Computational Constraints on Linguistic Descriptions

Anoop Sarkar
Computing Science, SFU
anoop@cs.sfu.ca

Life-cycle of a Linguist

Humans acquire linguistic rules

produce

Linguistic Behaviour

observed by

predictions

model

Concise description of the rules
Life-cycle of a Linguist (revised)

Humans acquire linguistic rules → predictions → Concise description of the rules

Linguistic Behaviour → observed by

Computational Constraints

Phonological Alternations

- Newton
- Paris
- practical
- feasible

- Newtonian
- Parisian
- impractical
- infeasible
Finnish Harmony

<table>
<thead>
<tr>
<th>Gloss</th>
<th>Nominative</th>
<th>Partitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>sky</td>
<td>taivas</td>
<td>taivas+ta</td>
</tr>
<tr>
<td>telephone</td>
<td>puhelin</td>
<td>puhelin+ta</td>
</tr>
<tr>
<td>plain</td>
<td>lakeus</td>
<td>lakeut+ta</td>
</tr>
<tr>
<td>reason</td>
<td>syy</td>
<td>syy+tä</td>
</tr>
<tr>
<td>short</td>
<td>lyhyt</td>
<td>lyhyt+tä</td>
</tr>
<tr>
<td>friendly</td>
<td>ystävällinen</td>
<td>ystävällinen+tä</td>
</tr>
</tbody>
</table>

`i,e are neutral wrt harmony`

talossansakaanko ‘not in his house either?’
kynässänsääänkö ‘not in his pen either?’

Rewrite Rules (Chomsky & Halle, 1968)

\[ a \rightarrow \ddot{a} / [\ddot{a},\ddot{o},y] C^* ([i,e] C^*)^* \]

\[ o \rightarrow \ddot{o} / [\ddot{a},\ddot{o},y] C^* ([i,e] C^*)^* \]

Articulatory Features

<table>
<thead>
<tr>
<th></th>
<th>high</th>
<th>low</th>
<th>back</th>
<th>round</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>−</td>
<td>+</td>
<td>+</td>
<td>−</td>
</tr>
<tr>
<td>o</td>
<td>−</td>
<td>−</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>u</td>
<td>+</td>
<td>−</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>ä</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>−</td>
</tr>
<tr>
<td>ö</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>y</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>i</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
</tr>
<tr>
<td>e</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
</tr>
</tbody>
</table>
Articulatory Features

Finnish Partitive
- taivas+tä
- puhelin+tä
- lakeut+tä
- syy+tä
- lyhyt+tä
- ystävällinen+tä

Rewrite Rule

\[ [V,+\text{back}] \rightarrow [V, -\text{back}] / [V, -\text{back}] \ C^* ([i,e] \ C^*)^* \]

Rewrite Rules

- Rewrite rules specify a relation between base-forms and observed forms, e.g. google+ing → googling
- Assume base-forms come from \((C^*(u|i))+ C^* (C^*V)^* C^*\)

\[ V \rightarrow i / i \ C^* \quad \text{kikV}kVkV \]
\[ V \rightarrow u / u \ C^* \quad \text{kikV}kV \text{kikik} kV \]

Output of one application feeds next application

\[ \text{left to right application} \]
Rewrite Rules

• Context dependent rewrite rules: \( \alpha \rightarrow \beta / \lambda \_ \__ \rho \)
  - \( (\lambda \_ \alpha \rho \rightarrow \lambda \_ \beta \rho) \); that is \( \alpha \) becomes \( \beta \) in context \( \lambda \_ \__ \rho \)

• How to apply rewrite rules:
  – Consider rewrite rule: \( a \rightarrow b / ab \_ \_ ba \)
  – Apply rule on string \( abababababa \)
  – Three different outcomes are possible:
    • \( abbbabbbbaba \) (left to right, iterative)
    • \( ababbbabba \) (right to left, iterative)
    • \( abb bbbbba \) (simultaneous)

Rewrite Rules

• Context dependent rewrite rules: \( \alpha \rightarrow \beta / \lambda \_ \__ \rho \)
• Can express **context sensitive** rules or **regular** relations
• Computational constraints on rewrite rules:
  – Consider rewrite rule: \( c \rightarrow acb / a \_ \_ b \)
  – Apply left to right iteratively on base-form \( c \)
  – Produces a sequence of strings:

<table>
<thead>
<tr>
<th>a</th>
<th>a</th>
<th>a</th>
<th>c</th>
<th>b</th>
<th>b</th>
<th>b</th>
</tr>
</thead>
</table>

Do we need such long-distance effects in morpho-phonological rules?
Computational Constraints on Rewrite Rules

- Rewrite rules express a context-sensitive grammar: \( \lambda \alpha \rho \rightarrow \lambda \beta \rho \), cf. \( \alpha \rightarrow \beta / \lambda \__ \rho \)
- CSGs are very powerful: they can generate languages like \{ 1^p : p is prime \}
- Kaplan and Kay:
  - Impose a simple constraint on how rewrite rules are applied: output cannot be re-written
    e.g. \( c \rightarrow acb / a __ b \)
  - Constraint ensures rewrite rules are equivalent to regular relations
  - Naturally expresses the local nature of morphophonemic properties

Finite state transducer
- each edge maps input:output
- defines a regular relation
Constraint-based approach

• Instead of explicit rules assume all variants are generated and surface constraints filter out illegal variants
  – Finnish: generate alternatives with both -ta and -tä
  – -ta is filtered out after [-back] vowels
  – -tä is filtered out after [+back] vowels
• Optimality Theory (Prince & Smolensky)
  – GEN produces all possible forms
  – rank-ordered violable constraints used to assign violations
  – form with least number of violations is produced

Constraint-based approach

• Karttunen:
  – No computational difference between traditional ordered rules and OT
  – Traditional ordered rules can be “compiled into” a regular relation with composed transducers
  – OT is “compiled into” a regular relation with leniently composed transducers
  – This notion of lenient composition captures the linguistic intuition: surface form has least violations
  – OT constraints must obey computational constraints (cf. Jason Eisner)
wh- questions in English

did Alice read which book

which book did Alice read 

has Leona said that Alice would read which book

which book has Leona said that Alice would read 

Mark asked to whom you had lent War & Peace

did Mark ask to whom you had lent what book

*what book did Mark ask to whom you had lent 

The Subjacency Condition

The non-local condition on "movement"

which book_i did [TP Alice read t_i]

which book_i has [TP Leona said [CP t_i that [TP Alice would read t_i]]]

*what book_i did [TP Mark ask to whom [TP you had lent t_i]]
Tree-Adjoining Grammars

• Construct a tree set out of tree fragments
• Each fragment contains only the structure needed to express the locality of various CSG predicates
• Each tree fragment is called an elementary tree
• In general we need to expand even those non-terminals that are not leaf nodes: leads to the notion of adjunction

TAG Analysis

(Kroch, Frank)
Alice would read which book?

Leona said that Tj has that Alice would read.
The text and diagram show TAG Analysis with a focus on the sentence: *what book i did Mark ask to whom you had lent t i*

The diagram illustrates the syntactic structure of the sentence, with tags and phrases indicating the grammatical roles of each word. The analysis highlights the local wff condition on wh-questions, adjunction handling long-distance dependency, the absence of intermediate traces, and a similar insight in GPSG, CCG, but not in HPSG.
TAG Analysis

*what book*$_i$ did Mark ask to whom you had lent *$_i$

- Kroch: such a tree is not well formed in English even in a single clause
- e.g. *I wonder what book to whom Mark gave
- Linguistic rules are used to only construct well formed elementary trees
- Computational model enforces local constraints

What about languages that allow multiple wh-fronting? e.g. Romanian

Summary

- Linguistic descriptions have computational properties: being sensitive to these properties provides insight
- Computational complexity explains the existence of locality in applicability of linguistic rules
- It is interesting from a computational viewpoint exactly how complex are the generalizations produced by linguists
- The computational treatment allows the “compilation” of linguistic rules into automata: a constructive proof of their efficacy