Lexicase Parsing: A Lexicon-driven Approach to Syntactic Analysis

Stanley STAROSTA
University of Hawaii Social Science Research Institute and Pacific International Center for High Technology Research Honolulu, Hawaii 96822, U.S.A.

Hirosato NOMURA
NTT Basic Research Laboratories Musashino-shi, Tokyo, 180, Japan

Abstract

This paper presents a lexicon-based approach to syntactic analysis, Lexicase, and applies it to a lexicon-driven computational parsing system. The basic descriptive mechanism in a Lexicase grammar is lexical features. The properties of lexical items are represented by contextual and non-contextual features, and generalizations are expressed as relationships among sets of these features and among sets of lexical entries. Syntactic tree structures are representaed as networks of pairwise dependency relationships among the words in a sentence. Possible dependencies are marked as contextual features on individual lexical items, and Lexicase parsing is a process of picking out words in a string and attaching dependents to them in accordance with their contextual features. Lexicase is an appropriate vehicle for parsing because Lexicase analyses are monostratal, flat, and relatively non-abstract, and it is well suited to machine translation because grammatical representations for corresponding sentences in two languages will be very similar to each other in structure and inter-constituent relations, and thus far easier to interconvert.

1. Introduction

There are a number of current frameworks of syntactic analysis which have been used as the basis for natural language processing. Many suffer from serious metatheoretical or practical defects, especially in the areas of power and descriptive adequacy. Several more recent syntactic frameworks, including Lexical-Functional Grammar [1], Generalized Phrase Structure Grammar [2], and Lexicase [3] have begun to take these problems seriously, and to consider applications to natural language processing. This paper will be concerned with the application of lexicase grammatical theory to computer parsing of natural language texts.

The point of view which we will adopt here is a very simple one: sentences are hierarchically structured strings of words, and grammar is a statement about the internal composition and external distributions of words. Proceeding from this basis, it is possible to construct a formal and explicit grammatical framework of limited generative power which is capable of stating language-specific and universal generalizations in a natural way, unhindered by pretheoretical a priori assumptions about VP's, etc. The framework so constructed, lexicase [3], [4], [5], turns out to have a significant potential for application in the processing of natural language [6].

The basic descriptive mechanism in a lexicase grammar is lexical features. The properties of lexical items are represented by contextual and non-contexttual features, and generalizations are expressed as relationships among sets of these features. The ways in which words can combine together are strongly restricted by the Sisterhead Constraint [3], which states that a word can contract a grammatical relationship only with the head of a dependent sister construction, and the One-bar Constraint [op. cit.], which requires every construction to have at least one lexical head. The result is syntactic tree representations which are flatter, since there are no intermediate nodes between lexical entries and their maximal projections, and more universal, since there are only a very limited number of ways in which languages can differ in their grammars. These properties turn out to make lexicase especially well suited to machine translation, since the grammatical representations for corresponding sentences in two languages will be very similar to each other in structure and inter-constituent relations, and thus far easier to interconvert.

This paper begins with a brief description of the basic structure of a lexicase grammar, and then describes an algorithm which applies lexicase principles to sentence parsing. Because of space limitations, we will not provide a full explication of the whole theory here. Instead, we will place the primary focus on the ways in which particular lexicase principles aid in the straightforward and efficient construction of syntactic tree representations for input sentences. Section 2 describes the way in which grammatical information can be presented as a set of generalizations about classes of lexical items represented in a dependency-type tree format. Section 3 describes the various types of lexicase features and their respective roles in a grammar. Section 4 discusses the representation of structural information about individual sentences in terms of a tree representation, and sections 5 and 6 present an algorithm showing how the information provided by a lexicase grammar may be used in parsing.

2. Rules and representations in lexicase theory

Lexicase is part of the generative grammar tradition, with its name derived from Chomsky's lexicalist hypothesis [7] and Fillmore's Case Grammar [8]. It has also been strongly influenced by European grammatical theory, especially the localistic case grammar and dependency approaches of John Anderson [9] and his recent and classical predecessors. Like Chomskyan generative grammar, it is an attempt to provide a psychologically valid description of the linguistic competence of a native speaker, but it differs from Chomsky's grammatical framework in power, since it has no transformational rules, and in generativity, since it requires grammatical rules and representations to be expressed formally and explicitly and not just talked about. The rules of lexicase grammar proper are lexical rules, rules that express relations among lexical items and among features within lexical entries. There are no rules for constructing or modifying trees, and trees are generated by the lexicon rather than by rules: the structural representation of a sentence is any sequence of words connected by lines in a way which satisfies the contextual features of all the words and does not violate the Sisterhead or One-bar Constraints or the conventions for constructing well-formed trees. A lexicase parsing algorithm, accordingly, is just a mechanism for linking pairs of words together in a dependency relationship which satisfies these contextual features and tree-forming conventions. ([11], [12], and [13] for a very similar but independently developed approach which evolved from the computational rather than the linguistic direction.)

Figure 1 lists the rule types in a lexicase grammar and their interrelationships. Redundancy rules supply all predictable features to lexical entries, which are stored in their maximally reduced forms, with all predictable features extracted. For example, all pronouns are necessarily members of the class of nouns, and since the feature [+N] is thus predictable from the [+prnn] (pronoun) feature, [+N] can be omitted from pronoun entries in the lexicon and supplied to the entry by a demon, a lexical Redundancy Rule, during processing.

Subcategorization rules characterize choices that are available within a particular category. These rules are of two subtypes, inflectional and lexical. For example, one inflectional

Subcategorization Rule states that English count nouns may be marked as either singular or plural. The other type of Subcategorization Rule does not allow an actual choice, but rather characterizes binary subcategories of a lexical category. For example, there is a non-inflectional Subcategorization Rule which states that English non-pronouns are either proper or common.

Inflectional Redundancy Rules state the contextual consequences of a particular choice of inflectional feature. Thus the choice of the feature 'plural' on a head noun triggers the addition of a contextual feature to its matrix stating that none of its dependent sisters may be singular.

Derivation Rules characterize relations between distinct but related lexical entries. For example, they provide a means of associating 'quality' adjectives with corresponding -ly manner adverbs. Due to the non-productivity of almost all derivational relations, both derived and underived lexical items must be stored and accessed separately in the lexicon, so these rules play only a minor role in parsing. (They are however the major lexicase mechanism for stating the interrelationships of sentence constructions such as active and passive clauses.)

Phrase-level phonological rules and anaphoric rules are the only non-lexical rules in the lexicase system. The latter mark pronouns, 'gaps' or 'holes', and other anaphoric devices as coreferential or non-coreferential, and so are a very important component of an adequate parsing system. However, a discussion of this question would go well beyond the intended boundaries of this paper.

With the rules and constraints outlined in this section, it is possible to radically simplify a grammar and the associated lexicon in ways which facilitate parsing, as detailed below.

3. Features in lexicase

As mentioned above, lexical features in a lexicase grammar are of two types: contextual and non-contextual. Contextual features specify ordering and dependency relationships among major syntactic categories ('parts of speech'), agreement and government requirements, and 'selection', semantic implications imposed by head items on their dependents. Non-contextual features characterize class memberships, including membership in major syntactic categories, subcategory features, inflectional features (including person, number, gender, and tense features as well as localistic case form and case relation features, which will not be discussed in this paper; but see [3]), and the minimum number of semantic features needed to distinguish non-synonyms from each other.

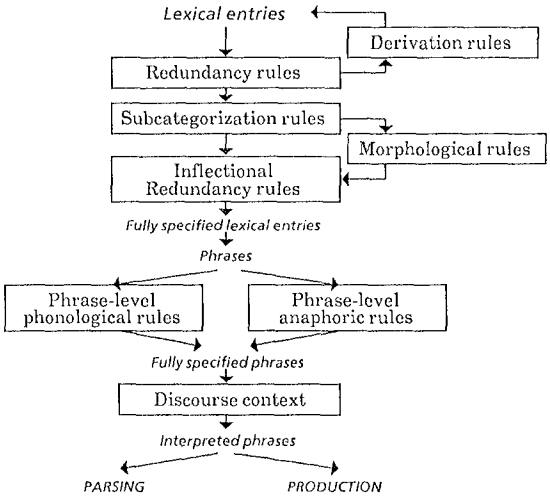


Fig. 1 Lexicase theory construction

(1) Case relations

Lexicase assumes only five 'deep' case relations, with inner and outer functions distinguished for three of them [5], as shown in Figure 2. The inventory of case relations is as short as it is because lexicase establishes a more efficient division of labor: much of the semantic information formerly carried by case relation differences in Fillmorean-type case relations is now carried by the semantic subcategory features of classes of verbs, and by the semantic features of the case markers themselves. The resulting reduced non-redundant case relation inventory improves the efficiency of case-related parsing procedures, and makes it possible to capture significant generalizations about case marking that are not possible with the usual extended inventories used in other case grammar and natural language processing systems. It is necessary to refer to case relations in parsing structures containing multi-argument predicates, in accounting for anaphora and semantic scope phenomena and text coherence, and of course in translation. Again, however, a discussion here of this aspect of lexicase parsing would go beyond the scope of this paper.

(2) Case forms

Unlike case relations, syntactic-semantic categories whose presence is inferred indirectly in order to account for lexical derivation and scope and anaphora phenomena, case forms are configurations of surface case markers such as word order, prepositions, postpositions, case inflections, or relator nouns which function to mark the presence of case relations. They are grouped together into equivalence classes functionally in terms of which case relations they identify, and semantically on the basis of shared localistic features as established by means of componential analysis. Case forms in a lexicase grammar are thus composite rather than atomic. Each is composed of one or more features, either purely grammatical ones such as ±Nominative (±Nom), which characterizes the grammatical subject of a sentence, or localistic ones such as source, goal, terminus, surface, association, etc.

Semantically, case forms carry most of the relational information in a sentence, and are used by the parser in recognizing the presence of particular case relations. For example, it is necessary to refer to them in for example identifying subjects in order to check for subject-verb agreement. Since so much 'case relation'-type information has been found to be present lexically in the case markers themselves, they bear much of the semantic load in the semantic analysis of relationships among lexical items, so that this information need not be duplicated by proliferating parallel case relations. This means that in parsing, such information is obtainable directly by simply accessing the lexical entries of the case-markers rather than by more complex inference procedures needed to identify the presence of the more usual Fillmore-type case relations.

Patient (PAT):
the perceived central participant in a state or event
Agent (AGT):
the perceived external instigator, initiator, controller, or experiencer of the action, event, or state
Locus (LOC):
inner: the perceived concrete or abstract source, goal, or location of the Patient
outer: the perceived concrete or abstract source, goal, or location of the action,event, or state
Correspondent (COR):
inner: the entity perceived as being in correspondence with the Patient
outer: the perceived external frame or point of reference for the action, event, or state as a whole
Means (MNS):
inner: the perceived immediate affector or effector of the Patient
outer: the means by which the action, state, or event as a whole is perceived as being realize

Fig. 2 Case relations in lexicase

(3) Syntactic category features

A small inventory of major atomic syntactic category features is assumed by lexicase, currently limited to the following seven: noun (N), verb (V), adverb (Adv), preposition or postposition (P), sentence particle (SPart), adjective (Adj), and determiner (Det).

Major syntactic categories are divided into syntactic subcategories based on differences in distribution. Thus nouns are divided into pronouns (no modifiers allowed), proper nouns (no adjectives and typically no determiners allowed), mass nouns (not pluralizable), etc., and similarly for the other syntactic classes. The contextual features associated with the words in these various distributional classes determine which words are dependent on which other words, and thus are very important in assigning correct trees to parsed sentences.

(4) Inflectional features

Traditional inflectional categories such as person, number, gender, case, tense, etc., are treated in lexicase as freely variable features which are not stored in their lexical entries (except in the cases of unpredictable forms), but are rather added as needed by a Subcategorization Rule in the course of processing. Inflection is typically involved in agreement, and agreement relationships (in conjunction with the Sisterhead Constraint) are important in locating and linking together those words bearing a head-dependent relationship to each other.

(5) Semantic features

Lexicase assumes that there must be enough semantic features marked on lexical items so that every lexical item is differentiated from every other (non-synonymous) item by at least one distinctive semantic feature. These features are not directly involved in parsing, but may figure in the identification of metaphors in sentences which do not have any other well-formed parsings.

(6) Contextual features

Contextual features are the part of the lexical representation which makes phrase structure rules unnecessary. A contextual feature is a kind of atomic valence, stating which other words may attach to a given word as dependents to form the molecules called 'sentences'. Contextual features may function syntactically, morphologically, or semantically. For example, the feature [–___[+Det]] on English nouns states that English determiners may not follow their nouns; another feature, [+[+Det]], is marked on definite common nouns to show that they must cooccur with determiners, and a third, [–[–plrl]], marks plural nouns as not allowing non-plural attributes. The feature [+([+Adj])] on common nouns states that they may have adjectival attributes, a possibility which would otherwise be excluded by the Omega-rule (see below).

Contextual features may refer to dependents occurring on the left or on the right, or they may be non-directional, referring to sister dependents on either side when the presence of some category is important but the order varies (as in topicalization and English subject-auxiliary inversion) or is irrelevant (as in free word-order languages).

Selectional features are also contextual, but they differ in function from grammatical contextual features. Thus a verb like 'love' may impose an animate interpretation on its subject by means of the following selectional feature: [⊃[+AGT, –anmt]]. Although the violation of a selectional feature does not result in ungrammaticality, selectional features are useful in parsing to pick the most promising branch in parsing a sentence when two or more different links are possible for a given word, or in identifying metaphors when no well-formed parse of a sentence is otherwise possible.

Since the 'range' of contextual features is sharply limited by the Sisterhead Constraint, only certain kinds of links between words are possible, and only those words directly connected by a single link need be checked for the satisfaction of grammatical requirements such as case frames, agreement features, etc. This greatly limits the number of places a parser has to check in determining the well-formedness of a given sentence, and so facilitates parsing.

Contextual features may be positive, negative or optional. Positive contextual features state the presence of a required dependent, and are used in parsing to establish initial links between pairs of words. Negative features identify classes of words which are not allowed to occur as dependent sisters, and serve in parsing to reject some of the links made in accordance with positive features. Optional features do not require or reject any links, but rather serve to keep open the possiblity of linking pairs of words by a general procedure applying near the end of the algorithm (see 6.3 below). All links which are not marked as permissible in this way are ruled out by the 'Omega Rule', a lexical Redundancy Rule which states the default value for the 'linkability' of given pairs of words: all linkings which are not explicitly allowed for are disallowed.

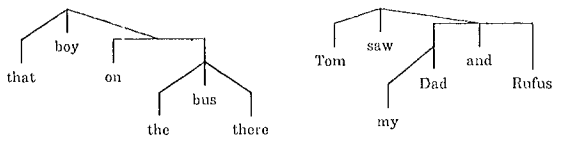
The most important characteristic for all contextual features for the purposes of parsing is the Sisterhead Constraint: in determining whether a contextual feature is satisfied for a given item, the parser need look only at the head words of its sister categories.

4. Lexicase tree representation

In lexicase, tree diagrams are graphic representations of dependency and constituency relationships holding among pairs of words in a sentence, and thus indirectly of relations among the constructions of which these words are the heads. Two types of constructions are recognized: endocentric and exocentric. These two construction types can be identified and their internal and external dependency relations determined directly from the kinds of lines by which they are connected in a lexicase tree representation (or, equivalently, by their bracketing in a LISP-type parenthesis notation):

i) vertical lines link a phrasal node with its head: a unit-length line indicates a lexical head, and a two-unit-length line identifies a phrasal head of an exocentric construction;
ii) slanting lines link an endocentric phrasal node with its dependents; and
iii) horizontal lines link the vertical lines above the lexical or phrasal heads of an exocentric construction.

An endocentric construction is any syntactic construction which has only one obligatory member, i.e. one head, which in accordance with the lexicase One-Bar Constraint must be a single lexical item. The other constituents of such constructions are phrases which are syntactically optional dependents of the head word. Noun Phrases and Sentences for example are endocentric constructions, headed by



(a) Noun phrase (b) Sentence

Fig. 3 Lexicase tree representations

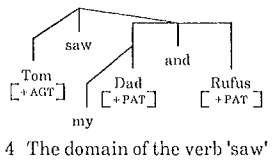
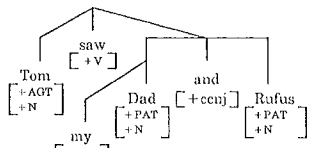


Fig. 4 The domain of the verb 'saw'



conj = coordinating conjunction

Fig. 5 Tree representation with category features

nouns and verbs respectively. In a tree, the head word of an endocentric construction has a vertical line of unit-length above it.

An exocentric construction on the other hand has more than one obligatory constituent. Again, the One-Bar Constraint requires that at least one of the constituents must be a single word, the lexical head of the construction. The other obligatory head (or heads) may be a word or a phrase. Examples of exocentric constructions are prepositional phrases and coordinate constructions. In a tree, each of the co-heads of an exocentric construction has a vertical line above it, of unit-length above lexical co-heads and two-unit-length above the lexical heads of phrasal co-heads. The apexes of the vertical lines are joined by a horizontal line, in effect an elongated node. Examples of both types of phrases appear in Figure 3.

The gramatically relevant relationships between pairs of nodes in a tree are expressed in lexicase in terms of the notions 'command' and 'cap-command' (from Latin caput, capitis 'head'):

i) a word cap-commands the lexical heads of its dependent sisters; thus in the two trees in Figure 3,
   a) 'boy' cap-commands 'that', 'on', and 'bus', since 'boy' has two dependent sister constituents (indicated by slanting lines), 'that' and 'on the bus there'. The lexical head of the construction 'that' (shown by a vertical line) is the word 'that'. However 'on the bus there' is an exocentric construction (shown by a horizontal line) which has two heads (shown by vertical lines), 'on' and 'the bus there'. The lexical head of 'on' is 'on', and the lexical head of 'the bus there (vertical line) is 'bus'.
   b) 'on' cap-commands 'bus', since 'on' has a single dependent sister (the phrasal co-head of the exocentric construction 'on the bus there'), 'the bus there', and the lexical head of 'the bus there' is 'bus'. Finally,
   c) 'bus' cap-commands 'the' and 'there', since 'bus' has two dependent sisters, 'the' and 'there', and the respective heads of these two constructions are the words 'the' and 'there'.

ii) a word X commands a word Y if either
   a) X cap-commands Y, or
   b) X cap-commands Z and Z commands Y.

Thus for example 'boy' commands 'there' because 'boy' cap-commands 'bus' and 'bus' cap-commands 'there'; however 'that' does not command 'there' because 'that' has no dependent sisters at all, and so does not cap-command anything.

The notion 'cap-command' plays a crucial role in defining the domain of subcategorization. To determine which constituents are relevant in subcategorization, lexicase appeals to the Sisterhead Constraint, which maintains that 'contextual features are marked on the lexical heads of constructions, and refer only to lexical heads of sister constructions' [3]. That is, a word is subcategorized only by the words which it cap-commands. For example, a verb may be subcategorized by the heads of the noun phrases which are its sisters, but not by the other constituents which are inside the NP's. Conversely, a noun may not be subcategorized by any constituent outside the NP. However, in the case of exocentric constructions such as prepositional phrases, the head words of both/all obligatory co-head constituents are accessible for subcategorization, since they are all cap-commanded by the higher head item.

To illustrate, in the Noun Phrase in Figure 3 (a), the lexical head of the construction is the noun 'boy'. Following the Sisterhead Constraint, the contextual features marked on 'boy' can refer only to features of the words it cap-commands, in this case 'that' and the heads of the exocentric PP, 'on' and 'bus', but not to 'the' or 'there'. The features of both the preposition and the head of its sister NP fall within the domain of subcategorization of the cap-commanding lexical item and jointly subcategorize it. Their features taken together are said to form a 'virtual matrix', i.e. a matrix which is not the lexical specification of any single lexical item, but which is rather a composite of the (non-contextual) features of all of the lexical heads of the construction [3]. In the lexicase parsing algorithm discussed in this paper, the effect of a virtual matrix has been achieved by copying the features of the phrasal head (the lexical head of the phrasal co-head, e.g. 'bus' in 'on the bus') into the matrix of the lexical head (e.g. 'on' in 'on the bus' in Figure 3). The matrix of the preposition 'on' then becomes in effect the virtual matrix of the exocentric construction, representing the grammatically significant features for the whole PP.

The Sisterhead Constraint makes it possible to define the notion of syntactic domain as all those constituents whose heads are referred to by the contextual features of a particular lexical item. For example, the domain of the verb 'saw' in the example of Figure 3 is indicated in Figure 4 with case relations. Thus the domain of the verb 'saw' in this sentence consists of the arguments marked [+PAT] and [+AGT]. The determiner 'my', on the other hand, is not in the domain of the verb; rather, it is in the domain of its own dominating noun, 'Dad'.

There are a number of other constraints in lexicase which apply to syntactic trees [3]. The effect of these constraints is to limit the class of possible trees and, consequently, the class of possible analyses. One constraint is that all terminal nodes are words, not morphemes or empty categories. A related constraint states that syntactic features are marked only on lexical items, not on nodes or on ad hoc abstract lexical categories. Finally, lexicase requires that every construction have at least one immediate lexical head; that is, there can be no intervening non-lexical node between the phrasal node and the lexical head of the phrase. In X-bar terminology, lexicase allows phrasal nodes with a maximum of one bar, where an S is equivalent to V-bar.

The interaction of the tree-drawing conventions, the One-bar limitation, and the Sisterhead Constraint makes it possible to eliminate both phrasal and major category labels from syntactic trees without any loss of information [3]. The matrix of an individual lexical item contains information about its syntactic category, making a category node label redundant. With the One-Bar Constraint, the nature of the phrasal construction can be determined with reference to the lexical category of the head of the construction, which is identifiable by the unit-length vertical line above it. Thus any node directly attached to a lower [+N] item by a vertical line of unit-length is an NP, so it is redundant to mark such a node by the label 'NP'. As a consequence, the tree representation in Figure 5 which has no node labels overtly marked is adequate for the representation of all constituency and dependency information. Note that the $\overline{\text{CCJN}}$ ('conjunction-bar') 'my Dad and Rufus' in Figure 5 is still an NP in function, because a coordinate construction is exocentric, and so the virtual matrix associated with 'my Dad and Rufus' contains the feature [+N] as well as [+cjn], making it an NP for external subcategorizing purposes.

The single-level lexicase tree notation incorporates the information carried by the three different kinds of tree structure contrasted by Winograd [10], dependency (head and modifier), phrase structure (immediate constituents), and role structure (slot and filler). Because it allows no VP constituent, it can equate constituent structure with dependency structure. The case role of a constituent is the case role of its lexical head. Thus semantic information is readily extracted from the syntactic representation, because the representation links together those words which are semantically as well as syntactically related.

5. The parsing algorithm

Figure 6 shows the fundamental components of the lexicase parser. The function of these components in brief is as follows:

(1) Pre-processor

This procedure replaces the word forms in the input sentence by homographic fully specified lexical entries, that is, entries with

identical spelling, specified for all contextual and non-contextual syntactic features as well as contextual and non-contextual semantic features ('selectional restrictions'). If an input form matches more than one lexical entry, replace the form by a 'cluster', a list of all the lexical entries whose forms match the input form. The output is a string composed of lexical entries and clusters of lexical entries which is isomorphous with the input string of word forms.

(2) Morphological analyzer

If an input form is not matched by any item listed in the lexicon, the morphological analyzer checks to see if the form matches any stored stem-affix pattern. If it does, the form is divided into stem plus inflectional affix and the stem is marked with the syntactic class features associated with the pattern. Using inflectional Subcategorization Rules, the stem is expanded into its full inflectional paradigm, and the original input word form is replaced by a 'cluster' composed of those (fully specified) members of the inflectional paradigm which are homographic with the original word form.

(3) Placeholder substitution

Each cluster of homographic lexical entries in the substitution string is temporarily replaced by a 'placeholder' entry composed of the intersection of the form and features of all the entries in the cluster. If the entries have nothing in common but the form itself, then the placeholder will be the form alone, with no associated feature matrix.

If the lexical entries in a cluster have enough features in common to be equivalent in terms of linking potential, they are linked into the tree structure as a group during the parsing process. When the structures containing clusters of entries are subsequently resolved into lexically unambiguous structures during placeholder expansion, many of the necessary links will have already been made, and will not have to be repeated for each separate but syntactically equivalent homographic entry.

(4) Placeholder expansion

Each substitution string containing placeholder clusters is expanded into separate structures by replacing the clusters with subclusters of items sharing more features in common, and ultimately with their original constituent individual entries. After each cluster is resolved into subclusters or individual entries, the resultant substitution strings are passed through the parser again to add links that become possible as the new clusters and entries become accessible.

As with the previous parsing phase, this phase establishes links that work for clusters of homographic items, so that these links do not have to be made separately and repeatedly for each substitution string containing a different homographic item. In this way, no sequence of words ever has to be reparsed.

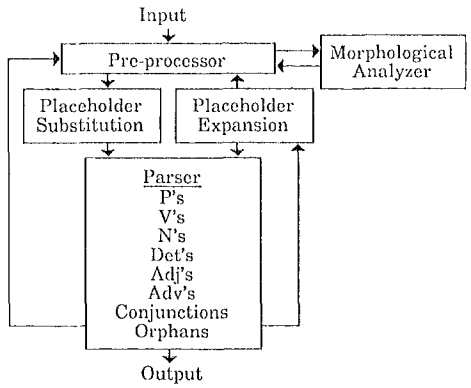


Fig. 6 Fundamental components of the lexicase parser

(5) Parser

Based on the positive contextual syntactic features of head lexical items, the heads are linked to eligible and accessible dependent items. As each link is established, the negative contextual features are checked. If there is a violation, that track is immediately abandoned. Note that exactly the same negative contextual feature mechanism takes care of two distinct contextual dependency phenomena:

i) general cooccurrence properties, such as the fact that English nouns may not have following Determiners, and
ii) grammatical agreement; thus for example subject-verb agreement is stated as a negative contextual feature: a finite verb marked for plural may not have a dependent Nominative sister marked singular. (Actually the matter is somewhat more complex than this, but a full discussion would go beyond the scope of this paper.)

After each pair of words has been linked in accordance with positive and negative grammatical contextual features, implicational semantic contextual features ('selectional restrictions') are checked for compatibility. If a violation is found, that string is semantically anomalous.

Lexicase theory is designed such that only the heads of sister categories need to be considered in determining whether there is an inconsistency in a structure being parsed. That is, only words directly connected by a single line need to be checked for the satisfaction or violation of any grammatical or selectional contextual requirement, and this checking can be done immediately after each link is first made. If a violation is found, the structure can be shunted off on a siding immediately without wasting time examining surrounding material. The parsing procedure will be considered in somewhat more detail in the section 6.

(6) Output

The output of the algorithm is zero or more syntactic analyses of the input sentence, but at the same time it can be considered an intensional semantic representation: it presents all the semantic distinctive features for each word, and specifies the head-modifier and semantic implication relations between each linked pair of words. The 'extensional' meaning of the sentence then is just the range of external situations which are compatible with the intension, the lexical meanings and interrelationships characterized by this structure. Lexicase is very well suited to characterizing this intensional semantic representation because it formally defines the range of possible lexical linkages. The structure is simple yet rich enough to in principle carry enough information to serve as the input to a knowledge extraction or machine translation system.

6. The parsing procedure

6.1 Words

(1) **Prepositions:** Link each preposition by contextual features with an accessible N, V, or P. Prepositions are linked first because they link with N's, V's, or other P's to form PP's which delimit closed domains whose internal non-head constituents are then inaccessible to connections with external elements. Subsequent parsing stages then search inside of or outside of these domains, but do not need to consider links between PP-internal non-heads and PP-external lexical items.

(2) **Verbs:** Verbs are linked with their attributes to form clauses or sentences. Note that in the lexicase framework, 'sentence' refers to any verb-headed construction, regardless of the finiteness of its verbal head or its position in the tree. The searching proceedes

from left to right in English, but would scan from right to left in a verb-final left-branching language such as Japanese. In a dependency grammar framework such as lexicase, a (verbal) sentence is defined as a verb together with its syntactic dependents. A sentence is the basic unit of syntax because it is the maximum domain of dependencies. Once a sentence unit has been established in this way, subsequent parsing stages can ignore links between sentence-internal and sentence-external items.

(3) Nouns: Nouns are linked with their dependents to form Noun Phrases. Noun Phrases and Sentences ('verb phrases') are the syntactically and semantically basic sentence constituents. Like other head items, nouns establish domains whose non-head constituents are inaccessible to external links, so that cross-domain linkages can be ignored on subsequent passes, thereby radically limiting the number of pairs of items that have to be considered on each subsequent pass and again cutting down on computation time.

(4) Determiners: Link each Determiner with an accessible Noun. In English, the Determiner marks the left boundary of a Noun Phrase. Linking the N and its Det establishes one boundary of the NP, and subsequent parsing can ignore links between elements inside this domain and elements outside it.

(5) Adjectives Link each Adjective with an adjacent noun. Because previous passes will have already delimited major constituent boundaries and radically narrowed the set of possible connections, very little checking will need to be done to link an Adjective with the correct head Noun.

(6) Adverbs: Link each Adverb with a head Verb or Adjective. Structural ambiguity is most likely to appear in connection with alternate attachments of PP's and Adverbs with other words in a sentence. By saving Adverb linking until near the end of the parsing sequence, we establish domains of inaccessibility which greatly reduce the number of possible Adverb attachment points which need to be considered.

6.2 Coordination

Link each conjunction with one or more major constituents (S, NP, PP, AdjP, or AdvP) on each side. At this point, all the major constituents have already been established, so the conjunction linking procedure needs to consider only the head word of each major constituent. Since every conjunction will at this time be either at the highest level, that is, linkable only to the immediate constituents of the sentence, or inside the domain of some other construction, the number of linking choices will be extremely limited.

6.3 Orphanage

Link all remaining upwardly unlinked Nouns, Determiners, Adjectives, Adverbs, Prepositions, and Verbs with an accessible 'elder sister' (or 'regent' [12]). At this point unattached lexical items will be found only embedded inside of other constructions, with very few accessible attachment possibilities to consider (usually only one). Thus there will generally be no backtracking and stacking required. The exception will be Adverbs and PP's, which account for most of the structural ambiguity likely to be encountered. By saving these alternative connection possiblities until near the end of the parsing process, we minimize the amount of computation that has to be done 'on top of' the alternative structures produced at this stage.

7. Overall assessment and conclusion

The parsing approach we advocate here is in principle very simple because lexicase requires no rules for normal parsing situations at all, and is based on linguistic principles designed to maximize the generality and simplicity of descriptions. It has no deep structure or transformations; instead, 'transformed' and 'untransformed' lexical entries are listed separately in the lexicon, thereby placing the parsing burden on memory rather than processing. Since lexicase automatically determines which items are relevant to the satisfaction of particular contextual requirements, no feature percolation or feature copying mechanism is needed to move features around in a tree to get them into a position where they are accessible to related items.

Lexicase parsing is bottom-up in the sense that it begins with individual words rather than some 'root node' S. It scans from left to right or vice versa, depending on whether the language is verb-initial, verb-medial, or verb-final, but in fact it is a mechanism which works from head to dependent rather than primarily from one end or the other. Since it forms constituents from heads and dependents at all levels simultaneously, it thus incorporates virtues of both top-down and bottom-up parsers. Lexicase accomplishes this by only making links allowed or required by contextual features of head lexical items, and since the 'overall structure of the sentence' is determined by just these features, it is not possible to make links which are not compatible with this overall structure.

Since lexicase has no Phrase Structure rules, a lexicase parser cannot blunder into the loops caused by left-recursive rules. Lexicase generates linguistically correct structures: they directly represent head-attribute relationships, they characterize the concept of grammatical relatedness, they allow various other important generalizations to be captured, and they account adequately for speakers' intuitions.

References

[1] Kaplan, R., and Bresnan, J., Lexical-Functional Grammar: A Formal System for Grammatical Representation. In J. Bresnan (ed), The Mental Representation of Grammatical Relations. Cambridge University Press. 1982.

[2] Gazdar, G., Klein. E., Pullum, G., and Sag, I., Generalized Phrase Structure Grammar. Harvard University Press. 1985.

[3] Starosta, S., The End of Phrase Structure as We Know it. Linguistic Agency - University of Duisburg (Trier) Series A, Paper no. 147. 1985.

[4] Starosta, S., Case in the Lexicon. Proceedings of the Eleventh International Congress of Linguists. 1975.

[5] Starosta, S., Patient Centrality and English Verbal Derivation. Proceedings of the thirteenth International Congress of Linguists. 1983.

[6] Starosta, S., and Nomura, H., Lexicase and Japanese Language Processing. Musashino Electrical Communication Laboratory Technical Report. 1984.

[7] Chomsky, N., Remarks on Nominalization. In Jacobs, R. A., and Rosenbaum, P. S. (eds), Readings in English Transformational Grammar. Ginn and Company. 1970.

[8] Fillmore, C. J., The Case for Case. in Bach, E., and Harms, R. T. (eds), Readings in English Transformational Grammar. Ginn and Company. 1970.

[9] Anderson, J., The Grammar of Case: Towards a Localistic Theory. Cambridge Studies in Linguistics 4. Cambridge University Press. 1971.

[10] Winograd, T., Language as a Cognitive Process, Volume I: Syntax. Addison-Wesley Publishing Company. 1983.

[11] Lehtola, A., Jäppinen, H., and Nelimarkka, E. Language-based environment for natural language parsing. Procedings of the Second European Conference of the Association for Computational Linguistics. 1985.

[12] Nelimarkka, E., Jäppinen, H., and Lehtola, A., A computational model of Finnish sentence structure. In Ann Såggeall Hein (ed), Föredrag vid De nordiska datalingvistik dagarna. 1983.

[13] Nelimarkka, E., Jäppinen, H., and Lehtola, A., Parsing an inflectional free word order language with two-way finite automata. In T. O'shea (ed), ECAI-'84: Advances in artificial intelligence. Elsevier Science Publishers B.V. 1984.