Realizational morphology by disjunctive composition

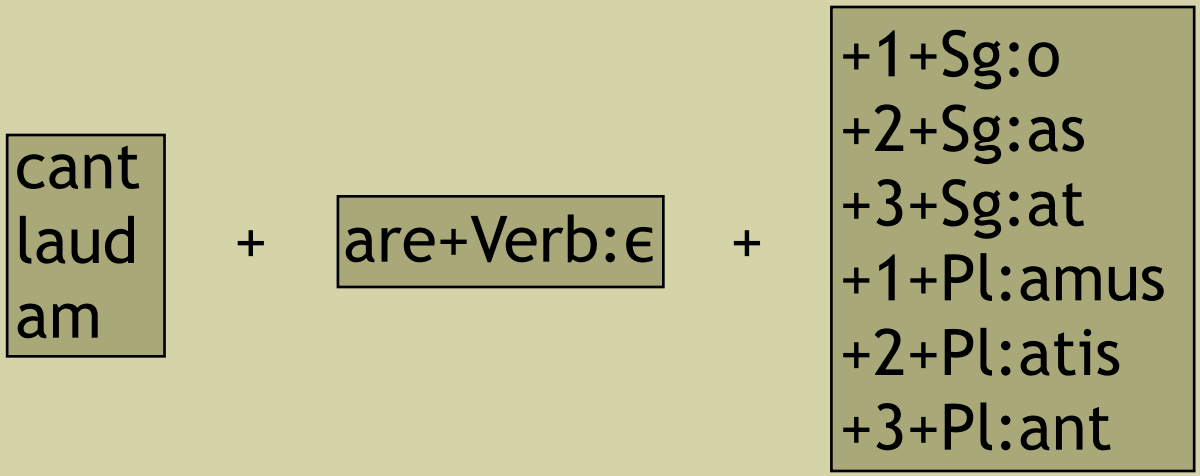
Robert Malouf
San Diego State University

Alpino Workshop
November 11, 2005

SAN DIEGO STATE UNIVERSITY

Finite state morphology

- Morphological and phonological rules can be expressed as finite state automata (Johnson 1972)
- Conventional finite state morphology (e.g., Beesley and Karttunen 2003)
- Finite state transducers (FSTs) for morphotactics:



Finite state morphology

- FSTs for morphophonology
- Compose cascade of FSTs into a single FST mapping between lexical and surface forms

  Lexical: *cantare+Verb+1+Sg*

  Surface: *canto*

- Finite state techniques provide an efficient and well-founded framework for implementation
- Familiar to linguists as ‘Item and Arrangement’ model of morphology

Finite state morphology

- One well-known problem with this model is the treatment of discontinuous dependencies within words
- E.g., Arabic case (Beesley and Karttunen 2003, 343)

kitaab+u	definite nominative
kitaab+a	definite accusative
kitaab+un	indefinite nominative
kitaab+an	indefinite accusative
al+kitaab+u	definite + definite
* *al+kitaab+un*	definite + indefinite

- Technical solutions (flag diacritics, etc.)

Word and Paradigm

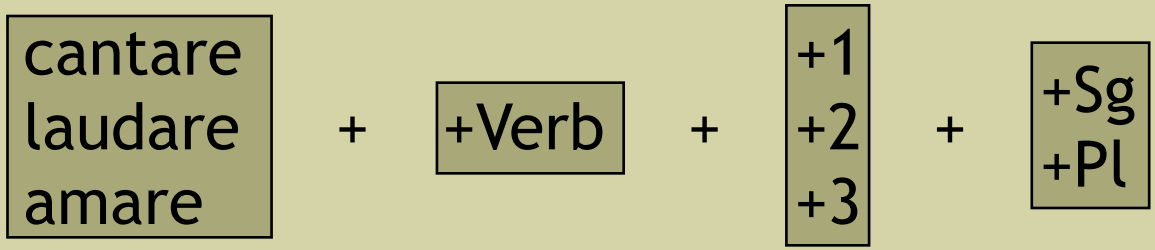
- Symptoms of a more general problem with morpheme-based models of morphology
- A solution to this and many other problems is offered by Word and Paradigm models of morphology (Robins, Matthews, Aronoff, Zwicky, et al.)
- In WP (aka *realizational* or *separationist*) models,
  - one set of rules build complete morphosyntactic representations
  - another set assigns a phonological form to a fully specified word
- Stump's (2001) Paradigm Function Morphology

Word and Paradigm

- This eliminates a technical problem with FSTs
- Many linguistic arguments that PFM provides a more natural analysis of complex inflectional systems
- As Karttunen (2003) shows, PFM finite state morphology requires only a slight change of perspective
  - Morphotactic rules construct lexical forms
  - Realizational rules build up the phonological realization of the given features
  - Final 'clean-up' transducer removes morphosyntactic features, leaving only the surface form

Word and Paradigm

- FSTs for morphotactics:



- Realizational rules:

If +*Verb*, delete -*are*

If +*1*+*Sg*, add -*o*
If +*2*+*Sg*, add -*as*
If +*3*+*Sg*, add -*at*
If +*1*+*Pl*, add -*amus*
. . .

Disjunctive rule ordering

- Karttunen's (2003) implementation combines realization rules via serial composition
- *Conjunctive* rule ordering: every rule applies to every form
- However, an important property of realization rules is that they are *disjunctively* ordered
  - Within each rule block, only the most specific applicable rule applies
  - More specific rules *block* the application of more general rules

Word and Paradigm

- PFM is not based on morphemes, but is still based on morphs
- PFM also requires abstract underlying representations (e.g., features to indicate verb conjugation)
- Blevins, Ackerman, and others have argued for a truly word-based morphology, drawing inspiration from traditional school grammars
  - Lexical entries list exemplary forms (principle parts)
  - Analogical rules form words from other words in the same paradigm

Word and Paradigm

- Representational ‘minimalism’
  - No declension features
  - No units smaller than a word
- Resolves many puzzles for morph(eme)-based derivations
  - Stem selection
  - Transderivational rules
  - Paradigm economy
- Parallels with construction grammar
- Psychological plausibility

Word and Paradigm

- "A correct principal-parts-and-paradigms statement and a correct morphophoneme-and-rule statement subsume the same actual facts of an alternation, the former more directly, the later more succinctly. We ought therefore to be free to use the latter, provided we specify that it is to be understood only as convenient shorthand for the former."

(Hockett 1967:222)

Modern Greek inflection

SINGULAR			PLURAL			
NOM	ACC	GEN	NOM	ACC	GEN	
nomos	*nomo*	*nomu*	*nomi*	*nomus*	*nomon*	‘law’
pateras	*patera*	*patera*	*pateres*	*pateres*	*pateron*	‘father’
papus	*papu*	*papu*	*papuðes*	*papuðes*	*papuðon*	‘grandfather’
imera	*imera*	*imeras*	*imeres*	*imeres*	*imeron*	‘day’
texni	*texni*	*texnis*	*texnes*	*texnes*	*texnon*	‘skill’
poli	*poli*	*poleos*	*poles*	*poles*	*poleon*	‘town’
maimu	*maimu*	*maimus*	*maimuðes*	*maimuðes*	*maimuðon*	‘monkey’

Haspelmath (2002, 125)

Modern Greek inflection

SINGULAR			PLURAL		
NOM	ACC	GEN	NOM	ACC	GEN
-os	*-o*	*-u*	*-i*	*-us*	*-on*
-as	*-a*		*-es*		*-on*
-us	*-u*		*-uðes*		*-uðon*
-a		*-as*	*-es*		*-on*
-i		*-is*	*-es*		*-on*
-i		*-eos*	*-es*		*-eon*
-u		*-us*	*-uðes*		*-uðon*

Modern Greek inflection

SINGULAR			PLURAL		
NOM	ACC	GEN	NOM	ACC	GEN
-Vs	*-V*	*-u*	*-i*	*-us*	*-on*
-Vs	*-V*		*-es*		*-on*
-Vs	*-V*		*-uðes*		*-uðon*
-V		*-Vs*	*-es*		*-on*
-V		*-Vs*	*-es*		*-on*
-V		*-eos*	*-es*		*-eon*
-V		*-Vs*	*-uðes*		*-uðon*

Modern Greek inflection

- Extensive syncretism (only the *-os/-u* declension has six distinct forms)
- The nom. sg. and gen. sg. forms are enough to know which declension a noun is in
- The acc. sg. is the same as whichever of the nom. sg. and gen. sg. doesn't end in *-s*
- Gen. pl. ending is *-on*, other plurals in *-es*
- Nouns in *-us/-u* or *-u/-us* have *-uð-* in plural
- General patterns all have exceptions

Lexicon

- Lexicon lists nom. sg. and gen. sg. forms of each word

```
macro(dictionary,{lex('nomos','nomu'),
                  lex('pateras','patera'),
                  lex('papus','papu'),
                  lex('imera','imeras'),
                  lex('texni','texnis'),
                  lex('poli','poleos'),
                  lex('maimu','maimus')}).
```

Disjunctive rule ordering

- Compiler for rule blocks
  - construct “narrower than” relation
  - transitive closure
  - disjunctive composition
- Implementation in prolog, using van Noord’s FSA Utilities

Realization rules

- A realization rule is pair (σ, φ), where the *structural description* σ is a recognizer and the *structural change* φ is a transducer
- A *rule block* is an unordered set of realization rules
- Singular paradigm rule block:

```
macro(block1,
block([rule(case='nom'&num='sg',first),
       rule(case='gen'&num='sg',second),
       rule(num='sg',
          {first,second} o ~suffix(s)),
       rule(case='acc'&num='sg'&first([o,s]),
          first o suffix([o,s]:[o]))
])).
```

Realization rules

- Within a block, for any two rules either:
  - they are **mutually inconsistent**

    Two rules (σ_1, φ_1) and (σ_2, φ_2) are mutually inconsistent iff $\sigma_1 \,\&\, \sigma_2 = \varnothing$

  - or one must be **narrower than** the other

    A rule (σ_1, φ_1) is narrower than a rule (σ_2, φ_2) iff $\sigma_1 - \sigma_2 = \varnothing$

- We check every pair of rules in a block and construct the "narrower than" partial order

Disjunctive composition

- Next, we modify the structural description of each rule so that it excludes words which narrower rules would apply to
- (σ_i, φ_i) becomes (σ'_i, φ_i), where $\sigma'_i = \sigma_i - \cup \sigma_j$ for all σ_j narrower than σ_i
- The result is a set of mutually exclusive realization rules, which can be combined via standard conjunction
- Cf. Erjavac's (1994) *ordered disjunction* and Karttunen's (1998) *priority union*

Disjunctive composition

- Disjunctively ordered rule blocks may be combined with each other via composition
- They also may be combined with conjunctively ordered rules, parallel rules, etc.
- A finite state model of WP inflection can be combined with conventional IA model of derivation
- Analogy

Conclusion

- Word and Paradigm morphology with disjunctively ordered realization rules fits comfortably within a finite state model of lexical processing
- This fact is of benefit to both theoretical linguistics and language engineers
  - Finite state models provide a robust, efficient platform to allow linguists to explore the consequences of their analyses
  - Realizational morphology provides an elaborated theoretical framework for language engineers to describe complex inflectional systems