Language and Computation

week 11, Tuesday, April 08

Tamás Biró
Yale University
tamas.biro@yale.edu
http://www.birot.hu/courses/2014-LC/

Practical matters

- **Post-reading:** JM 11
- **Pre-reading:** JM 17.1-2, 18.1, 19.1, 20.1
- `http://birot.hu/courses/2014-LC/readings.txt`
- Assignment 4 posted, due: 04/10.
- (To come(?): Viterbi and Forward-Backward – an example)
- Midterm returned.

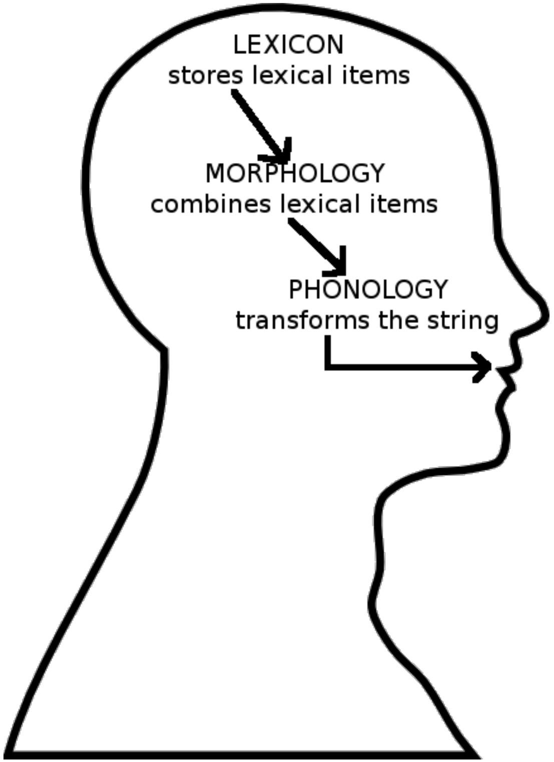
Today

- Optimality Theory: general definition
- Implementations of OT
- Learning OT

Next time: computational semantics.

Phonology as a (regular?) relation $(U, \mathrm{SF}(U))$

(While alternative approaches to phonology also exist,)
Lexicon + morphology → underlying form U
Phonology: $U \mapsto \mathrm{SF}$. Phonetics: SF to sound wave.



- Early generative phonology (SPE):
  cascade of context-sensitive rewrite rules.
  Procedural perspective

- Two-level phonology and morphology:
  declarative constraints.

- Optimality Theory: soft constraints.
  Teleological perspective

Optimality Theory: the basic idea

Optimality Theory

Simplified language **typology**:

- Stress on first syllable
- Stress on last syllable
- Stress on penultimate syllable
- No language with stress on second syllable as a rule

Optimality Theory

An OT model **to account for** this simplified language typology:

- EARLY: stress as early as possible
  # syllables intervening between left edge of word and stress.
- LATE: stress as late as possible
  # syllables intervening between stress and right edge.
- NONFINAL: stress not on last syllable.
  # of stresses on last syllable of the word.

Optimality Theory

$\text{Gen}(\sigma\sigma\sigma\sigma) = \{$[suuu], [usuu], [uusu], [uuus]$\}$.

$/\sigma\sigma\sigma\sigma/$	Early	Late	NonFinal
☞ [s u u u]	0	3	0
[u s u u]	1!	2	0
[u u s u]	2!	1	0
[u u u s]	3!	0	1

$\text{SF}(\sigma\sigma\sigma\sigma) =$[suuu]

Optimality Theory

$\text{Gen}(\sigma\sigma\sigma\sigma) = \{$[suuu], [usuu], [uusu], [uuus]$\}$.

	$/\sigma\sigma\sigma\sigma/$	NONFINAL	LATE	EARLY
	[s u u u]	0	3!	0
	[u s u u]	0	2!	1
☞	[u u s u]	0	1	2
	[u u u s]	1!	0	3

$\text{SF}(\sigma\sigma\sigma\sigma) =$[uusu]

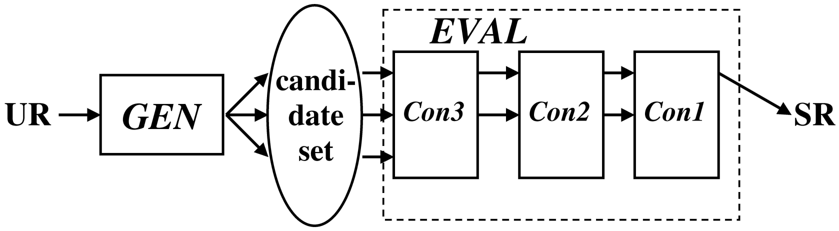
Optimality Theory

OT **accounts for** this simplified language typology:

- Stress on first syllable: $\textsc{Early} \gg \textsc{Late}, \textsc{NonFinal}$,
  as well as $\textsc{NonFinal} \gg \textsc{Early} \gg \textsc{Late}$
- Stress on last syllable $\textsc{Late} \gg \textsc{Early}, \textsc{NonFinal}$
- Stress on penultimate syllable
  $\textsc{NonFinal} \gg \textsc{Late} \gg \textsc{Early}$
- No language with stress on second syllable as a rule:
  No such hierarchy.

Basic ideas of Optimality Theory

- Gen and Eval



- Gen and constraints are universal.

- Constraints ranked into **strict domination hierarchy**

- Language typology due to differences in hierarchy
  $\rightarrow$ learning: find the correct hierarchy.

Basic ideas of Optimality Theory

Two views of Optimality Theory:

- Constraints as filters:
  "Clever" filters: filters out "worse ones", not "bad ones".

- Constraints as elementary functions:
  Find candidate that violates the "least" constraints.

Optimality Theory at a disciplinary crossroads

Theoretical linguistics	→	constraints
Computer science		
	→	optimization
Cognitive science		

OT: optimize some target function,
motivated by linguistic research.

Optimization in linguistics

$$\mathrm{SF}(u) = \underset{c \in \mathrm{Gen}(u)}{\arg\ \mathrm{opt}}\ H(c)$$

Harmony Grammar: $H(c) = \sum_{i=1}^{n} w_i \cdot C_i(c)$

opt: $\min$ for $<$ on $\mathbb{R}$.

Optimality Theory: $H(c) = (C_1(c), C_2(c), \ldots, C_n(c))$

opt: lexicographical order on $\mathbb{R}^n$.

Principles and Parameters: $H(c) = \bigwedge_{i=1}^{n} (w_i \vee C_i(c))$

opt: false "more optimal" than true.

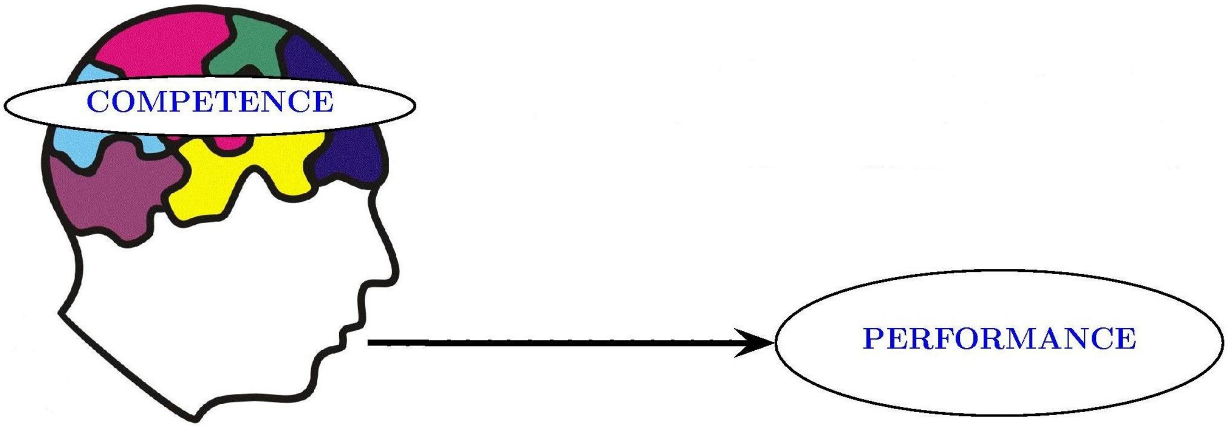
Implementing Optimality Theory

Implementations of Optimality Theory

How to find the most harmonic element of $\mathrm{Gen}(u)$?

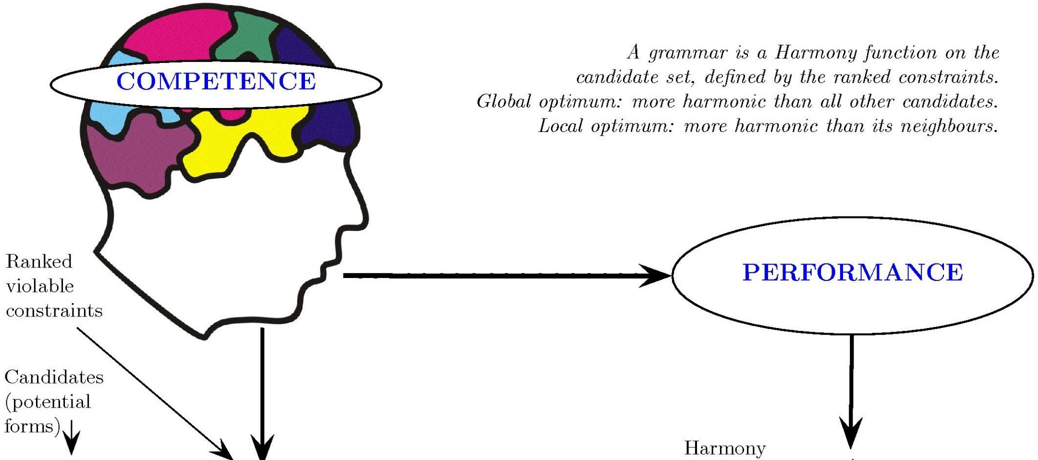
- Exhaustive search
- Finite state representations
- Dynamic programming / chart parsing
- Genetic algorithms
- Simulated annealing

Errors of the mental computation

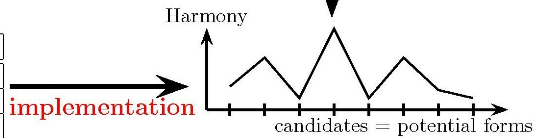


static knowledge	processing in the brain
Optimality Theory	Simulated Annealing for OT

Errors of the mental computation



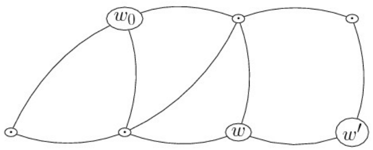
/aat/	NoCoda	Parse	Onset
☞ [a.a<t>]		*	**
[a.at]	*!		**
[<a>at]	*!	*	*

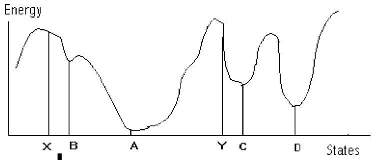


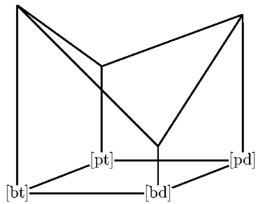
Optimality Theory
grammar
competence model
grammatical form = ☞ (globally) optimal candidate

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

Basic idea of Simulated Annealing



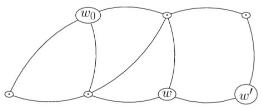


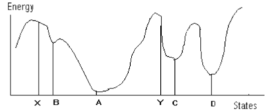


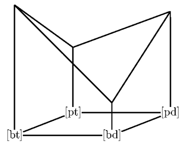
Step 1 – introducing landscape:

- 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



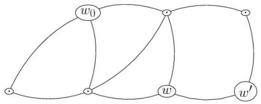


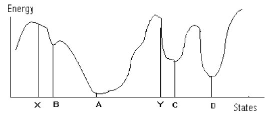


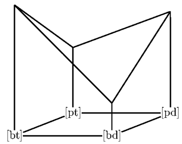
Step 2 – walking in this landscape:

- 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



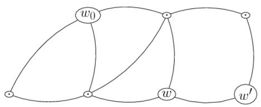


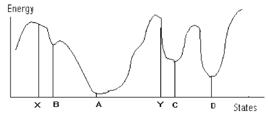


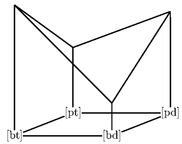
Step 3 – performing a random walk on this landscape:

- Start random walk from some initial position.
- End position returned as output of algorithm: form produced
- Hopefully, global optimum (grammatical form) found. Yet,
- Neighbourhood structure $\rightarrow$ 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!

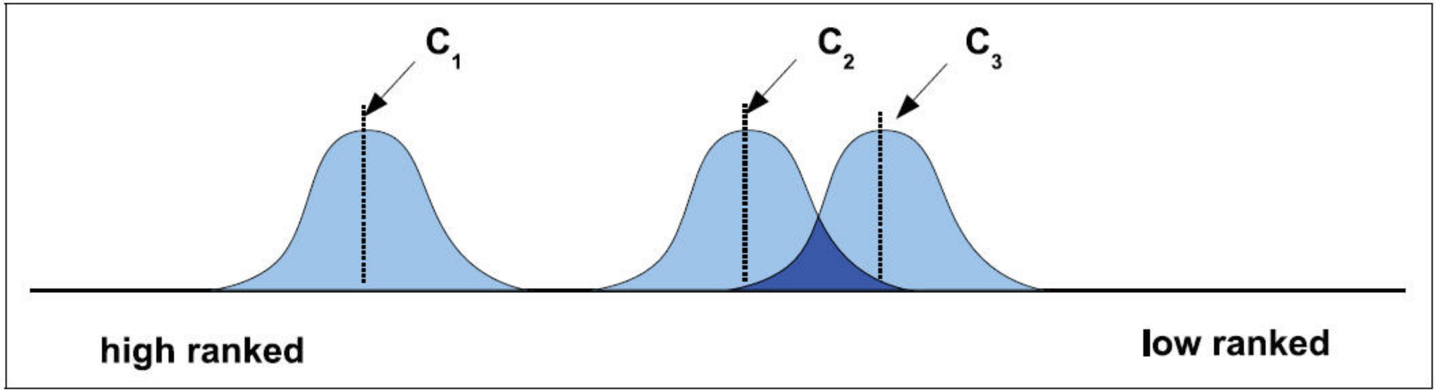
Basic idea of Simulated Annealing

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)	??

Variation in Optimality Theory

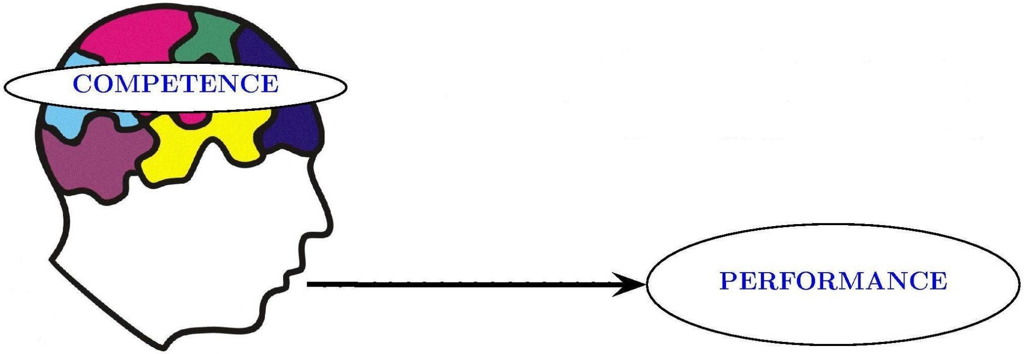
Often more than one grammatical form: $\mathrm{SF}_1 \sim \mathrm{SF}_2$.
Some possible approaches:

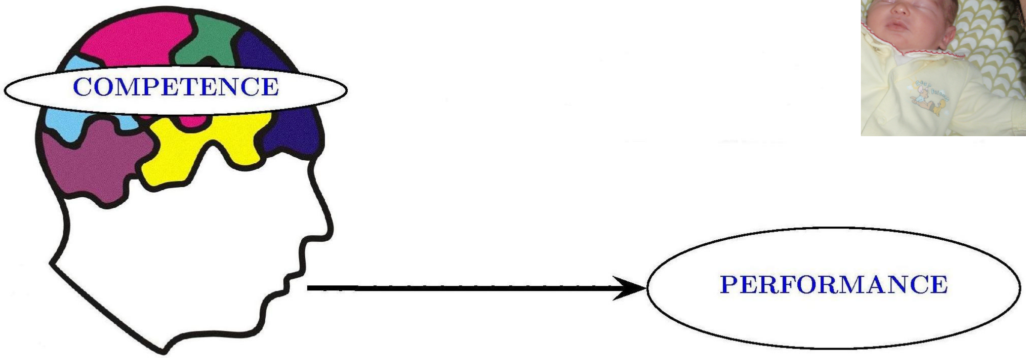
- More element in $\mathrm{Gen}(U)$, with same violation profile.
- Also generate other elements than $\mathrm{Gen}(U)$.
- 1 mental grammar = combination of more "elementary grammars".
  E.g, Paul Boersma's *Stochastic OT*:



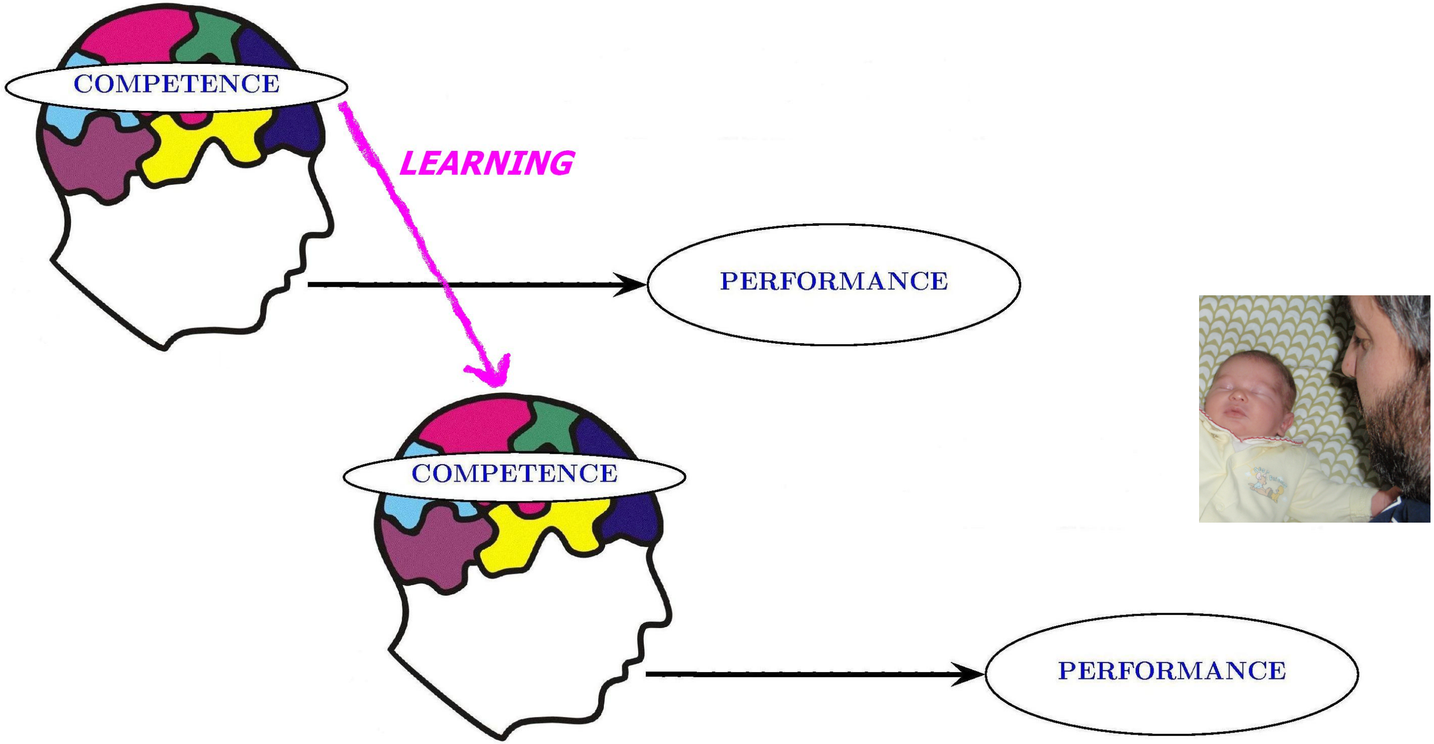
Learning Optimality Theory

Language acquisition

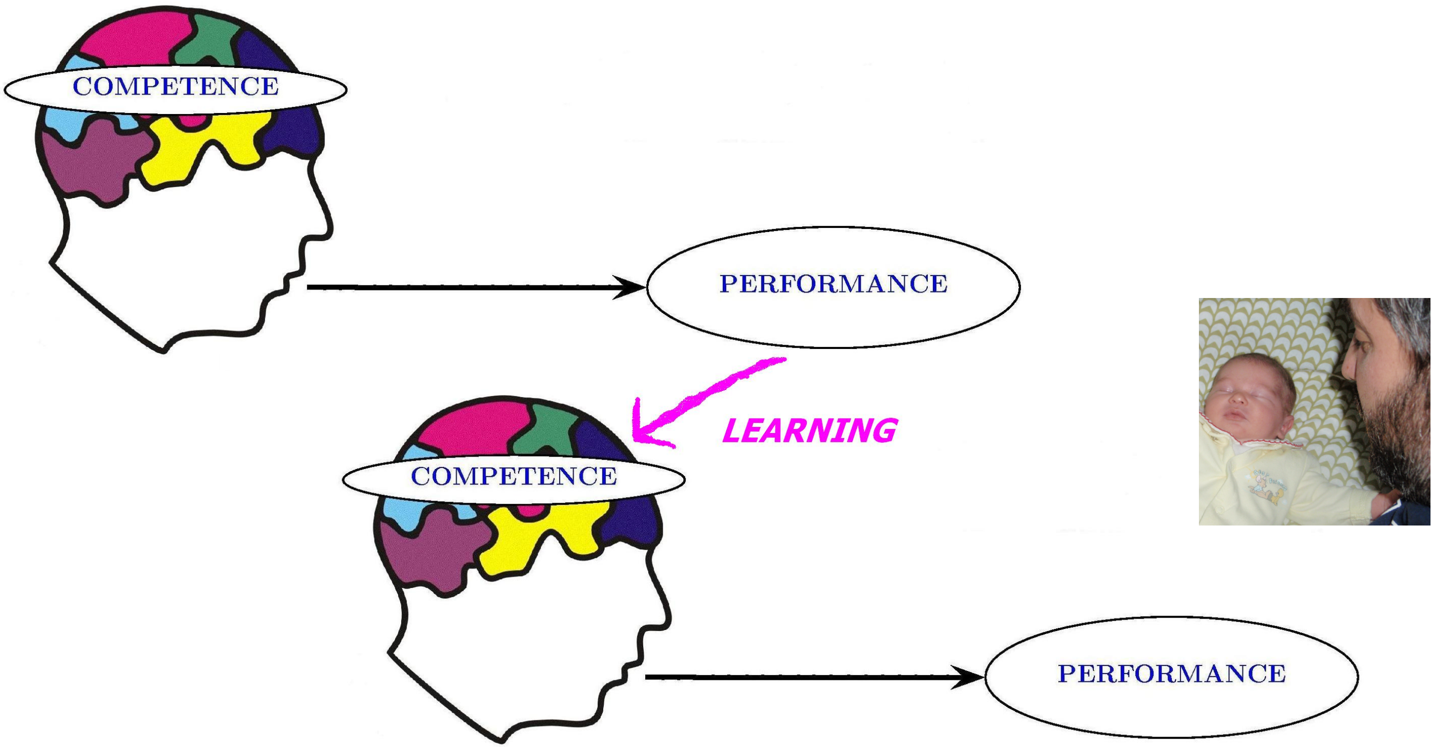




Language acquisition



Language acquisition



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 in Optimality Theory

General idea:

/underlying form/	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
Candidate 1 (learning observation)	$*!\longrightarrow$	$*\longrightarrow$			$*\longrightarrow$			
☞ Candidate 2 (learner's output)				$\longleftarrow*$		$\longleftarrow*$		

- Winner preferring constraints vs. Loser preferring constraints
- All L must be dominated by at least one W.
- Demote L, possibly promote W.

Learning in Optimality Theory

General idea:

	/underlying form/	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
	Candidate 1 (learning observation)	$*!\longrightarrow$	$*\longrightarrow$			$*\longrightarrow$			
☞	Candidate 2 (learner's output)				$\longleftarrow*$		$\longleftarrow*$		

- Recursive Constraint Demotion: off-line (batch learning)
- Error Driven Constraint Demotion: on-line
- Gradual Learning Algorithm

See you on Thursday!