Optimality Theory is not computable

Andrew Lamont
andrew.lamont@ucl.ac.uk

University College London

FLTiMaP – January 6

Handout available at
https://aphonologist.github.io/presentations

Optimality Theory is computationally complex

- Rule-based models of phonology are finite-state[1]
- Natural language phonology appears to be as well[2]
- Optimality Theory[3] is more powerful[4] and requires a lot of time and space[5] to compute

[1] Johnson (1972); Kaplan and Kay (1994)
[2] Heinz (2018)
[3] Prince and Smolensky (1993/2004)
[4] Eisner (1997, 2000); Frank and Satta (1998); Lamont (2021, 2022)
[5] Idsardi (2006); Hao (2024)

Optimality Theory is computationally complex

- In today's talk, I show that OT is not computable
- It is impossible to write an algorithm to determine the output of an arbitrary input and OT grammar

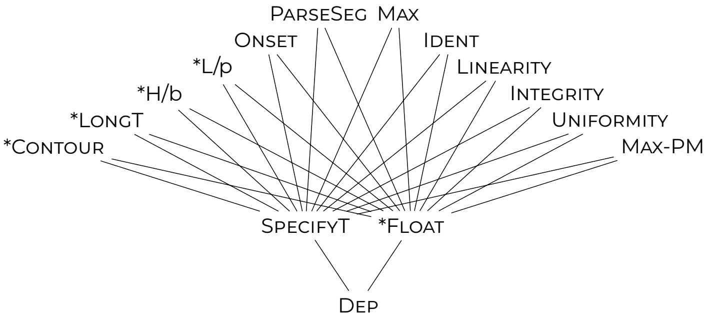
Related results

- OT-LFG has been shown not to be computable[1]
- Harmonic Grammar[2] can model arbitrary computations using non-linguistic representations[3]
- The present construction only uses off the shelf phonological tools and representations

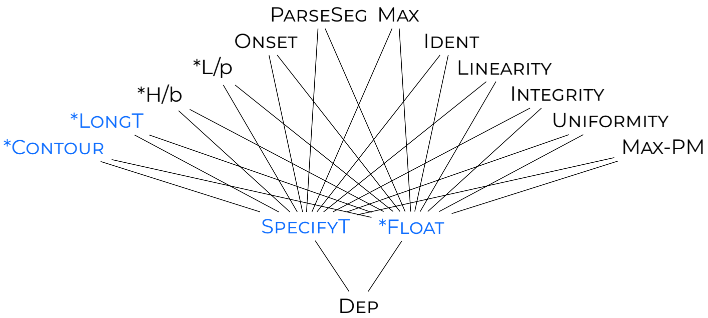
[1] Johnson (2002); Kuhn (2001, 2002, 2023)
[2] Legendre, Miyata, and Smolensky (1990)
[3] Smolensky (1992)

The grammar

- The relevant OT grammar has the following ranking:



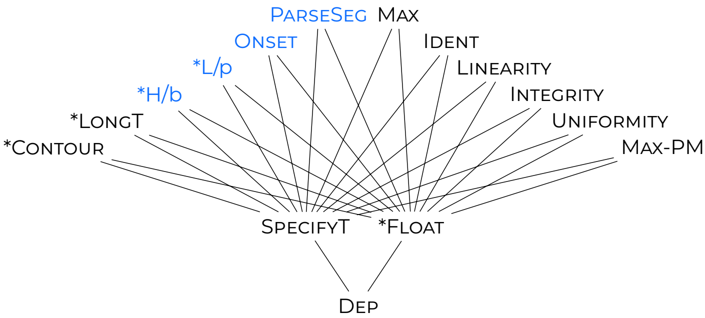
The grammar's phonotactics



- Every syllable must be linked to exactly one tone[4]
- Every tone must be linked to exactly one syllable

[4] Yip (2002)

The grammar's phonotactics



- High tones cannot link to [.l̤a.][4]
- Low tones cannot link to [.ha.]
- Segments must be parsed into .CV. syllables

[4] Lee (2008); Berkson (2013)

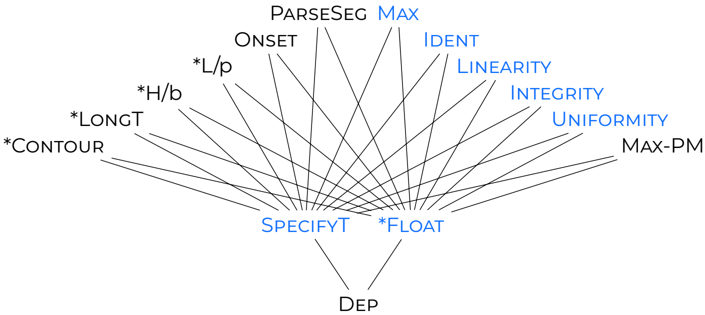
The grammar's phonotactics

- Wellformed strings satisfy all the phonotactic constraints and consist only of the syllables

H L
| |
.ha. and .la.

- One-to-one link between tones and syllables

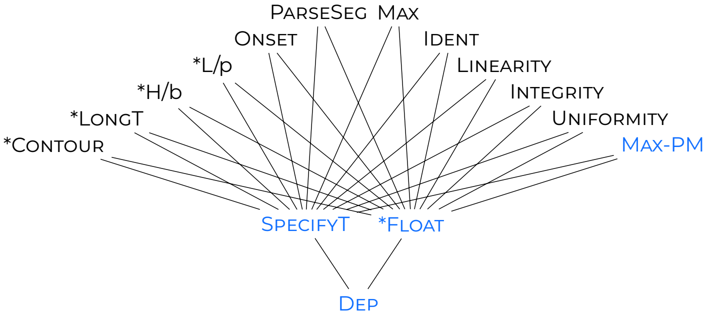
The grammar's mappings



- Wellformedness cannot be achieved by deleting/changing/reordering/. . . tones or segments[4]

[4] McCarthy and Prince (1994, 1995, 1999)

The grammar's mappings



- Wellformedness by inserting entire morphemes[4]
- MAX-PM: *tones/segs not associated to morphemes[5]

[4] Xu (2007, 2011); Wolf (2008, 2015); Rolle (2020)
[5] Walker and Feng (2004)

The grammar's lexicon

- Consider lexicons consisting of a finite set of unassociated strings of segments and tones

$$\text{lexicon} \subset \{\text{ha}, \text{l̤a}\}^* \times \{\text{H}, \text{L}\}^*$$

An illustration

- Given this ranking and the lexicon

  {hahaha, H H, H}

- The output of any monomorphemic input is:

```
 H   H   H          H   H   H          H   H   H
 |   |   |          |   |   |          |   |   |
ha  ha  ha    or   ha  ha  ha    or   ha  ha  ha
```

An illustration

/H/	MAX-PM	SPECIFYT	*FLOAT	DEP
a. H			W 1	L
b. .ʔə́.	W 2			L 2
c. .há.ha.ha.		W 1		L 6
☞ d. .há.há.há.				8
☞ e. .há.há.há.				8
☞ f. .há.há.há.				8
g. .há.há.há.há.há.há.				W 15

- SPECIFYT and *FLOAT defined as binary constraints[6]

[6] Frank and Satta (1998)

An illustration

/H/	MAX-PM	SPECIFYT	*FLOAT	DEP
a. H			W 1	L
b. .ʔə́.	W 2			L 2
c. .há.ha.ha.		W 1		L 6
☞ d. .há.há.há.				8
☞ e. .há.há.há.				8
☞ f. .há.há.há.				8
g. .há.há.há.há.há.há.				W 15

- Unassociated high tone fatally violates *FLOAT

An illustration

/H/	MAX-PM	SPECIFYT	*FLOAT	DEP
a. H			W 1	L
b. .ʔə́.	W 2			L 2
c. .há.ha.ha.		W 1		L 6
☞ d. .há.há.há.				8
☞ e. .há.há.há.				8
☞ f. .há.há.há.				8
g. .há.há.há.há.há.há.				W 15

- Inserting non-morphemic string violates MAX-PM

An illustration

/H/	MAX-PM	SPECIFYT	*FLOAT	DEP
a. H			W 1	L
b. .ʔə́.	W 2			L 2
c. .há.ha.ha.		W 1		L 6
☞ d. .há.há.há.				8
☞ e. .há.há.há.				8
☞ f. .há.há.há.				8
g. .há.há.há.há.há.há.				W 15

- Input tone can associate to inserted morpheme
- Introduces toneless syllables violating SPECIFYT

An illustration

/H/	MAX-PM	SPECIFYT	*FLOAT	DEP
a. H			W 1	L
b. .ʔə́.	W 2			L 2
c. .há.ha.ha.		W 1		L 6
☞ d. .há.há.há.				8
☞ e. .há.há.há.				8
☞ f. .há.há.há.				8
g. .há.há.há.há.há.há.				W 15

- High tone morphemes are inserted to avoid toneless syllables

An illustration

/H/	MAX-PM	SPECIFYT	*FLOAT	DEP
a. H			W 1	L
b. .ʔə́.	W 2			L 2
c. .há.ha.ha.		W 1		L 6
☞ d. .há.há.há.				8
☞ e. .há.há.há.				8
☞ f. .há.há.há.				8
g. .há.há.há.há.há.há.				W 15

- DEP selects the shortest well-formed candidate of infinitely many

Another illustration

- Given the same ranking and the lexicon

$$\{\text{lala}, \text{H H}, \text{H}\}$$

- All morphemic inputs are returned faithfully

Another illustration

/lala/	*H/b	Max	Ident	Max-PM	SpecifyT	Dep
☞ a. .la.la.					1	
b. .lá.lá.	W 2				L	W 2
c. ϵ		W 4			L	
d. .há.há.			W 2		L	W 2
e. .là.là.				W 2	L	W 2

Another illustration

/lala/	*H/b	MAX	IDENT	MAX-PM	SPECIFYT	DEP
☞ a. .la.la.					1	
b. .lá.lá.	W 2				L	W 2
c. ϵ		W 4			L	
d. .há.há.			W 2		L	W 2
e. .là.là.				W 2	L	W 2

- No morphemes have floating low tones
- Lexical insertion is ruled out by *H/b

Another illustration

/lala/	*H/b	MAX	IDENT	MAX-PM	SPECIFYT	DEP
☞ a. .la.la.					1	
b. .lá.lá.	W 2				L	W 2
c. ϵ		W 4			L	
d. .há.há.			W 2		L	W 2
e. .là.là.				W 2	L	W 2

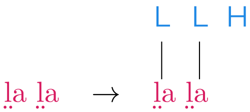
- Other operations violate constraints ≫ SPECIFYT

Technical wrinkle

- Given the same ranking and the lexicon

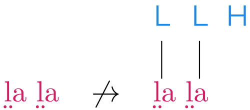
{lala, L L H}

- It is possible to satisfy SPECIFYT but not *FLOAT



Technical wrinkle

- If partially matched outputs are a formal problem, they can be ruled out with conjoined constraints[6]
  - SPECIFYT & DEP / word
  - *FLOAT & DEP / word
- These constraints block unfaithful mismatches



[6] Moreton and Smolensky (2002); Łubowicz (2002, 2003, 2005); Smolensky (2006)

General case

- In general, an OT grammar with this ranking either
  - Returns the faithful candidate (a), or
  - Returns the shortest well-formed candidate(s) (b′)

/x/	SPECIFYT	*FLOAT	DEP
☞ a. *x*	(1)	(1)	
b. *yxz*	(1)	(1)	W $\|y\| + \|z\|$
a′. *x*	(W 1)	(W 1)	L
☞ b′. *yxz*			$\|y\| + \|z\|$

General case

- Given a finite lexicon and an input defined over it, return the shortest candidate generated by inserting morphemes composed only of syllables

- If there is no such candidate, return the input as is
- Whether it exists is impossible to determine

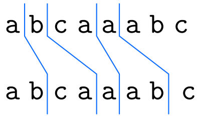
The PCP is not computable

- An instance of the Post Correspondence Problem[7] provides a finite set of domino types

$$\left\{ \begin{bmatrix} \text{b} \\ \hline \text{c a} \end{bmatrix}, \begin{bmatrix} \text{a} \\ \hline \text{a b} \end{bmatrix}, \begin{bmatrix} \text{c a} \\ \hline \text{a} \end{bmatrix}, \begin{bmatrix} \text{a b c} \\ \hline \text{c} \end{bmatrix} \right\}$$

- A solution is a finite sequence of domino tokens with the same string along the top and bottom

$$\begin{bmatrix} \text{a} \\ \hline \text{a b} \end{bmatrix} \begin{bmatrix} \text{b} \\ \hline \text{c a} \end{bmatrix} \begin{bmatrix} \text{c a} \\ \hline \text{a} \end{bmatrix} \begin{bmatrix} \text{a} \\ \hline \text{a b} \end{bmatrix} \begin{bmatrix} \text{a b c} \\ \hline \text{c} \end{bmatrix}$$



[7] Post (1946); Sipser (2013)

The PCP is not computable

- Post Correspondence Problem is not computable[7]
- It is impossible to write an algorithm that decides whether an arbitrary instance has a solution

[7] Post (1946)

The PCP is not computable

- Sipser (2013:§5.2) provides an intuitive proof
  - Given a Turing Machine and input, generate an instance of the PCP that models the computation
  - There is a solution if and only if the Turing Machine halts on the input
  - If an algorithm exists to solve the PCP, it implies a solution to the Halting Problem
  - But the Halting Problem is not computable[7]
  - Therefore no such algorithm exists

[7] Church (1936); Turing (1937)

The PCP is not computable

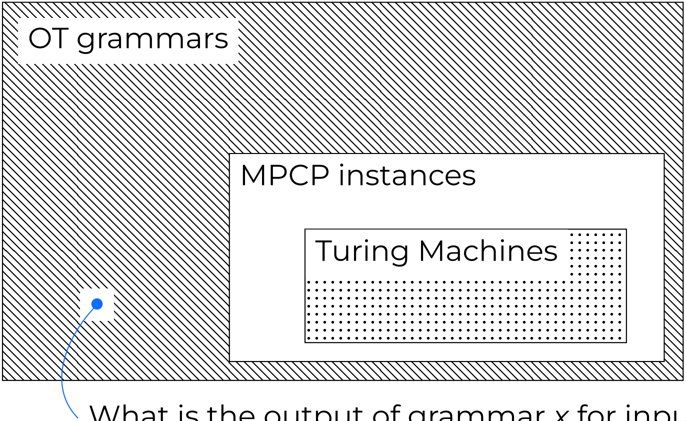
- Modified PCP specifies a starting domino
- It is also not computable[7]

[7] Harrison (1978)

OT is not computable

- Our OT grammar models the modified PCP and therefore it is impossible to write an algorithm that determines the output for an arbitrary input passed into arbitrary OT grammar
- OT is therefore not computable

OT is not computable



What is the output of grammar x for input y?

So what?

- OT is not the only constraint-based formalism that is not computable[8]
- However, it is primarily used to model phonology, which is a significantly larger gulf than a model of syntax/semantics

[8] Kaplan and Bresnan (1982); Johnson (1988); Kepser (2004); Francez and Wintner (2012); Kaplan and Wedekind (2023); Przepiórkowski (2023)

So what's next?

- The construction crucially depends on freedom of analysis and strictly ranked constraints
- Serialism does not automatically avoid this[9]
- Weighted constraints appear to (single signed)[10]
  - Removing one phonotactic violation cannot motivate arbitrarily many unfaithful operations
  - Can be shown to be finite-state with string constraints (work in prep.)
- Pursue alternative models like BMRS[11]

[9] Hampe (2022)
[10] Kimper (2016)
[11] Chandlee and Jardine (2021)

Thank you!

Handout available at
`https://aphonologist.github.io/presentations`

Kelly Harper Berkson. 2013. Phonation types in Marathi: An acoustic investigation. Doctoral dissertation, University of Kansas.

Jane Chandlee and Adam Jardine. 2021. Computational universals in linguistic theory: Using recursive programs for phonological analysis. *Language* 97:485–519.

Alonzo Church. 1936. An unsolvable problem of elementary number theory. *American Journal of Mathematics* 58:345–363.

Jason Eisner. 1997. What constraints should OT allow? Paper presented at LSA 71. Available at `http://roa.rutgers.edu/article/view/215`.

Jason Eisner. 2000. Directional constraint evaluation in Optimality Theory. In *Proceedings of the 18th International Conference on Computational Linguistics*, 257–263. The Association for Computational Linguistics.

Nissim Francez and Shuly Wintner. 2012. *Unification Grammars*. Cambridge: Cambridge University Press.

Robert Frank and Giorgia Satta. 1998. Optimality Theory and the generative complexity of constraint violability. *Computational Linguistics* 24:307–315.

Anton Hampe. 2022. The generative capacity of Harmonic Serialism. Bachelor's thesis, Universität Leipzig.

Sophie Hao. 2024. Universal generation for Optimality Theory is PSPACE-Complete. *Computational Linguistics* `https://doi.org/10.1162/coli_a_00494`.

Michael A. Harrison. 1978. *Introduction to Formal Language Theory*. Addison-Wesley Series in Computer Science, Reading, MA and Menlo Park, CA and London and Amsterdam and Don Mills, Ontario and Sydney: Addison-Wesley Publishing Company.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. In *Phonological Typology*, ed. by Larry M. Hyman and Frans Plank, 126–195. Berlin: De Gruyter Mouton.

William J. Idsardi. 2006. A simple proof that Optimality Theory is computationally intractable. *Linguistic Inquiry* 37:271–275.

C. Douglas Johnson. 1972. *Formal Aspects of Phonological Description*. The Hague: Mouton.

Mark Johnson. 1988. *Attribute-Value Logic and the Theory of Grammar*, volume 16 of *CSLI Lecture Notes*. Stanford, CA: Center for the Study of Language and Information.

Mark Johnson. 2002. Optimality-theoretic Lexical Functional Grammar. In *The Lexical Basis of Sentence Processing: Formal, computational and experimental issues*, ed. by Paola Merlo and Suzanne Stevenson, volume 4 of *Natural Language Processing*, 59–73. Amsterdam / Philadelphia: John Benjamins Publishing Company.

Ronald M. Kaplan and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In *The mental representation of grammatical representation*, ed. by Joan Bresnan, 173–281. Cambridge, MA: The MIT Press.

Ronald M. Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20:331–378.

Ronald M. Kaplan and Jürgen Wedekind. 2023. Formal and computational properties of LFG. In *Handbook of Lexical Functional Grammar*, ed. by Mary Dalrymple, volume 13 of *Empirically Oriented Theoretical Morphology and Syntax*, 1035–1082. Berlin: Language Science Press.

Stephan Kepser. 2004. On the complexity of RSRL. *Electronic Notes in Theoretical Computer Science* 53:146–162.

Wendell Kimper. 2016. Positive constraints and finite goodness in Harmonic Serialism. In *Harmonic Grammar and Harmonic Serialism*, ed. by John J. McCarthy and Joe Pater, 221–235. Sheffield: Equinox Publishing.

Jonas Kuhn. 2001. Formal and computational aspeects of optimality-theoretic syntax. Doctoral dissertation, Universität Stuttgart.

Jonas Kuhn. 2002. OT syntax: Decidability of generation-based optimization. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, ed. by Pierre Isabelle, Eugene Charniak, and Dekang Lin, 48–55. Philadelphia, PA: Association for Computational Linguistics.

Jonas Kuhn. 2023. LFG, Optimality Theory and learnability of languages. In *Handbook of Lexical Functional Grammar*, ed. by Mary Dalrymple, volume 13 of *Empirically Oriented Theoretical Morphology and Syntax*, 961–1032. Berlin: Language Science Press.

Andrew Lamont. 2021. Optimizing over subsequences generates context-sensitive languages. *Transactions of the Association for Computational Linguistics* 9:528–537.

Andrew Lamont. 2022. Optimality theory implements complex functions with simple constraints. *Phonology* 38:729–740.

Seunghun Julio Lee. 2008. Consonant-tone interaction in Optimality Theory. Doctoral dissertation, Rutgers, The State University of New Jersey.

Géraldine Legendre and Yoshiro Miyata and Paul Smolensky. 1990. Harmonic Grammar: a formal multi-level connectionist theory of linguistic well-formedness: an application. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, 884–891. Hillsdale, NJ: Erlbaum.

Anna Łubowicz. 2002. Derived environment effects in Optimality Theory. *Lingua* 112:243–280.

Anna Łubowicz. 2003. Local conjunction and comparative markedness. *Theoretical Linguistics* 29:101–112.

Anna Łubowicz. 2005. Locality of conjunction. In *Proceedings of the 24th West Coast Conference on Formal Linguistics*, ed. by John Alderete, Chung hye Han, and Alexei Kochetov, 254–262. Somerville, MA: Cascadilla Press.

John J. McCarthy and Alan Prince. 1994. The emergence of the unmarked: Optimality in prosodic morphology. In *Proceedings of NELS 24*, ed. by Mercè Gonzàlez, volume 2, 333–379. Amherst, MA: Graduate Linguistics Students Association.

John J. McCarthy and Alan Prince. 1995. Faithfulness and reduplicative identity. In *Papers in Optimality Theory*, ed. by Jill Beckman, Suzanne Urbanczyk, and Laura Walsh Dickey, 249–384. Amherst, MA: Graduate Linguistics Students Association.

John J. McCarthy and Alan Prince. 1999. Faithfulness and identity in prosodic morphology. In *The Prosody-Morphology Interface*, ed. by René Kager, Harry van der Hulst, and Wim Zonneveld, 218–309. Cambridge: Cambridge University Press.

Elliott Moreton and Paul Smolensky. 2002. Typological consequences of local constraint conjunction. In *Proceedings of WCCFL 21*, ed. by Line Mikkelsen and Christopher Potts, 306–319. Somerville, MA: Cascadilla Press.

Emil L. Post. 1946. A variant of a recursively unsolvable problem. *Bulletin of the American Mathematical Society* 52:264–268.

Alan Prince and Paul Smolensky. 1993/2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Malden, MA: Blackwell Publishing.

Adam Przepiórkowski. 2023. LFG and Head-Driven Phrase Structure Grammar. In *Handbook of Lexical Functional Grammar*, ed. by Mary Dalrymple, volume 13 of *Empirically Oriented Theoretical Morphology and Syntax*, 1861–1918. Berlin: Language Science Press.

Nicholas Rolle. 2020. In support of an OT-DM model: Evidence from clitic distribution in Degema serial verb constructions. *Natural Language & Linguistic Theory* 38:201–259.

Michael Sipser. 2013. *Introduction to the Theory of Computation*. Boston, MA: Cengage Learning, 3 edition.

Paul Smolensky. 1992. Harmonic Grammars for formal languages. In *Advances in Neural Information Processing Systems 5*, ed. by Stephen José Hanson, Jack D. Cowan, and C. Lee Giles, 847–854. San Francisco: Morgan Kaufmann Publishers Inc.

Paul Smolensky. 2006. Optimality in phonology II: Harmonic completeness, local constraint conjunction, and feature-domain markedness. In *The harmonic mind: From neural completeness to Optimality-Theoretic grammar*, ed. by Paul Smolensky and Géraldine Legendre, volume II, 27–160. Cambridge, MA: MIT Press.

A. M. Turing. 1937. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society* s2-42:230–265.

Rachel Walker and Bella Feng. 2004. A ternary model of morphology-phonology correspondence. In *Proceedings of WCCFL 23*, ed. by Vineeta Chand, Ann Kelleher, Angelo J. Rodríguez, and Benjamin Schmeiser, 787–800. Somerville, MA: Cascadilla Press.

Matthew Wolf. 2008. Optimal interleaving: Serial phonology-morphology interaction in a constraint-based model. Doctoral dissertation, University of Massachusetts Amherst.

Matthew Wolf. 2015. Lexical insertion occurs in the phonological component. In *Understanding Allomorphy: Perspectives from Optimality Theory*, ed. by Eulàlia Bonet, Maria-Rosa Lloret, and Joan Mascaró Altimiras, 361–407. Sheffield: Equinox.

Zheng Xu. 2007. Inflectional morphology in Optimality Theory. Doctoral dissertation, Stony Brook University.

Zheng Xu. 2011. Optimality Theory and morphology. *Language and Linguistics Compass* 5:424–508.

Moira Yip. 2002. *Tone*. Cambridge Textbooks in Linguistics, Cambridge: Cambridge University Press.