# Computational Aspects of Metrical Stress in OT

*Tamás Bíró*

*Alfa-Informatica, RUG*

`birot@let.rug.nl`

June 20, 2003

# Overview

- What is a finite state transducer (FST)?
- Finite state transducers and regular grammars
- OT as a FST:
  - Is Gen a FST?
  - Are constraints FSTs?

# What is a finite state transducer?

A mapping:


- A finite set of states
- A finite set of transition rules:
  $(actual\ state, input) \longmapsto (new\ state, output)$

Grammatical:

$$\left\{\begin{array}{c}\textit{Beer!}\\ \textit{Here you are!}\end{array}\right\}\left\{\begin{array}{c}\textit{Beer!}\\ \textit{Am I a servant?}\end{array}\right\}\left\{\begin{array}{c}\textit{I love you!}\\ \textit{Do you?}\end{array}\right\}$$

$$\left\{\begin{array}{c}\textit{Beer!}\\ \textit{That's not nice}\end{array}\right\}\left\{\begin{array}{c}\textit{Beer!}\\ \textit{That's not nice}\end{array}\right\}\left\{\begin{array}{c}\textit{I love you}\\ \textit{So do I!}\end{array}\right\}$$

Agrammatical:

$$*\left\{\begin{array}{c}\textit{Beer!}\\ \textit{Here you are!}\end{array}\right\}\left\{\begin{array}{c}\textit{I love you!}\\ \textit{Do you?}\end{array}\right\}\left\{\begin{array}{c}\textit{I love you!}\\ \textit{I don't!}\end{array}\right\}$$

# Finite state transducers and regular grammars

- Finite state transducers
- Regular grammars
- Regular expressions

have the same generative power

Remember: regular $\subset$ context-free $\subset$ context-sensitive

$$\textbf{lover} \longrightarrow \left\{ \begin{array}{c} \textit{Beer!} \\ \textit{Here you are, my dear.} \end{array} \right\} \textbf{lover}$$

$$\textbf{lover} \longrightarrow \left\{ \begin{array}{c} \textit{Beer!} \\ \textit{Here your are!} \end{array} \right\} \textbf{avarage}$$

$$\textbf{very angry} \longrightarrow \left\{ \begin{array}{c} \textit{I love you!} \\ \textit{I don't.} \end{array} \right\} \textbf{angry}$$

# Finite state transducers as language models:

Why usually men fail if they apply this model?

What is the problem with this model?

no long-term memory !!!

- This is a very strong restriction on the model.
- Can you describe human language with such a restricted model?

# Phonology as a finite state transducer

SPE 1968: context sensitive rules (too powerful)

Johnson 1972, Koskenniemi 1983, Kaplan and Kay 1994, etc:

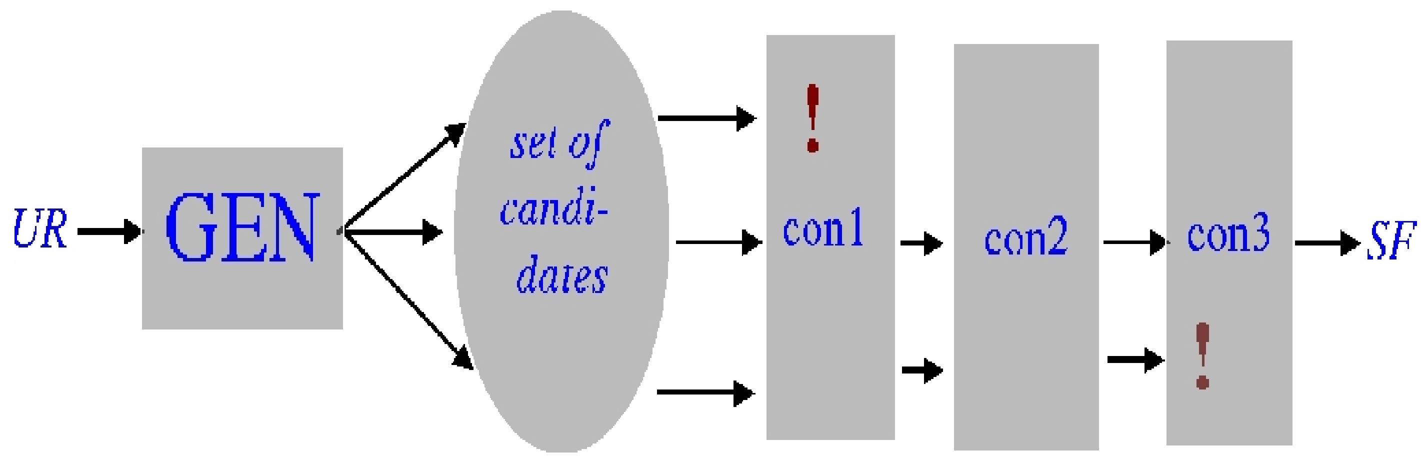
most of phonology has a generative power of a regular language.

Prince & Smolensky 1993:

- Is OT an adequate model for phonology?

# OT as a finite state transducer

- If yes, can one implement it as an FST?




- Implement Gen as an FST
- Implement the constraints as FSTs

# Implement Gen a finite state transducer?

Well... Which Gen? Say: metrical stress.

$$word =$$

$$\# \left| \left\{ \begin{array}{c} unprsd\ syl \\ n\text{-}hd\text{-}ft \end{array} \right\}^* \right| \; hd\text{-}ft \; \left| \left\{ \begin{array}{c} unprsd\ syl \\ n\text{-}hd\text{-}ft \end{array} \right\}^* \right| \#$$

# Implement Gen a FST? (cont'd)

$$unprsd\ syl = phonemes^*|.$$

$$n\text{-}hd\text{-}ft =$$

$$\left\{ \begin{array}{c} phonemes^*|\mathit{2}|. \\ phonemes^*|\mathit{2}|.|phonemes^*|. \\ phonemes^*|.|phonemes^*|\mathit{2}|. \end{array} \right\}$$

# Implement Gen a FST? (cont'd)

$$hd\text{-}ft =$$

$$\left\{ \begin{array}{c} phonemes^{*}|1|. \\ phonemes^{*}|1|.|phonemes^{*}|. \\ phonemes^{*}|.|phonemes^{*}|1|. \end{array} \right\}$$

- Transform regular expressions to FST

$$
\begin{array}{c}
\#\ \text{ab.ra.ka.dab.ra.}\# \\
\downarrow \\
\boxed{\text{FST in state } S} \\
\downarrow \\
\#\ \text{ab.}\{\text{ra.ka1}\}.[\text{dab2.ra}].\#
\end{array}
$$

# Are constraints finite state transducers?

Depends...

# Are constraints finite state transducers? (cont'd)

A typology for constraints:

The maximal number of violation marks that a candidate can be assigned is:

1. 1 (or: constant in the length of the word)
2. proportional to the length of the word
3. growing faster than the length of the word

# Are constraints finite state transducers? (cont'd)

Case 1: Max. 1 (constant) violation mark for each candidate. Example:

- ALIGN(Word,Foot,Left): align the left edge of the word with the left edge of some foot.

Easy to realize with finite state techniques.
(Frank and Satta 1998, Karttunen 1998).

Remark:

Max. 1 violation mark, but not Finite State-friendly constraints (not possible to assign violation marks):

MatchesOutputOfSPE: The output matches the result of applying Chomsky & Halle (1968) to the input. (J. Eisner, 1999)

Cf. OTP : “OT with primitive constraints” by J. Eisner.

# Are constraints finite state transducers? (cont'd)

Case 2: Number of violation marks proportional to the length of the word

Case 2a: Violation marks align nicely:

- ALIGN(Main-foot,Word,Left): align head-foot with word, left edge. $\sigma * \sigma * \sigma * [\sigma\ \sigma 1]\ \sigma\ \sigma$

Possible to realize using finite state techniques.

(Gerdemann and van Noord 2000, Bíró 2003)

# Are constraints finite state transducers? (cont'd)

Case 2b: 1 (constant) violation per locus, but anywhere. Examples:

- Parse-syllable: each syllable must be footed.
- Iambic: align the right edge of each foot with its head syllable.

Easy to assign the violation marks, but
hard to filter out the non-harmonic candidates.

# Are constraints finite state transducers? (cont'd)

Case 3: Number of violation marks growing faster than the lengths of the string. Example:

- ALIGN(Foot,Word,Left): align each foot with the word, left edge.

(Usually) not possible even to write a transducer that would assign the violation marks. (Bíró 2003, cf. J. Eisner's remarks)

# Conclusions

Message for phonologists:

- OT's power can be close to the very restricted class of regular languages,
- if you don't use certain constraints,
- such as gradient constraints.
- Cf. McCarthy's recent arguments against them.

# Conclusions

Hypotheses underlying OT (explicit in McCarthy 2002):

- *Locus hypothesis*: A violation mark is assigned for each *locus* of violation within a candidate.
- *Gradience hypothesis*: Some constraints are gradient: multiple violations to a single locus.
- *Homogeneity hypothesis*: Multiple violations of a constraint from either source are added together in evaluating a candidate.

# Conclusions

McCarthy: no need for gradient constraints. Reformulate them or throw them!

Gradient constraints that cannot be reformulated:

- "harmful" according to McCarthy (2002),
- impossible for a finite state approach
  (too strong generative power needed)

# Conclusions

{ *Beer!* *I love you!*}

Otherwise
you can be
optimistic about
a harmonic marriage
of OT and FST.