Being stressed by stress

Competence, performance and learning of metrical stress in Optimality Theory

Tamás Biró

ACLC, University of Amsterdam (UvA)

BGU, March 13, 2012

How to observe stress pattern in (Dutch) fast speech?

Quiz item:

Q4	*President Bush is een typische*	'President Bush is a typical'
A1	*intellectueel*	'intellectual'
A2	*amerikaan*	'American'
A3	*taalkundige*	'linguist'

Maartje Schreuder and Dicky Gilbers (2004). 'The Influence of Speech Rate on Rhythm Patterns'. In: *On the Boundaries of Phonology and Phonetics*. Groningen.

How to observe stress pattern in (Dutch) fast speech?

Quiz item:

Q4	*President Bush is een typische*	‘President Bush is a typical’
A1	*intellectueel*	‘intellectual’
A2	*amerikaan*	‘American’
A3	*taalkundige*	‘linguist’

Maartje Schreuder and Dicky Gilbers (2004). ‘The Influence of Speech Rate on Rhythm Patterns’. In: *On the Boundaries of Phonology and Phonetics*. Groningen.

How to observe stress pattern in (Dutch) fast speech?

Forms and frequencies observed by Schreuder and Gilbers:

fo.to.toe.stel 'camera'	*uit.ge.ve.rij* 'publisher'	*stu.die.toe.la.ge* 'study grant'	*per.fec.tio.nist* 'perfectionist'
susu	ssus	susuu	usus
fó.to.tòe.stel fast: slow:	*ùit.gè.ve.ríj* fast: **0.67** slow: **0.96**	*stú.die.tòe.la.ge* fast: **0.38** slow: **0.81**	*per.fèc.tio.níst* fast: **0.13** slow: **0.20**
fó.to.toe.stèl fast: slow:	*ùit.ge.ve.ríj* fast: **0.33** slow: **0.04**	*stú.die.toe.là.ge* fast: **0.62** slow: **0.19**	*pèr.fec.tio.níst* fast: **0.87** slow: **0.80**
shift to right	beat reduction	shift to right	shift to left

Stressing questions

- Account for stress patterns within some linguistic framework.
- Account for changing patterns: normal *vs.* fast speech.
  - Account for fast speech forms. Account for their frequencies.
  - Both forms present. Different word types have different frequencies.
  - Gradual change between 'normal' and 'fast'.
- Account for language acquisition:
  - Despite difficulties caused by framework (*viz.* feet).
  - (Despite difficulties caused by speech errors.)
- (Account for language change: iterated learning.)

Stressing questions

- Account for stress patterns within some linguistic framework.
- Account for changing patterns: normal *vs.* fast speech.
  - Account for fast speech forms. Account for their frequencies.
  - Both forms present. Different word types have different frequencies.
  - Gradual change between 'normal' and 'fast'.
- Account for language acquisition:
  - Despite difficulties caused by framework (*viz.* feet).
  - (Despite difficulties caused by speech errors.)
- (Account for language change: iterated learning.)

Stressing questions

- Account for stress patterns within some linguistic framework.
- Account for changing patterns: normal *vs.* fast speech.
    - Account for fast speech forms. Account for their frequencies.
    - Both forms present. Different word types have different frequencies.
    - Gradual change between ‘normal’ and ‘fast’.
- Account for language acquisition:
    - Despite difficulties caused by framework (*viz.* feet).
    - (Despite difficulties caused by speech errors.)
- (Account for language change: iterated learning.)

Stressing questions

- Account for stress patterns within some linguistic framework.
- Account for changing patterns: normal *vs.* fast speech.
    - Account for fast speech forms. Account for their frequencies.
    - Both forms present. Different word types have different frequencies.
    - Gradual change between ‘normal’ and ‘fast’.
- Account for language acquisition:
    - Despite difficulties caused by framework (*viz.* feet).
    - (Despite difficulties caused by speech errors.)
- (Account for language change: iterated learning.)

Overview

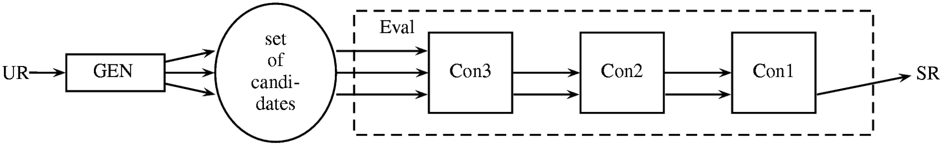
1. Optimality Theory: competence
2. Simulated Annealing for Optimality Theory: performance
3. Learning in Optimality Theory: problems
4. Conclusions

Overview

1. Optimality Theory: competence
2. Simulated Annealing for Optimality Theory: performance
3. Learning in Optimality Theory: problems
4. Conclusions

Traditional Optimality Theory

- Underlying form $\mapsto$ a set of candidates (potential surface forms).
- Constraints: elementary functions defined on the candidate set.
- Grammatical form predicted by this grammar:
  candidate that violates the highest ranked constraints the least.
- Serial evaluation approach:



Traditional Optimality Theory

- Underlying form $\mapsto$ a set of candidates (potential surface forms).
- Constraints: elementary functions defined on the candidate set.
- Grammatical form predicted by this grammar:
  candidate that violates the highest ranked constraints the least.

Grammar = constraint hierarchy

- Optimality Theory is a P&P-style model of competence.
- Constraints are universal "principles".
- Constraint ranking is language-dependent "parameter".

Traditional Optimality Theory

Prosodic phonology:

$$\Big[_{\mathrm{IP}}\ \Big[_{\mathrm{Wd}}\ \sigma\ [\grave{\sigma}\sigma]\ [\grave{\sigma}]\ \sigma\ [\sigma\acute{\sigma}]\ \Big]\ \Big]$$

- Intonational Phrase $\rightarrow$ Prosodic Word $\rightarrow$ Foot $\rightarrow$ Syllable $\rightarrow$ ...
- Stressed syllable: head (‘strong’) syllable of a word.
- Main stress: head syllable of head (‘strong’) foot.

Traditional Optimality Theory

Proposal of Schreuder and Gilbers:

Andante speech:

/*foto+toestel*/	OUTPUT-OUTPUT CORRESPONDENCE	FOOT REPULSION	PARSE SYLLABLE
☞ (fóto)(tòestel)		*	
(fóto)toe(s̀tel)	*!		*

NB: Further output forms in the language: *fóto* and *tóestel*.

Traditional Optimality Theory

Proposal of Schreuder and Gilbers:

Allegro speech:

/*foto+toestel*/	FOOT REPULSION	OUTPUT-OUTPUT CORRESPONDENCE	PARSE SYLLABLE
(fóto)(tòestel)	*!		
☞ (fóto)toe(s̀tel)		*	*

NB: Further output forms in the language: *fóto* and *tóestel*.

Traditional Optimality Theory

Grammar = constraint hierarchy

- Different grammar for fast speech.
  Hence, different competence in fast speech?
- How to account for both forms produced?
  - frequencies depending on speech rate, and
  - frequencies depending on word.

Constraint re-ranking

- Ad hoc re-ranking, motivated by analogy in music.
- Principled framework: Boersma's *Stochastic Optimality Theory*.

Traditional Optimality Theory

Grammar = constraint hierarchy

- Different grammar for fast speech.
  Hence, different competence in fast speech?
- How to account for both forms produced?
  - frequencies depending on speech rate, and
  - frequencies depending on word.

Constraint re-ranking

- Ad hoc re-ranking, motivated by analogy in music.
- Principled framework: Boersma's *Stochastic Optimality Theory*.

Traditional Optimality Theory

Grammar = constraint hierarchy

- Different grammar for fast speech.
  Hence, different competence in fast speech?
- How to account for both forms produced?
  - frequencies depending on speech rate, and
  - frequencies depending on word.

Constraint re-ranking

- Ad hoc re-ranking, motivated by analogy in music.
- Principled framework: Boersma's *Stochastic Optimality Theory*.

Traditional Optimality Theory

Grammar = constraint hierarchy

- Different grammar for fast speech.
  Hence, different competence in fast speech?
- How to account for both forms produced?
  - frequencies depending on speech rate, and
  - frequencies depending on word.

Constraint re-ranking

- Ad hoc re-ranking, motivated by analogy in music.
- Principled framework: Boersma's *Stochastic Optimality Theory*.

Overview

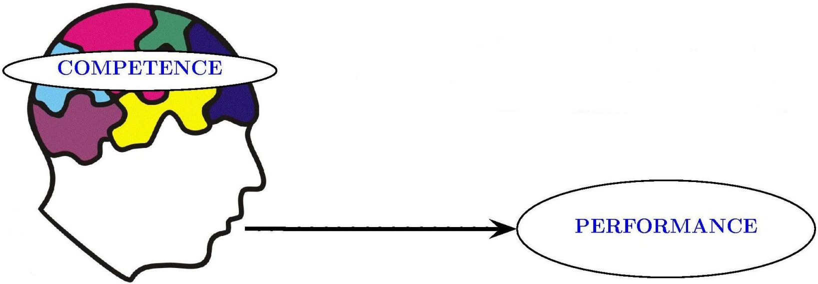
1 Optimality Theory: competence

2 Simulated Annealing for Optimality Theory: performance

3 Learning in Optimality Theory: problems

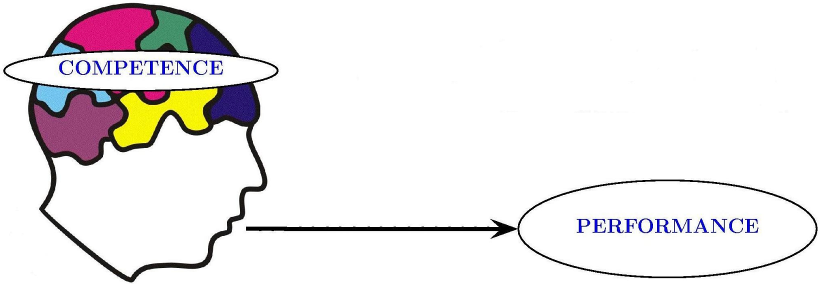
4 Conclusions

Errors of the mental computation



static knowledge | processing in the brain
Optimality Theory | Simulated Annealing for OT

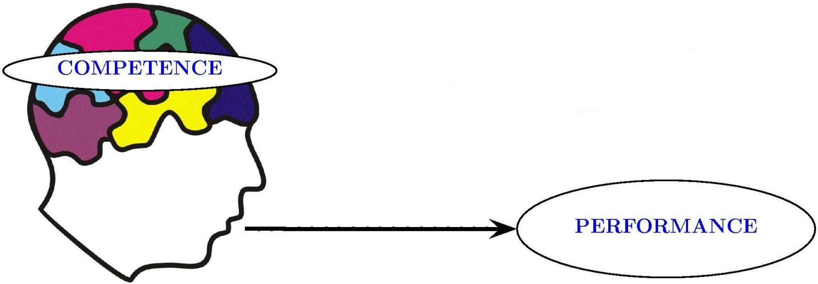
Errors of the mental computation



static knowledge processing in the brain

Optimality Theory Simulated Annealing for OT

Errors of the mental computation

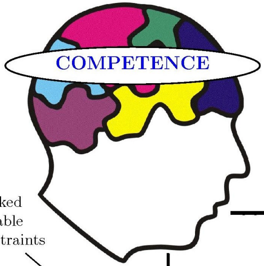


static knowledge — processing in the brain

Optimality Theory — Simulated Annealing for OT

Errors of the mental computation

A grammar is a Harmony function on the candidate set, defined by the ranked constraints.
Global optimum: more harmonic than all other candidates.
Local optimum: more harmonic than its neighbours.

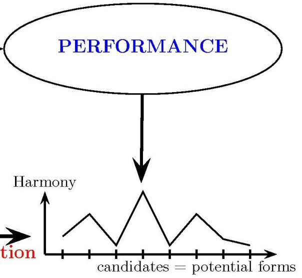


Ranked violable constraints

Candidates (potential forms)

/aat/	NoCoda	Parse	Onset
☞ [a.a<t>]		*	**
[a.at]	*!		**
[<a>at]	*!	*	*

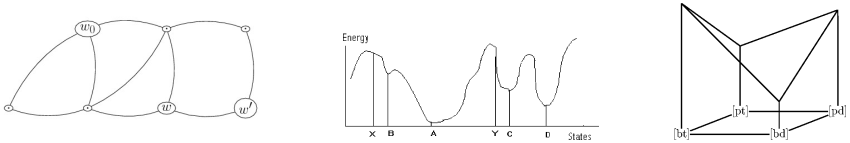
implementation



Optimality Theory
grammar
competence model
grammatical form = ☞ (globally) optimal candidate

SA-OT
implementation
performance model
produced forms = globally or locally optimal candidates

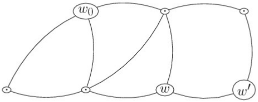
Basic idea of Simulated Annealing

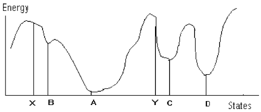


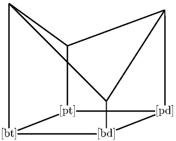
Step 1 – introducing landscape:

- Horizontal: universal *neighbourhood structure* (a.k.a. *topology*) on the universal candidate set.
- Vertical: grammar-dependent harmony (violation profile of the constraints).
- Random walk in this landscape.

Basic idea of Simulated Annealing



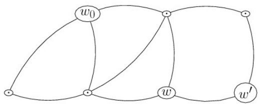


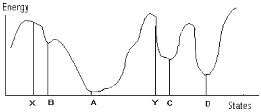


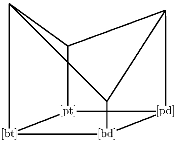
Step 2 – walking in this landscape:

- Pick a random neighbour of your position.
- If neighbour is more optimal: move.
- If less optimal: move in the beginning, don't move later.
  (Exponential expression applied to vector-valued target function.)

Basic idea of Simulated Annealing



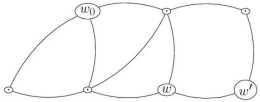


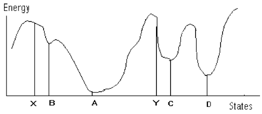


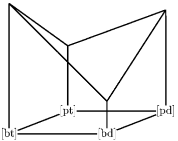
Step 3 – performing a random walk on this landscape:

- Start random walk from some initial position.
- End position returned as output of the algorithm: produced form.
- Hopefully, global optimum (grammatical form) is found. Yet...
- Neighbourhood structure → local optima, where random walker can get stuck. Performance errors.

Basic idea of Simulated Annealing







Step 4 – Precision of the algorithm

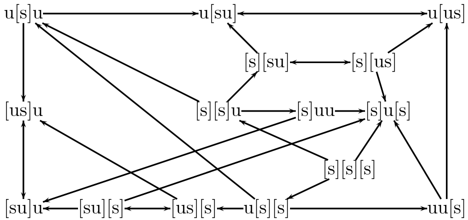
- **Precision** of the algorithm: chance of ending up in global optimum, and hence returning grammatical form.
- Precision of the algorithm depends on its speed.
- Trade precision for speed – just like human mind!

Basic idea of Simulated Annealing

Level	its product	its model	the product in the model
Competence in narrow sense: static knowledge of the language	grammatical form	standard OT grammar	globally optimal candidate
Dynamic language production process	acceptable or attested forms	SA-OT algorithm	local optima
Performance in its outmost sense	acoustic signal, etc.	(phonetics, pragmatics)	??

How to predict stress pattern in (Dutch) fast speech?

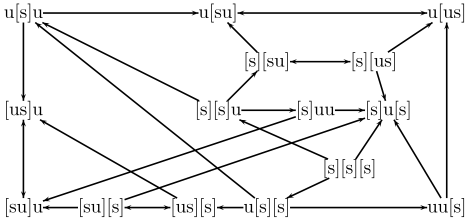
Landscape:



Basic steps that connect neighbours:

- Move foot boundary: [s] ⇔ [su]; [s] ⇔ [us].
- Change head of foot: [su] ⇔ [us].
- Insert/delete monosyllabic foot: [s] ⇔ u

How to predict stress pattern in (Dutch) fast speech?



Hierarchy:

ALIGN-LEFT $\gg$ $\text{OOC}_{z=2}$ $\gg$ FOOTREPULSION $\gg$ PARSESYLL $\gg$ TROCHAIC

- Global optimum: [s]u[su].
- Local optima: [s]u[su] and [su]u[s].
- Local optimum [su]u[s] has less harmonic neighbours: [su]uu, [su][s][s], [us]u[s], [s]uu[s], [su][us].

How to predict stress pattern in (Dutch) fast speech?

Local optimum [su]u[s] has less harmonic neighbours:
[su]uu, [su][s][s], [us]u[s], [s]uu[s], [su][us].

/fototoestel/	ALIGN-LEFT	$\text{OOC}_{z=2}$	FTREPULS	PARSES	TROCHAIC
∼ [su]u[s]	0	2	0	1	0
[su]uu	0	3	0	2	0
[su][s][s]	0	3	2	0	0
[us]u[s]	0	4	0	1	1
[s]uu[s]	0	2	0	2	0
[su][us]	0	2	1	0	1

How to predict stress pattern in (Dutch) fast speech?

`http://www.birot.hu/sa-ot/`

`http://www.birot.hu/OTKit/`

How to predict stress pattern in (Dutch) fast speech?

fo.to.toe.stel ‘camera’	*uit.ge.ve.rij* ‘publisher’	*stu.die.toe.la.ge* ‘study grant’	*per.fec.tio.nist* ‘perfectionist’
susu	ssus	susuu	usus
fó.to.tòe.stel fast: *0.82* slow: *1.00*	*ùit.gè.ve.ríj* fast: *0.65* / **0.67** slow: *0.97* / **0.96**	*stú.die.tòe.la.ge* fast: *0.55* / **0.38** slow: *0.96* / **0.81**	*per.fèc.tio.níst* fast: *0.49* / **0.13** slow: *0.91* / **0.20**
fó.to.toe.stèl fast: *0.18* slow: *0.00*	*ùit.ge.ve.ríj* fast: *0.35* / **0.33** slow: *0.03* / **0.04**	*stú.die.toe.là.ge* fast: *0.45* / **0.62** slow: *0.04* / **0.19**	*pèr.fec.tio.níst* fast: *0.39* / **0.87** slow: *0.07* / **0.80**

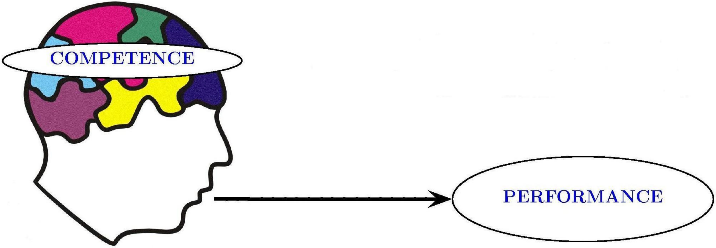
Simulated / **observed** (Schreuder) frequencies.

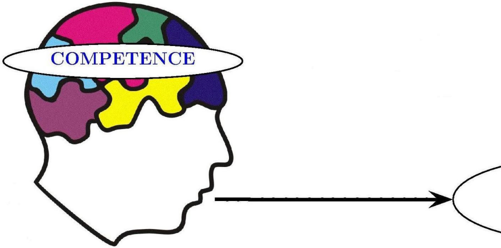
In the simulations, $T_{step} = 3$ used for fast speech and $T_{step} = 0.1$ for slow speech.

Overview

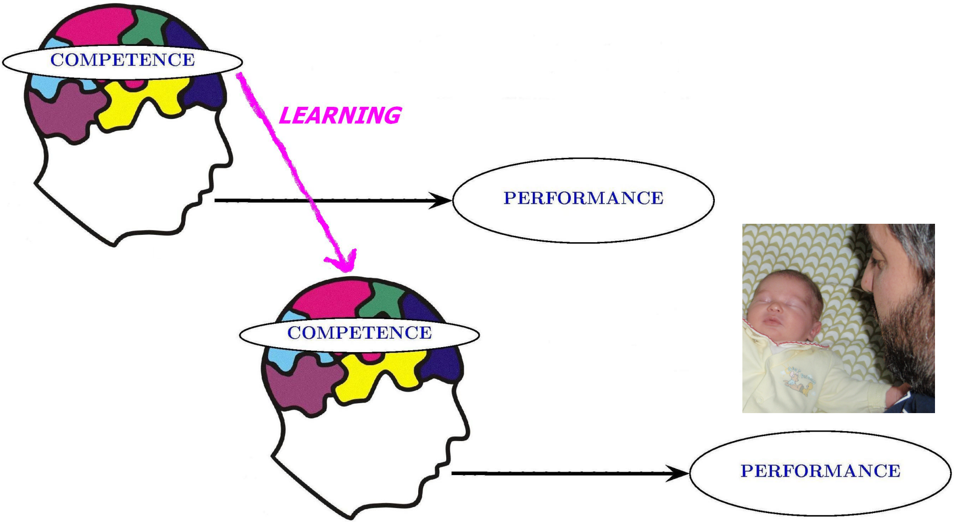
1. Optimality Theory: competence
2. Simulated Annealing for Optimality Theory: performance
3. Learning in Optimality Theory: problems
4. Conclusions

The language acquisition problem

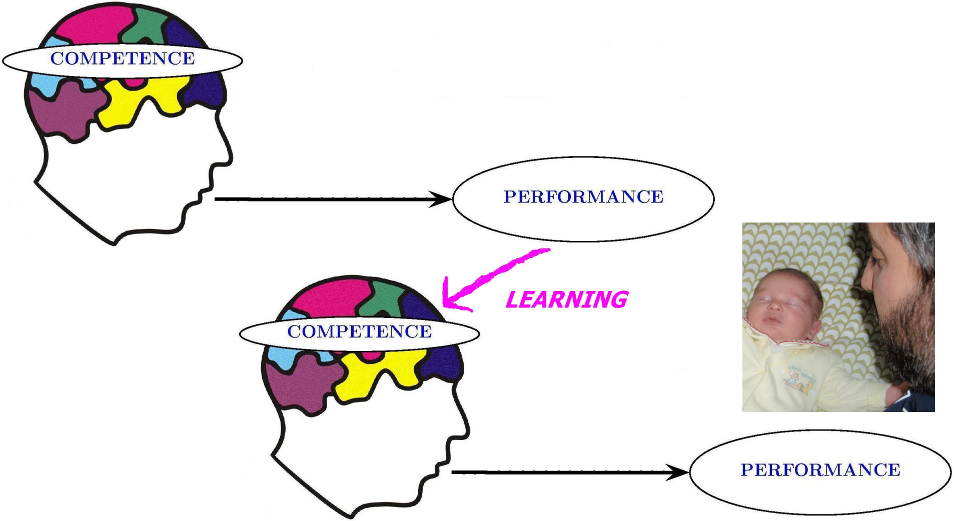




Learning from competence?



Learning from performance!



Learning in Optimality Theory

General idea:

- Speaker-teacher wants to say *underlying form* uf.
- Speaker-teacher's grammar produces *surface form* sf.
- Listener-learner hears *surface form* sf = *winner form* w.
- Listener-learner's grammar would produce uf as *loser form* l.
- Listener-learner updates her grammar,
  in order to produce w, and not l:

  Winner-preferring constraints are promoted and loser-preferring constraints are demoted in hierarchy hypothesized by the learner.

Learning hidden structures (e.g., metrical feet)

But foot boundaries are not pronounced:

- Speaker-teacher wants to say: *ab.ra.ka.dab.ra* (underlying form).
- Speaker-teacher's grammar produces:
  [àb.ra].ka.[dáb.ra] (surface form).
- Speaker-teacher utters: *àb.ra.ka.dáb.ra* (overt form).
- Listener-learner hears: *àb.ra.ka.dáb.ra* (overt form).
- Listener-learner hesitates:
  what is the grammatical surface form in the target languages?

[àb].ra.ka.[dáb].ra, *[àb.ra].ka.[dáb].ra*
[àb].ra.[ka.dáb].ra, *[àb].ra.ka.[dáb.ra]*, etc.?

Learning hidden structures (e.g., metrical feet)

But foot boundaries are not pronounced:

- Speaker-teacher wants to say: *ab.ra.ka.dab.ra* (underlying form).
- Speaker-teacher's grammar produces:
  [àb.ra].ka.[dáb.ra] (surface form).
- Speaker-teacher utters: *àb.ra.ka.dáb.ra* (overt form).
- Listener-learner hears: *àb.ra.ka.dáb.ra* (overt form).
- Listener-learner hesitates:
  what is the grammatical surface form in the target languages?

  [àb].ra.ka.[dáb].ra, *[àb.ra].ka.[dáb].ra*
  [àb].ra.[ka.dáb].ra, *[àb].ra.ka.[dáb.ra]*, etc.?

Learning hidden structures (e.g., metrical feet)

But foot boundaries are not pronounced:

- Speaker-teacher wants to say: *ab.ra.ka.dab.ra* (underlying form).
- Speaker-teacher's grammar produces:
  [àb.ra].ka.[dáb.ra] (surface form).
- Speaker-teacher utters: *àb.ra.ka.dáb.ra* (overt form).
- Listener-learner hears: *àb.ra.ka.dáb.ra* (overt form).
- Listener-learner hesitates:
  what is the grammatical surface form in the target languages?

  [àb].ra.ka.[dáb].ra, *[àb.ra].ka.[dáb].ra*
  [àb].ra.[ka.dáb].ra, *[àb].ra.ka.[dáb.ra]*, etc.?

Robust Interpretive Parsing (Tesar and Smolensky)

		/ab.ra.ka.dab.ra/	NonFinal	Trochaic	FootRepulsion
l	1.	*ab.ra.[ka.dáb].ra*		*	
w	2.	*[àb.ra].[ka.dáb].ra*		*	*
☞	3.	*[àb.ra].ka.[dáb.ra]*	*		

- Teacher: FootRepulsion ≫ Trochaic ≫ NonFinal, producing grammatical form ☞: *[àb.ra].ka.[dáb.ra]*.
- Learner: NonFinal ≫ Trochaic ≫ FootRepulsion, producing loser: *ab.ra.[ka.dáb].ra*.
- Learner hears *àb.ra.ka.dáb.ra*. Two possible candidates. The winner must have been, the best one, *[àb.ra].[ka.dáb].ra*.
- Compare w and l. Promote w-preferring constraints: none. Demote l-preferring constraints: FootRepulsion → deadlock!

Robust Interpretive Parsing (Tesar and Smolensky)

		/ab.ra.ka.dab.ra/	NONFINAL	TROCHAIC	FOOTREPULSION
l	1.	*ab.ra.[ka.dáb].ra*		*	
w	2.	*[àb.ra].[ka.dáb].ra*		*	*
☞	3.	*[àb.ra].ka.[dáb.ra]*	*		

- Teacher: FOOTREPULSION $\gg$ TROCHAIC $\gg$ NONFINAL, producing grammatical form ☞: *[àb.ra].ka.[dáb.ra]*.
- Learner: NONFINAL $\gg$ TROCHAIC $\gg$ FOOTREPULSION, producing loser: *ab.ra.[ka.dáb].ra*.
- Learner hears *àb.ra.ka.dáb.ra*. Two possible candidates. The winner must have been, the best one, *[àb.ra].[ka.dáb].ra*.
- Compare w and l. Promote w-preferring constraints: none. Demote l-preferring constraints: FOOTREPULSION $\rightarrow$ deadlock!

Robust Interpretive Parsing (Tesar and Smolensky)

		/ab.ra.ka.dab.ra/	NONFINAL	TROCHAIC	FOOTREPULSION
l	1.	*ab.ra.[ka.dáb].ra*		*	
w	2.	*[àb.ra].[ka.dáb].ra*		*	*
☞	3.	*[àb.ra].ka.[dáb.ra]*	*		

- Teacher: FOOTREPULSION $\gg$ TROCHAIC $\gg$ NONFINAL, producing grammatical form ☞: *[àb.ra].ka.[dáb.ra]*.
- Learner: NONFINAL $\gg$ TROCHAIC $\gg$ FOOTREPULSION, producing loser: *ab.ra.[ka.dáb].ra*.
- Learner hears *àb.ra.ka.dáb.ra*. Two possible candidates. The winner must have been, the best one, *[àb.ra].[ka.dáb].ra*.
- Compare w and l. Promote w-preferring constraints: none. Demote l-preferring constraints: FOOTREPULSION → deadlock!

Robust Interpretive Parsing (Tesar and Smolensky)

		/ab.ra.ka.dab.ra/	NONFINAL	TROCHAIC	FOOTREPULSION
l	1.	*ab.ra.[ka.dáb].ra*		*	
w	2.	*[àb.ra].[ka.dáb].ra*		*	*
☞	3.	*[àb.ra].ka.[dáb.ra]*	*		

- Teacher: FOOTREPULSION ≫ TROCHAIC ≫ NONFINAL, producing grammatical form ☞: *[àb.ra].ka.[dáb.ra]*.
- Learner: NONFINAL ≫ TROCHAIC ≫ FOOTREPULSION, producing loser: *ab.ra.[ka.dáb].ra*.
- Learner hears *àb.ra.ka.dáb.ra*. Two possible candidates. The winner must have been, the best one, *[àb.ra].[ka.dáb].ra*.
- Compare w and l. Promote w-preferring constraints: none. Demote l-preferring constraints: FOOTREPULSION → deadlock!

Robust Interpretive Parsing (Tesar and Smolensky)

		/ab.ra.ka.dab.ra/	NonFinal	Trochaic	FootRepulsion
l	1.	*ab.ra.[ka.dáb].ra*		*	
w	2.	*[àb.ra].[ka.dáb].ra*		*	*
☞	3.	*[àb.ra].ka.[dáb.ra]*	*		

- Teacher: FootRepulsion ≫ Trochaic ≫ NonFinal, producing grammatical form ☞: *[àb.ra].ka.[dáb.ra]*.
- Learner: NonFinal ≫ Trochaic ≫ FootRepulsion, producing loser: *ab.ra.[ka.dáb].ra*.
- Learner hears *àb.ra.ka.dáb.ra*. Two possible candidates. The winner must have been, the best one, *[àb.ra].[ka.dáb].ra*.
- Compare w and l. Promote w-preferring constraints: none. Demote l-preferring constraints: FootRepulsion → deadlock!

Revised *Robust Interpretive Parsing* (Biró, under review)

		/ab.ra.ka.dab.ra/	NONFINAL	TROCHAIC	FOOTREPULSION
l	1.	*ab.ra.[ka.dáb].ra*		*	
	2.	*[àb.ra].[ka.dáb].ra*		*	*
☞	3.	*[àb.ra].ka.[dáb.ra]*	*		
w			0.5	0.5	0.5

- Teacher: FOOTREPULSION ≫ TROCHAIC ≫ NONFINAL, → ☞*[àb.ra].ka.[dáb.ra]*.
- Learner: NONFINAL ≫ TROCHAIC ≫ FOOTREPULSION, → l *ab.ra.[ka.dáb].ra*.
- Learner: w is either *[àb.ra].[ka.dáb].ra* or *[àb.ra].ka.[dáb.ra]*.
- Calculate (weighted) average, as *winner violation profile*. Compare it to loser. Promote w-preferring constraints: TROCHAIC. Demote l-preferring constraints: FOOTREPULSION and NONFINAL.
- → solution: TROCHAIC ≫ NONFINAL ≫ FOOTREPULSION. *Learner's new grammar different from, but equivalent to teacher's!*

Revised *Robust Interpretive Parsing* (Biró, under review)

/ab.ra.ka.dab.ra/			NONFINAL	TROCHAIC	FOOTREPULSION
l	1.	*ab.ra.[ka.dáb].ra*		*	
	2.	*[àb.ra].[ka.dáb].ra*		*	*
☞	3.	*[àb.ra].ka.[dáb.ra]*	*		
w			0.5	0.5	0.5

- Teacher: FOOTREPULSION ≫ TROCHAIC ≫ NONFINAL, → ☞*[àb.ra].ka.[dáb.ra]*.
- Learner: NONFINAL ≫ TROCHAIC ≫ FOOTREPULSION, → l *ab.ra.[ka.dáb].ra*.
- Learner: w is either *[àb.ra].[ka.dáb].ra* or *[àb.ra].ka.[dáb.ra]*.
- Calculate (weighted) average, as *winner violation profile*. Compare it to loser. Promote w-preferring constraints: TROCHAIC. Demote l-preferring constraints: FOOTREPULSION and NONFINAL.
- → solution: TROCHAIC ≫ NONFINAL ≫ FOOTREPULSION. *Learner's new grammar different from, but equivalent to teacher's!*

Revised *Robust Interpretive Parsing* (Biró, under review)

	/ab.ra.ka.dab.ra/	NonFinal	Trochaic	FootRepulsion
l 1.	*ab.ra.[ka.dáb].ra*		*	
2.	*[àb.ra].[ka.dáb].ra*		*	*
☞ 3.	*[àb.ra].ka.[dáb.ra]*	*		
w		0.5	0.5	0.5

- Teacher: FootRepulsion ≫ Trochaic ≫ NonFinal, → ☞*[àb.ra].ka.[dáb.ra]*.
- Learner: NonFinal ≫ Trochaic ≫ FootRepulsion, → l *ab.ra.[ka.dáb].ra*.
- Learner: w is either *[àb.ra].[ka.dáb].ra* or *[àb.ra].ka.[dáb.ra]*.
- Calculate (weighted) average, as *winner violation profile*.
  Compare it to loser. Promote w-preferring constraints: Trochaic.
  Demote l-preferring constraints: FootRepulsion and NonFinal.
- → solution: Trochaic ≫ NonFinal ≫ FootRepulsion.
  Learner's new grammar different from, but equivalent to teacher's!

Revised *Robust Interpretive Parsing* (Biró, under review)

	/ab.ra.ka.dab.ra/	NONFINAL	TROCHAIC	FOOTREPULSION
l	1. *ab.ra.[ka.dáb].ra*		*	
	2. *[àb.ra].[ka.dáb].ra*		*	*
☞	3. *[àb.ra].ka.[dáb.ra]*	*		
w		0.5	0.5	0.5

- Teacher: FOOTREPULSION ≫ TROCHAIC ≫ NONFINAL, → ☞*[àb.ra].ka.[dáb.ra]*.
- Learner: NONFINAL ≫ TROCHAIC ≫ FOOTREPULSION, → l *ab.ra.[ka.dáb].ra*.
- Learner: w is either *[àb.ra].[ka.dáb].ra* or *[àb.ra].ka.[dáb.ra]*.
- Calculate (weighted) average, as *winner violation profile*. Compare it to loser. Promote w-preferring constraints: TROCHAIC. Demote l-preferring constraints: FOOTREPULSION and NONFINAL.
- → solution: TROCHAIC ≫ NONFINAL ≫ FOOTREPULSION. *Learner's new grammar different from, but equivalent to teacher's!*

Revised *Robust Interpretive Parsing* (Biró, under review)

	/ab.ra.ka.dab.ra/	NonFinal	Trochaic	FootRepulsion
l 1.	*ab.ra.[ka.dáb].ra*		*	
2.	*[àb.ra].[ka.dáb].ra*		*	*
☞ 3.	*[àb.ra].ka.[dáb.ra]*	*		
w		0.5	0.5	0.5

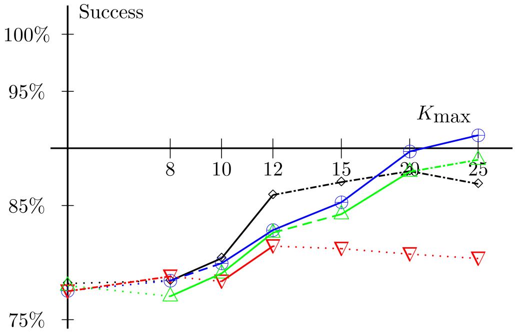
- Teacher: FootRepulsion $\gg$ Trochaic $\gg$ NonFinal, $\rightarrow$ ☞ *[àb.ra].ka.[dáb.ra]*.
- Learner: NonFinal $\gg$ Trochaic $\gg$ FootRepulsion, $\rightarrow$ l *ab.ra.[ka.dáb].ra*.
- Learner: w is either *[àb.ra].[ka.dáb].ra* or *[àb.ra].ka.[dáb.ra]*.
- Calculate (weighted) average, as *winner violation profile*. Compare it to loser. Promote w-preferring constraints: Trochaic. Demote l-preferring constraints: FootRepulsion and NonFinal.
- $\rightarrow$ solution: Trochaic $\gg$ NonFinal $\gg$ FootRepulsion.
  Learner's new grammar different from, but equivalent to teacher's!

Revised *Robust Interpretive Parsing* (Biró, under review)

/ab.ra.ka.dab.ra/	NONFINAL	TROCHAIC	FOOTREPULSION
l 1. *ab.ra.[ka.dáb].ra*		*	
2. *[àb.ra].[ka.dáb].ra*		*	*
☞ 3. *[àb.ra].ka.[dáb.ra]*	*		
w	0.5	0.5	0.5

- Teacher: FOOTREPULSION ≫ TROCHAIC ≫ NONFINAL, → ☞ *[àb.ra].ka.[dáb.ra]*.
- Learner: NONFINAL ≫ TROCHAIC ≫ FOOTREPULSION, → l *ab.ra.[ka.dáb].ra*.
- Learner: w is either *[àb.ra].[ka.dáb].ra* or *[àb.ra].ka.[dáb.ra]*.
- Calculate (weighted) average, as *winner violation profile*. Compare it to loser. Promote w-preferring constraints: TROCHAIC. Demote l-preferring constraints: FOOTREPULSION and NONFINAL.
- → solution: TROCHAIC ≫ NONFINAL ≫ FOOTREPULSION. *Learner's new grammar different from, but equivalent to teacher's!*

Revised *Robust Interpretive Parsing* (Biró, under review)



Success rate of learning a random target grammar, as a function of parameter K_{max}, for different update rules. Random initial grammar and random target grammar, with twelve constraints.

Overview

1. Optimality Theory: competence
2. Simulated Annealing for Optimality Theory: performance
3. Learning in Optimality Theory: problems
4. Conclusions

Conclusions

- *Optimality Theory:* model of linguistic competence.
  Grammatical form = (globally) optimal candidate.
- *Simulated Annealing for Optimality Theory:* model of performance.
  Produced form = locally optimal candidate.
  Predicting frequencies. More error, if SA-OT is run faster.
- Learning an OT grammar poses challenges,
  e.g., due to hidden structures. But solutions exists!
- (Omitted from this talk: OT in other domains (e.g., syntax),
  learning in presence of performance errors, iterated learning.)

Conclusions

- *Optimality Theory:* model of linguistic competence.
  Grammatical form = (globally) optimal candidate.
- *Simulated Annealing for Optimality Theory:* model of performance.
  Produced form = locally optimal candidate.
  Predicting frequencies. More error, if SA-OT is run faster.
- Learning an OT grammar poses challenges,
  e.g., due to hidden structures. But solutions exists!
- (Omitted from this talk: OT in other domains (e.g., syntax), learning in presence of performance errors, iterated learning.)

Conclusions

- *Optimality Theory:* model of linguistic competence.
  Grammatical form = (globally) optimal candidate.
- *Simulated Annealing for Optimality Theory:* model of performance.
  Produced form = locally optimal candidate.
  Predicting frequencies. More error, if SA-OT is run faster.
- Learning an OT grammar poses challenges,
  e.g., due to hidden structures. But solutions exists!
- (Omitted from this talk: OT in other domains (e.g., syntax), learning in presence of performance errors, iterated learning.)

Conclusions

- *Optimality Theory:* model of linguistic competence.
  Grammatical form = (globally) optimal candidate.
- *Simulated Annealing for Optimality Theory:* model of performance.
  Produced form = locally optimal candidate.
  Predicting frequencies. More error, if SA-OT is run faster.
- Learning an OT grammar poses challenges,
  e.g., due to hidden structures. But solutions exists!
- (Omitted from this talk: OT in other domains (e.g., syntax),
  learning in presence of performance errors, iterated learning.)

Thank you for your attention!

Tamás Biró:
t.s.biro@uva.nl

Work supported by:

Amsterdam Center for Language and Communication ACLC

NWO Netherlands Organisation for Scientific Research