ADVANTAGES OF CONSTITUENCY: COMPUTATIONAL PERSPECTIVES ON SAMOAN WORD PROSODY

KRISTINE M. YU

SEPTEMBER 24, 2016

NORTHEAST COMPUTATIONAL PHONOLOGY CIRCLE 2016

UMASS AMHERST

ASSUMING CONSTITUENTS TO CAPTURE GENERALIZATIONS

- If we're right that **phonological constituents** really exist, then we want to see evidence that, assuming their existence:
  - We're able to give a "better" account of natural language than we could otherwise
  - That is, assuming constituents allows us to capture generalizations in some sense

EVIDENCE THAT CONSTITUENCY CAPTURES GENERALIZATIONS

- What gets reduplicated, restrictions on minimal words (McCarthy and Prince 1986/1996)
- Restrictions on allowed stress patterns (Liberman and Prince 1977)
- Domains of segmental processes (Selkirk 1980, Nespor and Vogel 1986)
- Patterns for where certain tones get placed (Pierrehumbert 1980)
- Patterns of variation in phonetic duration (Wightman et al. 1992) and the strength of articulatory gestures (Fougeron and Keating 1997)

PHONOLOGISTS ARE OFTEN EXPLICIT ABOUT WHETHER THEY SUBSCRIBE TO LEVEL ORDERING OR OUTPUT–OUTPUT CORRESPONDENCE (RARELY BOTH). BUT WE TEND TO HELP OURSELVES TO PROSODIC DOMAINS WITHOUT FURTHER COMMENT.

Kie Zuraw, 2009

https://www.mcgill.ca/linguistics/files/ linguistics/Handout_RevisedForMcGill.pdf

BUT…

- Controversy about syllable as a unit (Steriade)
- Some analyses of stress patterns without feet (Bailey 1995; Gordon 2002, 2011)
- Not clear that phonological analyses referring to prosodic constituents are "better" than alternative ones that don't
  - *Samoan word prosody*. (Zuraw, Yu, and Orfitelli 2014)
    - ALIGN constraints that impose PWds at domain of footing, or
    - ALIGN constraints that place feet directly at morpheme boundaries, bypassing PWds
- Computational descriptions of phonological patterns have revealed hypothesized strong structural universals <u>without referring to constituents</u> (e.g. work by UDel phonology lab)

RESEARCH QUESTION

Do constituents make phonological grammars for Samoan word stress more succinct?

- One way to start to get a grip on whether we get explanatory advantage by assuming existence of constituents: succinctness (Chomsky 1965; Berwick 1982, 2015, i.a.)
- Succinctness as a consequence of constituency has not been carefully explored computationally in phonology
- *Case study*: Comparison of succinctness of four grammar fragments generating Samoan stress patterns in monomorphemic words, with and without reference to feet as constituents

Similar kinds of succinctness comparisons include: Chomsky (1965); Chomsky and Halle (1968); Meyer and Fischer (1971); Hartmanis (1980); Stabler (2013); Berwick (2015); Rasin and Katzir (To appear)

RESEARCH QUESTION

Do constituents make phonological grammars for Samoan word stress more succinct?

DEFINITION OF LANGUAGE "LITTLE SAMOAN" (LSMO)

- Simplified version of description of Samoan stress in monomorphs in Zuraw, Yu, and Orfitelli 2014
- Language of strings of light and heavy syllables marked for primary, secondary, or no stress

DEFINITION OF LANGUAGE "LITTLE SAMOAN" (LSMO)

- Simplified version of description of Samoan stress in monomorphs in Zuraw, Yu, and Orfitelli 2014
- Language of strings of light and heavy syllables marked for primary, secondary, or no stress

Basic primary stress pattern: moraic trochee at the right edge

. . .ˈH#	la(ˈvaː)	'energised'
. . .ˈLL#	(ˈmanu)	'bird'
. . .ˈLLL#	i(ˈŋoa)	'name'

DEFINITION OF LANGUAGE "LITTLE SAMOAN" (LSMO)

- Simplified version of description of Samoan stress in monomorphs in Zuraw, Yu, and Orfitelli 2014
- Language of strings of light and heavy syllables marked for primary, secondary, or no stress

Basic primary stress pattern: moraic trochee at the right edge

. . .ˈH#	la(ˈvaː)	'energised'
. . .ˈLL#	(ˈmanu)	'bird'
. . .ˈLLL#	i(ˈŋoa)	'name'

Secondary stress in LSmo vs. Fijian (Zuraw et al., 2014, (8)), (Hayes, 1995, p. 143–4, (44a, 47))

String	LSmo	Fijian	Gloss
LLLH	(ˈmini)si(ˈtaː)	mi(ˌnisi)(ˈtaː)	'minister'
LLLLL	(ˈtemo)ka(ˈlasi)	pe(ˌresi)(ˈtẽndi)	'democracy', 'president'

DEFINITION OF LANGUAGE "LITTLE SAMOAN" (LSMO)

- Simplified version of description of Samoan stress in monomorphs in Zuraw, Yu, and Orfitelli 2014
- Language of strings of light and heavy syllables marked for primary, secondary, or no stress

Basic primary stress pattern: moraic trochee at the right edge

. . .ˈH#	la(ˈvaː)	'energised'
. . .ˈLL#	(ˈmanu)	'bird'
. . .ˈLLL#	i(ˈŋoa)	'name'

Secondary stress in LSmo vs. Fijian (Zuraw et al., 2014, (8)), (Hayes, 1995, p. 143–4, (44a, 47))

Initial dactyl effect in Samoan, LSmo

String	LSmo	Fijian	Gloss
LLLH	(ˈmini)si(ˈtaː)	mi(ˌnisi)(ˈtaː)	'minister'
LLLLL	(ˈtemo)ka(ˈlasi)	pe(ˌresi)(ˈtendi)	'democracy', 'president'

RESEARCH QUESTION

Do constituents make phonological grammars for Samoan word stress more succinct?

OVERVIEW OF GRAMMARS

- **Four grammars:**
  - Direct account with feet
  - Direct account referring to syllables only
  - Karttunen OT with feet
  - Karttunen OT referring to syllables only
- Direct accounts: directly describe restrictions on the surface stress patterns. **Important: boundaries not placed in the alphabet (boundary symbol theory; Chomsky 1965, Selkirk 1980). Boundaries placed by grammar!**
- Karttunen OT: finite state implementation of OT, maps underlying forms directly to surface forms rather than violation vectors, no EVAL
  - EVAL is not a finite state process! Number of states required for EVAL cannot be bounded (Eisner 1997, Karttunen 1998).

DEFINING THE GRAMMARS IN `xfst`

- All of these grammar fragments can be expressed in a regular grammar
- We define the grammars in **`xfst`**, a formalism explicitly designed to make it natural to state phonological grammars
  - Includes pre-defined operators and capacity for definition of own operator and units which allow us to write grammars at very high level, e.g. with SPE style rules `A -> B || L _ R`
  - Compiles our high-level grammars to machine-level finite state transducers for us
  - Provides common formalism in which we can define all four grammars and measure grammar size in a controlled comparison

(Beesley and Karttunen, 2003), https://web.stanford.edu/~laurik/fsmbook/home.html

RESEARCH QUESTION

Do constituents make phonological grammars for Samoan word stress more succinct?

DEFINING SUCCINCTNESS

Succinctness: the size of the grammar, i.e. the number of symbols it takes to write it down in `xfst`

- Special case of minimum description length (MDL), which balances:
  - Minimizing size of grammar: favors simple grammars that often overgenerate
  - Minimizing size of data encoded by grammar: favors restrictive but often overly memorized grammars
- **Here, MDL reduces to size of grammar**
  - Common `xfst` formalism for expressing the grammars
  - Data same across comparisons: stress patterns up to 5 syllables
  - All grammars admit exactly same set of stress patterns up to 5 syllables
    - Size of encodings of sequences up to 5 syllables is (nearly) exactly the same, since possibilities allowed by grammars in that range is (nearly) identical
    - Limit testing empirical coverage of stress patterns to monomorphs of 5 syllables due to lack of data on longer words

OPERATIONALIZED RESEARCH QUESTION

Do constituents make phonological grammars for Samoan word stress more succinct?

OPERATIONALIZED RESEARCH QUESTION

Does reference to feet reduce the number of symbols used in `xfst`, in defining direct approach and Karttunen OT grammars for stress patterns in Samoan monomorphs?

CODE IS AVAILABLE AT . . .

https://github.com/krismyu/smo-constituency-feet

COMMON GEN FOR ALL GRAMMARS

Input: `LL`

GEN

Output:

`P[L]P[L]`
`P[L]W[L]`
`P[L]S[L]`
`W[L]P[L]`
`W[L]W[L]`
`W[L]S[L]`
`S[L]P[L]`
`S[L]W[L]`
`S[L]S[L]`

DIRECT ACCOUNT WITH FEET IN `xfst`

- Parse into feet, e.g. two LLs form a foot, any heavy syllable is a foot

```
define ParseFoot [ [ [ ? Light ]^2 | [ ? Heavy ] ] -> "(" ... ")" ];
```

- Define feet and restrictions on feet, e.g. trochaic foot form

```
# A foot is a string of non-parentheses enclosed in parentheses


define Foot [ "(" [ \["(" | ")"] ]*  ")" ];
```

- Define restrictions on words in terms of feet, e.g. word must terminate in foot bearing primary stress, initial dactyl effect

```
# Don't have a sequence of a weak light followed by a LL foot at the
#  beginning of the word if the LLL sequence is non-final (non-final
#  to allow for 3L WPW).


define InitialDactyl ~[ WeakLight LLFoot ?+ ];
```

DIRECT ACCOUNT WITH FEET IN `xfst`

- Parse into feet, e.g. two LLs form a foot, any heavy syllable is a foot

```
define ParseFoot [ [ [ ? Light ]^2 | [ ? Heavy ] ] -> "(" ... ")" ];
```

- Define feet and restrictions on feet, e.g. trochaic foot form

```
# A foot is a string of non-parentheses enclosed in parentheses


define Foot [ "(" [ \["(" | ")"] ]*  ")" ];
```

- Define restrictions on words in terms of feet, e.g. word must terminate in foot bearing primary stress, initial dactyl effect

```
# Don't have a sequence of a weak light followed by a LL foot at the
#  beginning of the word if the LLL sequence is non-final (non-final
#  to allow for 3L WPW).


define InitialDactyl ~[ WeakLight LLFoot ?+ ];
```

We can define units in terms of feet and then refer to them!

DIRECT ACCOUNT WITH SYLLABLES IN `xfst`

- No parsing into feet!
- Allow only strings containing exactly one primary stress
- Restrict heavy syllables to be stressed
- Restrict position of primary lights and secondary lights

```
# Secondary light must always by followed by a non-final weak light
# and must either be string-initial, or preceded by a weak light or a
# heavy syllable.
```

```
define StressSLight [ SecondaryLight => [.#. | W2 | [StressedSyll WeakLight] | Heavy] _ WeakLight ? ];
```

- Restrict position of lapses

DIRECT ACCOUNT WITH SYLLABLES IN `xfst`

- No parsing into feet!
- Allow only strings containing exactly one primary stress
- Restrict heavy syllables to be stressed
- Restrict position of primary lights and secondary lights

```
# Secondary light must always by followed by a non-final weak light
# and must either be string-initial, or preceded by a weak light or a
# heavy syllable.
```

```
define StressSLight [ SecondaryLight => [.#. | W2 | [StressedSyll WeakLight] | Heavy] _ WeakLight ? ];
```

- Restrict position of lapses

Many statements of case-by-case restrictions!

KARTTUNEN OT WITH FEET: CONSTRAINTS

(2.2) Partial ranking of constraint set for Karttunen OT, using feet

I. Stratum 1

i. FOOTBINARITY (FOOTBIN) A foot must contain exactly two moras.

ii. RHYTHMTYPE=TROCHEE (RHTYPE=TROCHEE) A foot must have stress on its initial mora, and its initial mora only.

iii. ALIGN(PWD,R; 'FT,R) (EDGEMOST-R) The end of the prosodic word must coincide with the end of a primary-stressed foot.

II. Stratum 2

i. PARSE-σ Every syllable must be included in a foot.

ii. ALIGN(PWD;L,FT,L) The beginning of the prosodic word must coincide with the beginning of a foot.

- Constraint set taken from Zuraw, Yu, Orfitelli (2014)
- Partial ranking computed with OTSoft (Hayes et al., 2016)

KARTTUNEN OT, WITH FEET: ALIGN FAMILIES

- Treat ALIGN(PWd;L,Ft,L) as categorical, as in Zuraw et al. 2014

```
# Definition: beginning of PWd must coincide with beginning of foot.
# Change to "beginning of input string must coincide with beginning of foot"

define AlignWdLFtL ["(" ?*];
```

- But compute EDGEMOST('Ft, R; Wd, R) as categorical rather than gradient
  - Fine since undominated

```
define Foot ["(" [\[ "(" | ")" ]]*  ")"];
define PrimaryFoot [ Foot & $["P"] ];
define EdgemostR [ \P* PrimaryFoot];
```

KARTTUNEN OT, WITH FEET: ALIGN FAMILIES

- Treat ALIGN(PWd;L,Ft,L) as categorical, as in Zuraw et al. 2014

```
# Definition: beginning of PWd must coincide with beginning of foot.
# Change to "beginning of input string must coincide with beginning of foot"

define AlignWdLFtL ["(" ?*];
```

- But compute EDGEMOST('Ft, R; Wd, R) as categorical rather than gradient
  - Fine since undominated

```
define Foot ["(" [\[ "(" | ")" ]]*  ")"];
define PrimaryFoot [ Foot & $["P"] ];
define EdgemostR [ \P* PrimaryFoot];
```

Can restrict all constraints to be categorical!

KARTTUNEN OT, WITH FEET: PARSE FAMILY

- Parse constraint, even though categorical, must be expanded and approximated by family of constraints in Karttunen OT
  - Can have multiple loci of violation (and thus multiple violations)
  - Need to be able to count how many violations
  - Finite system can't make infinitely many degrees of well-formedness

```
define ParseSyll ~[$"X"] ;
define ParseSyll1 ~[[$"X"]^>1] ;
define ParseSyll2 ~[[$"X"]^>2] ;
define ParseSyll3 ~[[$"X"]^>3] ;
define ParseSyll4 ~[[$"X"]^>4] ;
define ParseSyll5 ~[[$"X"]^>5] ;
define ParseSyll6 ~[[$"X"]^>6] ;
define ParseSyll7 ~[[$"X"]^>7] ;
```

KARTTUNEN OT, WITH FEET: PARSE FAMILY

- Parse constraint, even though categorical, must be expanded and approximated by family of constraints in Karttunen OT
  - Can have multiple loci of violation (and thus multiple violations)
  - Need to be able to count how many violations
  - Finite system can't make infinitely many degrees of well-formedness

```
define ParseSyll ~[$"X"] ;
define ParseSyll1 ~[[$"X"]^>1] ;
define ParseSyll2 ~[[$"X"]^>2] ;
define ParseSyll3 ~[[$"X"]^>3] ;
define ParseSyll4 ~[[$"X"]^>4] ;
define ParseSyll5 ~[[$"X"]^>5]
define ParseSyll6 ~
define ParseSyll7 ~[
```

If multiple violations possible, must be defined as constraint family in Karttunen OT!

KARTTUNEN OT, SYLLABLES ONLY: CONSTRAINTS

(2.3) Partial ranking of constraint set for Karttunen OT, using syllables only

I. Stratum 1

i. WEIGHTTOSTRESS (WSP) A heavy syllable must be stressed.

ii. NONFINALITYL A word-final light syllable must be unstressed.

iii. NOLAPSEFOLLOWINGHEAVY A heavy syllable must not be followed by two unstressed syllables.

iv. NOINITIALWS A word must not begin with an unstressed-stressed sequence.

II. Stratum 2, 3, 4, 5

i. ALIGN(x2,R,x0,PWd) Assign a violation for every grid mark of level 2 that does not coincide with the right edge of level 0 grid marks in a prosodic word.

ii. *CLASH Assign a violation for every sequence of two stressed syllables.

iii. *LAPSE Assign a violation for every sequence of two unstressed syllables.

iv. ALIGN(x1;L,x0,PWd) Assign a violation for every grid mark of level 1 that does not coincide with the right edge of level 0 grid marks in a prosodic word.

- Constraint set based on Gordon 2002, 2011, Kager 2005, plus ad-hoc ones that were necessary to get the empirical coverage desired
- Partial ranking computed with OTSoft (Hayes et al., 2016)

KARTTUNEN OT, SYLLABLES ONLY: CLASH FAMILY

Requires counting, doing arithmetic

```
define S2 [Stressed Stressed];

define No1Clash ~[$[S2]];
define No2Clash ~[$[Stressed]^3] & ~[[$[S2]]^2];
define No3Clash ~[$[Stressed]^4] &  ~[[$[S2]]^3] & ~[?* [Stressed]^3 $[[Stressed]^2]] & ~[$[[Stressed]]^2 [Stressed]^3 ?*];
define No4Clash ~[$[Stressed]^5] &  ~[[$[S2]]^4] & ~[?* [Stressed]^4 $[[Stressed]^2]] & ~[$[[Stressed]^2] [Stressed]^4 ?*] & ~[[$[[Stresse
```

Penalize 1 clash: 10 symbols
Penalize 2 clashes: 25 symbols
Penalize 3 clashes: 61 symbols
Penalize 4 clashes: 131 symbols

KARTTUNEN OT, SYLLABLES ONLY: CLASH FAMILY

Requires counting, doing arithmetic

```
define S2 [Stressed Stressed];

define No1Clash ~[$[S2]];
define No2Clash ~[$[Stressed]^3] & ~[[$[S2]]^2];
define No3Clash ~[$[Stressed]^4] &  ~[[$[S2]]^3] & ~[?* [Stressed]^3 $[[Stressed]^2]] & ~[$[[Stressed]]^2 [Stressed]^3 ?*];
define No4Clash ~[$[Stressed]^5] &  ~[[$[S2]]^4] & ~[?* [Stressed]^4 $[[Stressed]^2]] & ~[$[[Stressed]^2] [Stressed]^4 ?*] & ~[[$[[Stresse
```

Penalize 1 clash: 10 symbols
Penalize 2 clashes: 25 symbols
Penalize 3 clashes: 61 symbols
Penalize 4 clashes: 131 symbols

If multiple violations possible, must be defined as constraint family in Karttunen OT!

KARTTUNEN OT, SYLLABLES ONLY: ALIGN-X1-L FAMILY

Requires counting, doing arithmetic

```
define SWStar [Stressed [Weak]*];

define Alignx1L1 ~[AnySyllable SWStar];
define Alignx1L2 ~[AnySyllable Weak SWStar];
define Alignx1L3 ~[AnySyllable W2 SWStar] & ~[AnySyllable Stressed SWStar];
define Alignx1L4 ~[AnySyllable W2 Weak SWStar] & ~[AnySyllable Stressed Weak SWStar];
define Alignx1L5 ~[AnySyllable W2^2 SWStar] & ~[AnySyllable Stressed W2 SWStar] & ~[AnySyllable Weak Stressed SWStar];
define Alignx1L6 ~[AnySyllable Weak^5 SWStar] & ~[AnySyllable Stressed W2 Weak SWStar] & ~[AnySyllable Weak Stressed Weak SWStar] & ~[AnySy
define Alignx1L7 ~[AnySyllable Weak^6 SWStar] & ~[AnySyllable Stressed W2^2 SWStar] & ~[AnySyllable Weak Stressed W2 SWStar] & ~[AnySyllabl
define Alignx1L8 ~[AnySyllable Weak^7 SWStar] & ~[AnySyllable Stressed Weak^5 SWStar] & ~[AnySyllable Weak Stressed Weak^3 SWStar] & ~[AnyS

define Alignx1L9 ~[AnySyllable Weak^8 SWStar] & ~[AnySyllable Stressed Weak^6 SWStar] & ~[AnySyllable Weak Stressed Weak^4 SWStar] & ~[AnyS

define Alignx1L10 ~[AnySyllable Weak^9 SWStar] & ~[AnySyllable Stressed Weak^7 SWStar] & ~[AnySyllable Weak Stressed Weak^5 SWStar] & ~[Any

# Don't have more than n violations (up to 10)

define Alignx1Lg1 [Alignx1L1 & Alignx1L2 & Alignx1L3 & Alignx1L4 & Alignx1L5 & Alignx1L6 & Alignx1L7 & Alignx1L8 & Alignx1L9 & Alignx1L10];
define Alignx1Lg2 [Alignx1L2 & Alignx1L3 & Alignx1L4 & Alignx1L5 & Alignx1L6 & Alignx1L7 & Alignx1L8 & Alignx1L9 & Alignx1L10];
define Alignx1Lg3 [Alignx1L3 & Alignx1L4 & Alignx1L5 & Alignx1L6 & Alignx1L7 & Alignx1L8 & Alignx1L9 & Alignx1L10];
define Alignx1Lg4 [Alignx1L4 & Alignx1L5 & Alignx1L6 & Alignx1L7 & Alignx1L8 & Alignx1L9 & Alignx1L10];
define Alignx1Lg5 [Alignx1L5 & Alignx1L6 & Alignx1L7 & Alignx1L8 & Alignx1L9 & Alignx1L10];
define Alignx1Lg6 [Alignx1L6 & Alignx1L7 & Alignx1L8 & Alignx1L9 & Alignx1L10];
define Alignx1Lg7 [Alignx1L7 & Alignx1L8 & Alignx1L9 & Alignx1L10];
define Alignx1Lg8 [Alignx1L8 & Alignx1L9 & Alignx1L10];
define Alignx1Lg9 [Alignx1L9 & Alignx1L10];
```

KARTTUNEN OT, SYLLABLES ONLY: ALIGN-X1-L FAMILY

Requires counting, doing arithmetic

```
define SWStar [Stressed [Weak]*];

define Alignx1L1 ~[AnySyllable SWStar];
define Alignx1L2 ~[AnySyllable Weak SWStar];
define Alignx1L3 ~[AnySyllable W2 SWStar] & ~[AnySyllable Stressed SWStar];
define Alignx1L4 ~[AnySyllable W2 Weak SWStar] & ~[AnySyllable Stressed Weak SWStar];
define Alignx1L5 ~[AnySyllable W2^2 SWStar] & ~[AnySyllable Stressed W2 SWStar] & ~[AnySyllable Weak Stressed SWStar];
define Alignx1L6 ~[AnySyllable Weak^5 SWStar] & ~[AnySyllable Stressed W2 Weak SWStar] & ~[AnySyllable Weak Stressed Weak SWStar] & ~[AnySy
define Alignx1L7 ~[AnySyllable Weak^6 SWStar] & ~[AnySyllable Stressed W2^2 SWStar] & ~[AnySyllable Weak Stressed W2 SWStar] & ~[AnySyllabl
define Alignx1L8 ~[AnySyllable Weak^7 SWStar] & ~[AnySyllable Stressed Weak^5 SWStar] & ~[AnySyllable Weak Stressed Weak^3 SWStar] & ~[AnyS

define Alignx1L9 ~[AnySyllable Weak^8 SWStar] & ~[AnySyllable Stressed Weak^6 SWStar] & ~[AnySyllable Weak Stressed Weak^4 SWStar] & ~[AnyS

define Alignx1L10 ~[AnySyllable Weak^9 SWStar] & ~[AnySyllable Stressed Weak^7 SWStar] & ~[AnySyllable Weak Stressed Weak^5 SWStar] & ~[Any

# Don't have more than n violations (up to 10)

define Alignx1Lg1 [Alignx1L1 & Alignx1L2 & Alignx1L3 & Alignx1L4 & Alignx1L5 & Alignx1L6 & Alignx1L7 & Alignx1L8 & Alignx1L9 & Alignx1L10];
define Alignx1Lg2 [Alignx1L2 & Alignx1L3 & Alignx1L4 & Alignx1L5 & Alignx1L6 & Alignx1L7 & Alignx1L8 & Alignx1L9 & Alignx1L10];
define Alignx1Lg3 [Alignx1L3 & Alignx1L4 & Alignx1L5 & Alignx1L6 & Alignx1L7 & Alignx1L8 & Alignx1L9 & Alignx1L10];
define Alignx1Lg4 [Alignx1L4 & Alignx1L5 & Alignx1L6 & Alignx1L7 & Alignx1L8 & Alignx1L9 & Alignx1L10];
define Alignx1Lg5 [Alignx1L5 & Alignx1L6 & Alignx1L7 & Alignx1L8 & Alignx1L9 & Alignx1L10];
define Alignx1Lg6 [Alignx1L6 & Alignx1L7 & Alignx1L8 & Alignx1L9 & Alignx1L10];
define Alignx1Lg7 [Alignx1L7 & Alignx1L8 & Alignx1L9 & Alignx1L10];
define Alignx1Lg8 [Alignx1L8 & Alignx1L9 & Alignx1L10];
define Alignx1Lg9 [Alignx1L9 & Alignx1L10];
```

Gradient Align constraint statement symbol blowup!

DISCUSSION

Does reference to feet reduce the number of symbols used in `xfst`, in defining direct approach and Karttunen OT grammars for stress patterns in Samoan monomorphs?

- Surprisingly, except for Karttunen syllable OT account, size of grammars very similar.
  - Direct foot: 141 symbols
  - Direct syllable: 145 symbols
  - Karttunen OT foot: 335 symbols
  - Karttunen OT syllable: blowup!!
- Within the Karttunen OT formalism, reference to feet does make the grammar more succinct
- But within “direct” approach, reference to feet does not make grammar more succinct

CONCLUSION

Do constituents make phonological grammars for Samoan word stress more succinct?

- Strongly dependent on "grammar formalism", e.g. direct account vs. Karttunen OT (also, vs. violation-transducing OT)
- Here, exploration very preliminary; not clear that counting symbols right way to assess how well capturing generalizations
  - Grammars defined for direct accounts referring to only syllable, or also to feet, almost identical in size
    - But clearly more structure in the grammar referring to feet
    - Without feet, require case-by-case stipulations in grammar

APPENDIX: CONVENTIONS FOR SYMBOL COUNTING

Conventions for xfst expressions and symbol counting

a. In a `define` command, regular expressions are always delimited by square brackets

b. Auxiliary terms are defined for any regular expression that appears more than once in the set of transduction rules

c. A conjunct or disjunct which is longer than one symbol is enclosed in brackets

d. Semicolons ending a line counts as a symbol; spaces do not count

e. A character enclosed in double quotes, e.g. `"("` counts as one symbol, a number counts as one symbol

f. Each variable name, command, operator, and atomic expression counts as a single symbol: `define`, `Heavy`, `WeakLight`, etc., `?`, `*`, `+`, `^`, `.#.`, `[`, `]`, `(`, `)`, `|`, `\&`, `~`, `\`, `$`, `->`, `,`, `=>`, `->@ _`, `...`, `.o.`, `.O.` are the symbols that appear in our transducers.

g. Symbols that are used in the specification of the input (GEN) do not contribute to the symbol count; transductions that introduce foot structure do.