

# SRI International

## GENERATION AND RECOGNITION OF INFLECTIONAL MORPHOLOGY

Technical Note 461

March 20, 1989

By: John Bear  
Computer Scientist  
Artificial Intelligence Center  
Computer and Information Sciences Division

**APPROVED FOR PUBLIC RELEASE:  
DISTRIBUTION UNLIMITED**

This research was funded by the Defense Advanced Research Projects Agency  
under the Office of Naval Research contract N00014-85-C-0013.



333 Ravenswood Ave. • Menlo Park, CA 94025  
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

Generation and Recognition  
of  
Inflectional Morphology

John Bear  
Artificial Intelligence Center  
SRI International  
333 Ravenswood Ave  
Menlo Park, California 94025

March 20, 1989

**Abstract**

Koskenniemi's two-level morphological analysis system can be improved upon by using a PATR-like unification grammar for handling the morphosyntax instead of continuation classes, and by incorporating the notion of negative rule feature into the phonological rule interpreter. The resulting system can be made to do generation and recognition using the same grammars.

Topic Areas: morphology, unification grammar, generation.

# 1 Introduction

Since Koskenniemi published his 1980 dissertation on two-level morphology it has met with mixed reviews. Many scholars have pointed to the declarativeness of the system as very desirable feature. The two-level formalism for expressing phonological generalizations has the obvious advantage that any description of phonological facts can be used as well for generation as recognition. Furthermore, working recognition systems have been built for several languages using this paradigm, and it seems to have the potential to be used in many more.

Detractors have pointed out that there are many unsolved problems with the two-level paradigm. In particular, using a system of continuation classes to describe the morphosyntactic regularities of a language is cumbersome at best. Furthermore, while it has been demonstrated how to do recognition using continuation classes, it is not nearly so clear how to use them in doing generation. With regard to the model's handling of phonology, the use of arbitrary diacritics within words for encoding information, such as exceptionality, is considered by many to be unwieldy and highly undesirable. Even more undesirable is the need to manipulate transition tables for finite-state transducers rather than some form of phonological rule.

The intent of this paper is to describe work that has been done to remedy these problems. The main point being argued here is that this general paradigm of two-level morphology is both theoretically and practically useful. The text that follows presents my system for dealing with English inflectional morphology. This system has been implemented in Quintus Prolog, and in Common Lisp. It can be used for doing both recognition and generation of words. This work is based on the ideas of Koskenniemi [10,11,12], Karttunen and Wittenburg [7], and Shieber [15].

# 2 Basic Paradigm

There is a lexicon of morphemes. Each morpheme has two parts: a spelling and a collection of features and values. I use a unification grammar, so the collections of features are just directed acyclic graphs (dags). The spelling of a morpheme is what would be found in a dictionary. For instance, the English noun stem "spy" might have the following entry in the lexicon:

```
spelling: spy
features: [cat: noun_stem
          lex: spy
          type: regular]
```

There are two sets of rules describing linguistic phenomena. There are phonological rules (actually orthographic), which describe mappings between the spellings of morphemes, as they appear in the lexicon, and the spellings that occur in text. And there are morphosyntactic rules, which constitute a unification-based grammar specifying which morphemes may combine with which, and what the resulting dag (or feature set) should be.

The challenge is to arrive at two sets of declarative, reversible rules that may be used conjointly for both generation and recognition of word forms. The recognition task is clear. Given a word, e.g., "spies," one would like the recognizer to produce, for instance, the dag

```
[cat: noun
```

number: plural  
type: regular  
stem: spy].

Given that, the generation task becomes clear. Given a dag like the one immediately above, a generator should produce the word “spies,” and not, for instance, “spys,” or “spis,” or “spyes.”

Two types of mappings are involved here. One is the mapping of dictionary spellings to spellings of words as they appear in text. For example:

‘‘spy+s’’ <-> ‘‘spies.’’

The other is the mapping between a dag representing a word as it appears in text and a sequence of morphemes that would combine to form it, e.g.

[cat: noun		[cat: noun_stem	
number: plural	<->	lex: spy	+ [lex: s]
type: regular		type: regular]	
stem: spy].			

When the program is set to generate, it is given a dag as input and then it applies rules of the grammar to arrive at a sequence of dags, each representing a morpheme. It extracts a morpheme’s spelling from each of the dags to produce a sequence of morphemes. In the foregoing example, it would find the list [spy, s]. A morpheme boundary character is inserted between each pair of morphemes to give: spy+s. This last string serves as input to the phonological component. The phonological component decides which surface spellings might correspond to that string. In this case, there is only one alternative: spies.

The rest of this paper describes in more detail how the two grammars are used.

### 3 Morphosyntax

The morphosyntactic grammar is based on PATR (Shieber 1986), with an augmentation for doing disjunction based on an algorithm described by Karttunen (1984). The rules are all necessarily binary. They state how two dags can be combined to form a third. They can also be interpreted as specifying how a given dag can be decomposed into two others. Appendix 1 contains several of the verb formation rules used by this system.

In doing recognition, the parser is primed with a dag whose only feature is [category: empty]. The parser then proceeds through the word until it finds a string of characters that can constitute a morpheme. When it finds such a string, it retrieves the dag associated with it and then looks for a rule in the grammar telling how to unify the dag just found with the “empty” dag. If there is such a rule, it will describe how to build a new dag. This new dag is carried along as the parser looks farther down the word until another morpheme is found. The comparison process is then repeated with the new morpheme’s dag, and the dag that was being carried along.

Of course, the first morpheme that is found might be one that is not destined to lead anywhere. To handle such cases, the parser also continues down the word as if the morpheme just found were not really a morpheme. In short, every time a morpheme is found, two paths must be pursued. One

path assumes that the morpheme will indeed be part of the final analysis, while the other assumes that it will not.

In doing generation, the process is reversed. The engine starts with a dag and tries to find a rule telling how to decompose it. There might be more than one, of course. In that case, all the possible paths are followed. When a rule is found that tells how to decompose the dag at hand, two new dags are built and the generation engine is told to try to decompose one of them while simply remembering the other. This process continues until the generator is trying to decompose the dag [category: empty]. At that point it ceases doing decompositions and assembles a list of morphemes from the list of dags. This list of morphemes is passed as input to the phonology component whose task is to produce the appropriate spelling of the word.

## 4 Phonology and Orthography

The philosophy behind the phonological component is that there are two types of strings we are concerned with: morphemes in the lexicon and words in the text. When morphemes in the lexicon combine to form words, sometimes spelling changes happen. For instance (to use the “spy” example again), at the lexical level we could have the verb stem “spy” and the ending “s”. At the text or surface level, we have “spies.” We may talk of the following mapping:

(lexical level)	s	p	y	+	s
(surface level)	s	p	i	e	s.

We use two devices to describe the possible mappings between surface and lexical (underlying) strings of characters. One device is the list of feasible pairs of characters. It tells us which characters may ever correspond to which other characters. The default in this system is that any lexical alphabetic character (a-z) may correspond to itself on the surface, while lexical diacritics may correspond to the empty string (written as 0). In the example with “spy+s” above, the list of feasible pairs would need to indicate that, for instance, a lexical /s/ may correspond to a surface /s/; a lexical /p/ may correspond to a surface /p/; a lexical /y/ may correspond to a surface /i/, and so on. The converse is also true. If we know that a lexical /y/ may correspond to a surface /i/, then we also know that a surface /i/ may correspond to a lexical /y/.

The other device we use to describe possible mappings between surface and underlying strings is that of two-level constraints. The two-level constraints employed in this system are an outgrowth of those of Koskeniemi [10], of Karttunen and Wittenburg [7], and of Karttunen, Koskeniemi and Kaplan [5].

The basic idea is to stipulate constraints on when a certain lexical character is allowed to correspond to a certain surface character (and vice versa). Two different kinds of constraints are used: those which specify that certain string pairs are never allowable, and those that specify a certain character pair is only allowable in a certain context. There is third type of constraint that combines the other two. This last type looks very similar in form to a generative phonological rule, but it has additional power in that the context may refer to characters at both levels.

For instance the constraint:

+ --> e / {x | z | y/i | s ( h ) | c h } \_ s,

abbreviates a combination of two simpler constraints:

+/e allowed in context {x | z | y/i | s ( h ) | c h } \_ s

and

+/0 disallowed in context {x | z | y/i | s ( h ) | c h } \_ s.

The rules have several abbreviatory conventions, – for example, any character appearing by itself in the context really means a pairing of a lexical occurrence of that character with a surface occurrence. When a rule mentions a lexical character that is different from the surface character it is paired with, the pair is written with a slash, as is the pair y/i in the rule above. The symbol 0 is used to represent the empty string.

Sometimes it is necessary or desirable to be able to write a general constraint and then mark certain morphemes as exceptions to it. This can be done with negative rule features. For instance, if some morpheme were an exception to the rule given above, (and the rule had the name epenthesis1), then the morpheme's entry in the lexicon would contain the notation “-epenthesis1”, meaning that the epenthesis1 constraint need not hold of that morpheme. The use of negative rule features is described in more detail in another paper [3].

## 5 Conclusion

Many of the original objections that linguists have had to Koskenniemi's two-level model for doing computational morphology are no longer valid. Solutions to them have been found. There are elegant ways of dealing with the morphosyntax. There are also nice ways of dealing with exceptions to phonological rules. Moreover, it is possible to use the same grammars for both generation and recognition.

## Acknowledgments

I have benefited greatly from conversations with Lauri Karttunen and Kimmo Koskenniemi on the general problem of two-level phonology. I would also like to thank Stuart Shieber for discussions on the topic of unification grammars. This research was funded by the Defense Advanced Research Projects Agency under Office of Naval Research Contract N00014-85-C-0013.

## References

- [1] Bear, John (1985) “Interpreting Two-level Rules Directly,” presented at a Stanford workshop on finite-state morphology.
- [2] Bear, John (1986) “A Morphological Recognizer with Syntactic and Phonological Rules,” *COLING 86*.
- [3] Bear, John (1988) “Two-level Rules and Negative Rule Features,” to appear in the proceedings of COLING 88.
- [4] Karttunen, Lauri (1983) “Kimmo: A General Morphological Processor,” in *Texas Linguistic Forum #22*, Dalrymple et al., eds., Linguistics Department, University of Texas, Austin, Texas.

- [5] Karttunen, Lauri, Kimmo Koskenniemi and Ronald Kaplan (1987) "TWOL: A Compiler for Two-level Phonological Rules," distributed at the 1987 Summer Linguistic Institute at Stanford University, Stanford, California.
- [6] Karttunen, Lauri (1984) "Features and Values," in *COLING 84*.
- [7] Karttunen, Lauri and Kent Wittenburg (1983) "A Two-level Morphological Analysis Of English," in *Texas Linguistic Forum #22*, Dalrymple et al., eds., Linguistics Department, University of Texas, Austin, Texas.
- [8] Kay, Martin (1983) "When Meta-rules are not Meta-rules," in K. Sparck-Jones, and Y. Wilks, eds. *Automatic Natural Language Processing*, John Wiley and Sons, New York, New York.
- [9] Kay, Martin (1987) "Nonconcatenative Finite-State Morphology," paper presented at a workshop on Arabic Morphology, Stanford University, Stanford, California.
- [10] Koskenniemi, Kimmo (1983) *Two-level Morphology: A General Computational Model for Word-form Recognition and Production*. Publication No. 11 of the University of Helsinki Department of General Linguistics, Helsinki, Finland.
- [11] Koskenniemi, Kimmo (1983) "Two-level Model for Morphological Analysis," *IJCAI 83*, pp. 683-685.
- [12] Koskenniemi, Kimmo (1984) "A General Computational Model for Word-form Recognition and Production," *COLING 84*, pp. 178-181.
- [13] Schane, Sanford (1973) *Generative Phonology*, Prentice Hall, Englewood Cliffs, New Jersey.
- [14] Selkirk, Elizabeth (1982) *The Syntax of Words*, MIT Press, Cambridge, Massachusetts.
- [15] Shieber, Stuart (1986) *An Introduction to Unification-Based Approaches to Grammar*, CSLI Lecture Notes Series, Stanford University, Stanford, California.

Appendix 1 (Morphosyntax)

% Some of the verb rules from this system's English grammar.

% verb --> empty + verb\_stem

% 1 2 3

<2 cat> = empty

<3 cat> = verb\_stem

<3 type> = regular

<1 type> = <3 type>

<1 cat> = verb

<1 word> = <3 lex>

<1 form> = {inf

[tense: pres

pers: {1 2}

nbr: {pl sg}] } .

% verb --> empty + verb\_stem

% 1 2 3

<2 cat> = empty

<3 cat> = verb\_stem

```

<3 type> = irregular
<1 type> = <3 type>
<1 cat> = verb
<1 word> = <3 word>
<1 form> = <3 form> .
% verb --> verb + ing
% 1      2      3
<2 cat> = verb
<3 lex> = ing
<2 form> = inf
<1 cat> = verb
<1 word> = <2 word>
<1 form> = [tense: pres_part] .
% verb --> verb + ed
% 1      2      3
<3 lex> = ed
<2 cat> = verb
<2 form> = inf
<2 type> = regular % to prevent "sleaped"
<1 cat> = verb
<1 word> = <2 word>
<1 form> = [tense: {past_part past}] .
% verb --> verb + s
% 1      2      3
<3 lex> = s
<2 cat> = verb
<2 form> = inf
<1 cat> = verb
<1 word> = <2 word>
<1 form> = [tense: pres
           pers: 3
           nbr: sg] .
% verb --> empty + verb_past
% 1      2      3
<2 cat> = empty
<3 cat> = verb_past
<1 cat> = verb
<1 word> = <3 word>
<1 form> = [tense: past] .

```

## Appendix 2 (Lexicon)

```

ed
□
.
ing
□
.

```



```

s
[]
.
try
[cat: verb_stem
 type: regular]
.
spy
[cat: verb_stem
 type: regular]
.
banjo
- epenthesis2
[cat: noun_stem
 type: regular]
.
piano
- [epenthesis2 epenthesis3]
[cat: noun_stem
 type: regular]
.
box
- gemination
[cat: verb_stem
 type: regular]
.

```

### Appendix 3 (Phonology)

```

%%% -- Mode: PROLOG --

```

```

epenthesis1:

```

```

'+' --> e / {x | z | y/i | s ( h ) | c h } _ s.

```

```

epenthesis2:

```

```

'+' --> e / o _ s.

```

```

epenthesis3:

```

```

'+' / e allowed in context o _ s.    % Allow words like 'banjoes'.

```

```

epenthesis4:

```

```

O/e allowed in context z '+' / z _ s.

```

```

gemination:

```

```

'+' --> c1 / {cC | q u} vV c1 _ {0/vV | vV} where c1 is in cC.

```

```

y_spelling:

```

```

y --> i / cC _ '+' / '=' cC.

```

```

y_spelling:

```

```

y --> i / cC _ '+' / '=' e cC.    % For spied, tried,

```

```

elision:

```

```

e --> 0 / cC _ '+' / 0 { e | i }.

```

```

elision:

```

```

e --> 0 / vV _ '+' / 0 e.

```

elision:

e --> 0 / i/y \_ '+'/0 vV.

elision:

e/0 allowed in context c\_CG \_ '+'/0 a.

ie\_spelling:

i --> y / \_ e/0 '+'/0 i.

ie\_spelling:

l --> 0 / \_ e/0 '+'/0 l y.

ie\_spelling:

e --> 0 / l/0 \_ '+'/0 l y.

#### Appendix 4

List of feasible pairs in addition to the pairs (x,x), where x is any alphabetic character:

(' ,0), (l,0), (e,0), (i,y), (y,i), (+,b), (+,d), (+,f), (+,g), (+,l), (+,m), (+,n), (+,p), (+,r), (+,s), (+,t), (+,z), (+,e), (+,0)

Progressive is the progressive or *during* time, occurring during the cake-frosting event:

- (14) *precede(Start, Progressive)*  
*precede(Progressive, End)*

The perfect time is Perfect. By the clause *precede(Progressive, Perfect)*, the progressive time Progressive is constrained to precede the perfect time. In other words, for a perfect progressive sentence, the requirement is that some portion of the main event lie before the perfect time. The perfect time is constrained by the clause *strictly-precede(now, Perfect)* to lie in the future.

Given a representation like this one, the advantages of Harper and Charniak's system are retained. Underspecification of relations among times yields results that match the natural-language semantics of sentences.<sup>4</sup> Use of a perfect and a progressive time allows uniform treatment of perfects and progressives without the complication of introducing otherwise unwarranted reference events into the representation of simple tenses. The disadvantages of Harper and Charniak's rather cumbersome use of events instead of times in their representations are also allayed.

## 6 Temporal Interpretation of Nontensed Elements

Not only tensed verbs, but also other nontensed elements in the sentence – adjectives, nouns, prepositions, and so on – must be interpreted with respect to some instant or time interval. Consider the sentence

- (15) *Are there any warm cakes?*

The adjective *warm* must be interpreted relative to some time: in this case, the present. The question is about cakes that are currently warm.

The interpretation of nontensed elements does not always depend on the time the sentence is uttered, though. The sentence

- (16) *The third-year students had to take an exam last year.*

can be interpreted in two ways. Under one interpretation, those who were third-year students last year (the current fourth-year students) had to take a test last year. The interpretation of the noun phrase *the third-year students* is dependent on the tense of the main verb in this case. Under the other interpretation, those who are currently third-year students took a test last year, when they were second-year students.

Another example of the same type is:

---

<sup>4</sup>We have not yet enriched the representation of individual predicates to include inherent aspect, as described in, for example, Passoneau (1987). We feel, though, that the resulting representations will still involve the use of perfect and during times, and will still be amenable to the treatment of nontensed elements described in the next section.

(17) *John will frost a warm cake at 3:00.*

Under one interpretation, the cake is warm at 3:00 when he frosts it, but it is not necessarily warm at the time of utterance of the sentence. Under the other interpretation, the cake is warm now, but it may not be warm when he frosts it.

However, the interpretation of nontensed elements with respect to the tense of the main verb in the sentence is not entirely unconstrained. Consider the following sentence:

(18) *The wife of the president was working in K-Mart in 1975.*

*Wife* and *president* are both predicates that must be interpreted with respect to a particular time. The current president is not the same as the 1975 president; if he divorced and remarried, his 1975 wife is not necessarily the same person as his current wife.

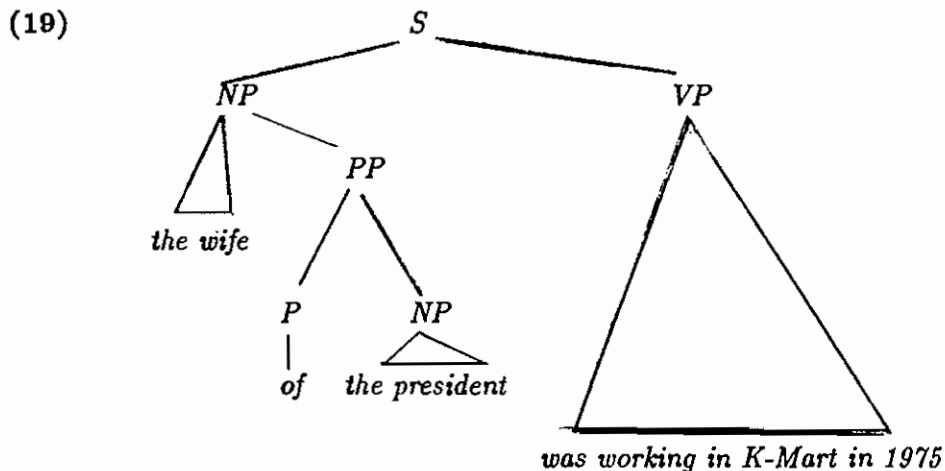
Given the fact that both *wife* and *president* are interpretable with respect to either the time of the tensed verb or the time of the utterance, there ought to be four possible interpretations of this sentence. In fact, there are only three:

- He is the current president and she is his current wife
- He is the current president and she was his wife in 1975
- He was the president in 1975 and she was his wife then (but perhaps he is divorced and no longer president)

The missing interpretation is that

- He was the president in 1975 and she is his current wife (but was not his wife then)

A skeletal tree for this example is:



The sentence involves the syntactic embedding of one NP (*the president*) inside another NP (*the wife*). The unavailable interpretation is one in which the embedded NP is interpreted with respect to the event time of the higher verb, whereas the intervening NP is not. That is, the unavailable interpretation involves interpreting a discontinuous portion of the tree with respect to the main verb.<sup>5</sup> It is not possible to interpret a deeply embedded predicate with respect to the event time of the main tensed verb unless the intervening material is also interpreted in the same manner.

As a first approximation, one may think of the main-verb event time as being passed or disseminated through the tree. It may be passed down to embedded predicates in the tree only when it is passed through intermediate predicates and used in their interpretation. If a predication is interpreted with respect to the current time rather than to the event time of the main verb, all predications that are syntactically subordinate to it are also interpreted with respect to the current time. When this happens, the main-verb event time ceases to be passed down and may not be reinstated for interpretation.

Note, however, that the verb time and the time with respect to which the nontensed elements are interpreted are not always completely coextensive. Consider again the example

(20) *John will be frosting a warm cake at 3:00.*

Under the interpretation that the cake is warm while John is frosting it, the time span during which the cake is warm must include the time 3:00, but the cake may have been warm before John starts frosting it; moreover, it may continue to be warm after John finishes frosting it. That is, the starting and ending points of the cake-frosting event need not coincide exactly with the starting and ending points of the interval at which the cake is warm. The only requirement is that both events must hold at 3:00.

Now consider the sentence

(21) *John built a new house.*

The building event can be thought of as beginning before the event of the house's being new. At the start of the building event, there is no house, nor, obviously, is there any event of the house's being new. In a situation like this, one does not want to require that the building event be contained within the event of the house's being new, but rather, merely to require that the two events should overlap.

Our claim is that, in general, temporal interpretation of nontensed elements relative to the tense of the main verb of the sentence requires only that the event denoted by the main verb overlap (not be coextensive with or be contained in) the events denoted by the nontensed elements. We shall accomplish this by positing a time for each main verb, the

---

<sup>5</sup>As we will see in the next section, it is possible to construct a context in which the "missing interpretation" is in fact available for this sentence. The claim made here is that this interpretation is not available by means of the syntactic variable-passing mechanism discussed in this section, but is only available by appealing to the context constructed. The "missing interpretation" is missing when there is no context to refer to for additional interpretations.

during time, and passing this time through the syntactic tree. The event denoted by the main verb, as well as the events denoted by any predicates interpreted relative to the main verb, must hold at this *during* time.

For example, here is the logical form for the sentence *John frosted a warm cake*:

- (22) *exists C Start1 End1 During Start2 End2*  
*holds(frost(john, C), Start1, End1)*  
*ℰ cake(C)*  
*ℰ precede(End1, now)*  
*ℰ precede(Start1, During)*  
*ℰ precede(During, End1)*  
*ℰ holds(warm(C), Start2, End2)*  
*ℰ precede(Start2, During)*  
*ℰ precede(During, End2)*

There are two predicates in this example that are interpreted with respect to a temporal interval: *warm* and *frost*. The first *holds* predicate

- (23) *holds(frost(john, C), Start1, End1)*

specifies that John frosts C (required by the second clause in the example to be a cake) from time Start1 to time End1. The second *holds* predicate

- (24) *holds(warm(C), Start2, End2)*

specifies that C is warm from time Start2 to time End2. The *precede* predicates specify the relations between these times. The predicate *precede(End1, now)* specifies that the end of the cake-frosting event precedes *now*, the time of utterance. That is, the cake-frosting event must lie in the past. The next two predicates

- (25) *precede(Start1, During)*  
*precede(During, End1)*

specify that time *During* is the *during* time for this event. The final two predicates

- (26) *precede(Start2, During)*  
*precede(During, End2)*

specify that the *during* time of the cake's being warm is also *During*. In other words, there must be a time *During* that occurs during both the cake-frosting event and the event of the cake's being warm: the two events must overlap.

Since the interpretation of every sentence involves positing a *during* time, we propose that progressive aspect involves nothing more than a kind of syntactic access to this *during* time. What Harper and Charniak call the “progressive event” is simply the *during* time in a syntactically accessible form.

For example, consider the sentence

(27) *John was frosting warm cakes at 3:00.*

The modifier *at 3:00* instantiates the *during* time to 3:00. In this case, there are two predicates whose interpretation is temporally dependent: the adjective *warm* and the verb *frost*. Instantiating the progressive/*during* time to 3:00 means not only that there must be an event of John’s frosting the cakes going on at 3:00, but also that the cakes must be warm at 3:00. It is not enough simply to require that the cakes were warm at any time during the event of John frosting them.

Further, all elements within a NP node are interpreted with respect to the same event. It is not possible, for example, to interpret some elements of a noun phrase with respect to the time of utterance, others with respect to the main verb *during* time. Consider the sentence

(28) *John frosted three stale warm cakes yesterday.*

Despite the pragmatic predilection for interpreting *stale* and *warm* at different times (it is hard to imagine how cakes that are still warm could already be stale), this sentence has only two interpretations:

- John frosted three cakes that were both stale and warm yesterday.
- John frosted three cakes yesterday that are both stale and warm now.

It is not possible to give the sentence the interpretation that the cakes he frosted were warm yesterday and are stale now, or were stale yesterday and are warm now. Both adjectives must be interpreted with respect to the same time.

If a system like Harper and Charniak, in which events and not instants are taken to be the relevant reference points, were extended to include interpretation of nontensed elements as described here, such a system might use primitives such as those of Allen (1984). However, none of the primitives of Allen’s system is suitable for defining the relation of the *during* time to the main event: *during(DuringEvent, MainEvent)* is not sufficient, since Allen’s “during” relation does not permit the *DuringEvent* to coincide with the beginning or end points of the main event. The example “John built a new house” shows that this is necessary; in this case, it is precisely the end point of the building event that coincides with the beginning of the event of the house being new. In a system using Allen’s primitives, the proper relation between the *DuringEvent* and the *MainEvent* would be a disjunction:

(29) *during(DuringEvent, MainEvent) OR*  
*starts(DuringEvent, MainEvent) OR*  
*ends(DuringEvent, MainEvent)*

## 7 Passing the During Time: Rules for Temporal Interpretation

In the previous section, we examined the temporal interpretation of phrases with respect to the *during* time of the main verb. In addition, we proposed a constraint on the passing of this *during* time from the verb through its arguments and adjuncts, according to which predicates interpreted according to the *during* time must occupy a nondiscontinuous portion of the tree. From the point of view of the tenseless phrase, however, the same process can be seen in a different light.<sup>6</sup>

We may think of the interpretation of temporally dependent elements in a phrase as proceeding in the following manner:

- The phrase is interpreted with respect to a temporal modifier internal to the phrase; otherwise
- The phrase is interpreted with respect to the closest higher tensed element (allowing for restrictions on the distribution of the *during* variable); otherwise
- The phrase is interpreted with respect to some contextually relevant time.

Temporally dependent nontensed elements in previous sections were always contained in phrases that lacked internal temporal modifiers, so the first option was not applicable. One of two interpretations was given for tenseless elements: they were interpreted with respect either to the *during* time of the main verb or to *now*, the time of utterance. Interpretation with respect to *now* seems to be a particular instance of the general possibility of interpretation with respect to a contextually relevant time; since no context was given for the examples in the previous sections, no other contextually relevant time was available. When a phrase contains a phrase-internal temporal modifier, the predicates in that phrase must be interpreted with respect to that modifier, as in the example

(30) *The 1975 president is living in California.*

The modifier *1975* in the phrase *the 1975 president* provides the temporal interpretation of the phrase: it must be interpreted with respect to that time. It is not possible to interpret *president* relative to the *during* time of the main verb.

Hinrichs (1987) also proposes that noun phrases be interpreted relative to a time restricted by the context; the difference between his analysis and ours is that, of the three options presented above, he offers only the last. He contends that the only option for temporal interpretation of nontensed elements is the third one, namely, by reference to context.

Given an analysis like that of Hinrichs, it is difficult to explain the facts noted in the preceding section. In the absence of context (or when the sole context is the moment of utterance), Hinrichs would not predict the absence of one reading for sentences such as

---

<sup>6</sup>I am grateful to Bill Croft for helpful discussion on these points.



(31) *The wife of the president was working in K-Mart in 1975.*

(cannot mean that he was the president in 1975 and she is his current wife, but was not his wife then)

In an analysis like the one presented here, where the interpretation of nontensed elements is determinable in some instances through syntactic processes, the absence of these readings is expected.

It seems that it is not possible to pass a *during* time through a NP containing an internal temporal modifier. Consider the following sentence:

(32) *The 1970 wife of the president was working in K-Mart in 1975.*

The main verb *during* time is constrained to be in the year 1975, whereas the *wife* predicate must hold in 1970. It is difficult to attach any strict temporal interpretation to the noun phrase *the president*; it seems that its interpretation must be done pragmatically.<sup>7</sup>

Enc (1981) and Hinrichs (1987) both argue convincingly that there are many instances in which a temporally dependent element is interpreted with respect to a time that is neither the *during* time nor *now*. Hinrichs furnishes the following example:

(33) *Oliver North's secretary testified before the committee.*

*then sec'y*

At the time she testified, she was no longer his secretary; she was also not his secretary at the time this sentence was uttered. It would receive the following interpretation:

(34) *exists FH Start1 End1 During1 Start2 End2 During2*  
*holds(secretary(north, FH), Start1, End1)*  
*ℰ precede(Start1, During1)*  
*ℰ precede(During1, End1)*  
*ℰ holds(testify(FH), Start2, End2)*  
*ℰ precede(Start2, During2)*  
*ℰ precede(During2, End2)*  
*ℰ precede(End2, now)*

The first four clauses specify simply that there is an event of FH being secretary to Oliver North that holds from Start1 to End1. During1 is a time during this event.

The next clauses specify that there is a testifying event by FH that holds from time Start2 to time End2. The *during* time for this event is During2. As indicated by the clause *precede(End2, now)*, this event is in the past.

<sup>7</sup>It is, of course, possible to construct a discourse context for this example such that either 1970 or 1975 will be the preferred time with respect to which *the president* is interpreted; the claim here is that these interpretations do not result from the syntactic option of passing of a *during* time through a tree, but are supplied pragmatically.

In this example, the event of FH's being a secretary is not required to overlap with the event of FH's testifying. The *during* times are merely existentially quantified over. In a more complete representation, the appropriate restrictions would be imposed on time Start2: the time during which FH is a secretary would be restricted by the context, in line with Hinrichs' suggestions.

## 8 Progressive Aspect and Stative Verbs

There are interesting similarities between progressives and statives. These similarities come about because the *during* time is important in the interpretation of both. Mathiessen (1984) provides the following example in which both perfect aspect and a stative verb are used:

(35) *In 1970, John had lived in Kuala Lumpur for five years.*

This contrasts with a sentence not involving a stative verb, like the following:

(36) *In 1970, John had read War and Peace.*

In the first example, it is not necessarily the case that John is no longer living in Kuala Lumpur, nor that he did not continue to live there after 1970. On the other hand, in the second example, John must have finished *War and Peace* by 1970; he cannot still have been in the process of reading it.

On the other hand, consider the following sentence:

(37) *In 1970, John had been reading War and Peace for five years.*

Here it is possible that John is still reading *War and Peace*, and that he continued to read it after 1970. These facts parallel those in the example containing the stative verb *live*.

The same facts obtain in simple tenses. Consider the following sentence:

(38) *John lived in Kuala Lumpur in 1975.*

Let us assume that the current time is after 1975. This sentence can be true in a situation when John is still living in Kuala Lumpur; the following sentence is not a contradiction:

(39) *John lived in Kuala Lumpur in 1975, and he is still living there.*

However, if a nonstative verb is used, the facts are different:

(40) *John frosted the cake at 3:00.*

This sentence entails that the event of frosting the cake is complete: that its end point lies in the past. The following sentence is not felicitous:

(41) # *John frosted the cake at 3:00, and he is still frosting it.*

However, this sentence is not a contradiction:

(42) *John was frosting the cake at 3:00, and he is still frosting it.*

We claim that this is because statives and progressives are alike in the following way: both involve predication not over the beginning or end of the main event, but over the *during* time. This is further evidence that the *during* time is involved in the interpretation of every sentence, not just progressives.

## 9 Further Results

It appears that the *during* time of the main clause is used in some instances in the interpretation of tensed subordinate clauses: for example, in the interpretation of relative clauses. Consider the sentence “He will catch the dog that is running.” Under one interpretation of this sentence, the catching event is simultaneous with the running event – both events take place in the future. In this case, the interpretation of the main verb in the relative clause depends on the *during* time of the main clause. There is also another interpretation, according to which the dog that will be caught later is running now. In this case, the interpretation of the relative clause depends on the time of utterance of the sentence. This seems to be possible only with future tense sentences, however: there is only one interpretation for “He caught the dog that is running”, the one where the dog is running when the sentence is uttered. The interpretation where the dog is running when it is caught is not available.

One remaining task is to provide a reasonable analysis of the bare present using this system, including generic sentences like “It is (usually) cold in Alaska.” We feel that such an analysis awaits the incorporation of a representation of inherent lexical aspect as in Passoneau (1987); without a representation of the distinction between (for example) states and activities, a coherent representation of simple present tense sentences is not possible.

Another task is to provide a reasonable account of the imperfective paradox. For a sentence like

(43) *John was reading a book at 3:00, but he never finished reading it.*

the end point of the reading event is asserted to exist and to follow the *during* time of the book-reading event; the *during* time is instantiated to 3:00 and precedes the End time. The second clause asserts that the culmination of the reading event, usually taken to be equivalent to the end point of the reading event, did not occur. This remains an outstanding

problem; the solution might be to allow events to end without being culminated, though.<sup>8</sup> That is, the end of an event would not always be associated with the completion of that event; in the case described above, the end of the book-reading event would occur, but the inference that John had therefore finished the book would not be valid.

## 10 Conclusion

We have shown that distributing an existentially quantified *during* time variable throughout the tree enables interpretation of nontensed elements in the sentence according to the time of the main verb. Further, the *during* time is useful in the interpretation of several sentence types: progressives, statives, and sentences containing relative clauses. Finally, an analysis that utilizes underspecified relations among times (not events) provides a good prospect for analyzing tense and aspect in English.

---

<sup>8</sup>This suggestion was made by a discussant at the ACL conference, Buffalo, N.Y., June 1988.

## References

- Allen, James F. 1984. "Towards a General Theory of Action and Time." *Artificial Intelligence* 23:2, July 1984.
- Enc, Murvet. 1981. "Tense without Scope: An Analysis of Nouns as Indexicals." Ph.D. dissertation, University of Wisconsin, Madison, Wisconsin.
- Harper, Mary P. and Eugene Charniak. 1986. "Time and Tense in English." Proceedings of the ACL Conference, Columbia University, New York, New York.
- Hinrichs, Erhard. 1987. "A Compositional Semantics of Temporal Expressions in English." Proceedings of the ACL Conference, Stanford University, Stanford, California.
- Hornstein, Norbert. 1977. "Toward a Theory of Tense." *Linguistic Inquiry*, Vol. 8, No. 3.
- Mathiessen, Christian. 1984. "Choosing Tense in English." ISI Research Report RR-84-143. Marina Del Rey, California.: Information Sciences Institute.
- Partee, Barbara H. 1973. "Some Structural Analogies between Tense and Pronouns." *The Journal of Philosophy*, Vol. 70, No. 18.
- Passoneau, Rebecca. 1987. "Situations and Intervals." Proceedings of the ACL Conference, Stanford University, Stanford, California.
- Pereira, Fernando. 1983. "Logic for Natural Language Analysis." Technical Note 275. Menlo Park, California.: SRI International.
- Reichenbach, Hans. 1947. *Elements of Symbolic Logic*. New York, New York: Macmillan.
- Webber, Bonnie. 1987. "The Interpretation of Tense in Discourse." Proceedings of the ACL Conference, Stanford University, Stanford, California.
- Yip, Kenneth M. 1985. "Aspect and the Cognitive Representation of Time". Proceedings of the ACL Conference, University of Chicago, Chicago, Illinois.