittenburg, Kent. 1986a. Natural language processing with combinatory categorial grammar in a graph-unification-based formalism. Doctoral Dissertation, University of Texas at Austin.

____. 1986b. A parser for portable NL interfaces using graph-unification-based grammars. The Proceedings for the 5th National Conference on Artificial Intelligence, 1053-1058.