# Generalizations and conspiracies

Edward Stabler (UCLA)
Kristine M. Yu (UMass Amherst)

**MG+1 2023**

Some ideas emerging from Stabler&Yu 2023, etc,
on specifying and implementing MGs and interfaces

these slides: https://github.com/epstabler/star



2023-10-08

Generalizations and conspiracies
Edward Stabler (UCLA)
Kristine M. Yu (UMass Amherst)
MG+1 2023
Some ideas emerging from Stabler&Yu 2023, etc,
on specifying and implementing MGs and interfaces
these slides: https://github.com/epstabler/star

. . . when faced with different concerns, we should separate them as completely as possible, and deal with them in turn.
EWD636

You see, while we all know that unmastered complexity is at the root of the misery, we do not know what degree of simplicity can be obtained, nor to what extent the intrinsic complexity of the whole design has to show up in the interfaces.
EWD1304-2

## Taming the complexity

0.1 MG syntax + OT prosody (Stabler&Yu'23)
0.2 MG syntax with agreement (Ermolaeva&Kobele'23)
0.3 MG syntax + logical transduction (Vu,De Santo&Dolation'22)
0.4 Incremental comprehension, production (Hunter&al'19)
0.5 MG syntax with head movement (Stanojević'17)
. . .

- all these are difficult to understand, combinations more so!

⇒ **Factoring: Discover (additional) modularity**



2023-10-08

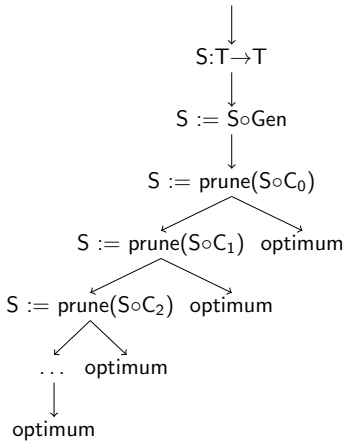
Taming the complexity

Taming the complexity
0.1 MG syntax + OT prosody (Stabler&Yu'23)
0.2 MG syntax with agreement (Ermolaeva&Kobele'23)
0.3 MG syntax + logical transduction (Vu,De Santo&Dolation'22)
0.4 Incremental comprehension, production (Hunter&al'19)
0.5 MG syntax with head movement (Stanojević'17)
. . .
• all these are difficult to understand, combinations more so!
⇒ Factoring: Discover (additional) modularity

When talking to Chomsky and Chomskians:
concern that MGs do not recognize and pull out
internal-external merge similarities, labeling, linearization.

They are right.
There have been some proposals, but best way has been unclear.
Making these basics easy will reveal/enable further extensions.

# Bennett&al Irish prosody




syntax

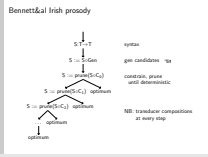
gen candidates ☜

constrain, prune
until deterministic

NB: transducer compositions
at every step



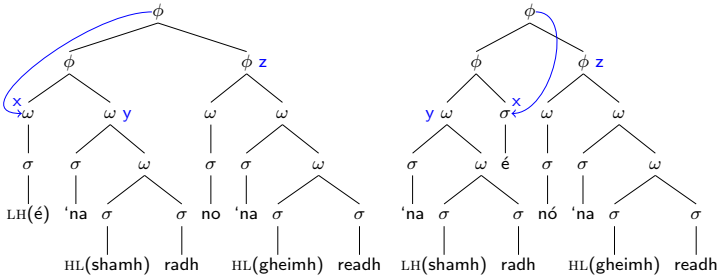




Bennett&al propose that an optimal prosodic structure is determined not just by syntax, but by constraints sensitive to phonological features like stress and accent. And in some cases, the optimal prosodic structure is even one that modifies linear order.

Stabler&Yu capture the Bennett&al analysis like this:
Initially, syntax is composed from parts, but here
we can think of it as an identity transduction on a single tree,
from which gen generates possible candidates, including some with modified linear order; then optimal structure selected
by evaluating constraints in rank order

**The operations here apply not to particular structures, but to finite transducers, each generating possibly infinitely many structures**



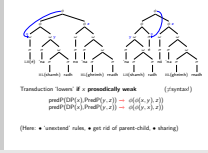

Transduction 'lowers' **if** $x$ **prosodically weak** (≠syntax!)

$$\text{predP}(\text{DP}(x),\text{PredP}(y,z)) \rightarrow \phi(\phi(x,y),z))$$
$$\text{predP}(\text{DP}(x),\text{PredP}(y,z)) \rightarrow \phi(\phi(y,x),z))$$

(Here: • 'unextend' rules, • get rid of parent-child, • sharing)







Note!! • the trigger is not syntactic, • the effect is not one syntax could achieve, and • that effect has no semantic consequences.
I.E. This is not syntax!

The SCiL paper/talk gets to here,
but lacks space expose ingredients in full generality.

## Unextend for transducer composition

$$\text{predP}(\text{DP}(\text{q0}(x)),\text{PredP}(\text{q1}(y),\text{q2}(z)))) \rightarrow \text{q3}(\phi(\phi(x,y),z)))$$

**Engelfriet&al: For composition, compile to '1-normal form':**

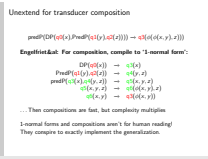
$$\begin{array}{rcl}
\text{DP}(\text{q0}(x)) & \rightarrow & \text{q3}(x) \\
\text{PredP}(\text{q1}(y),\text{q2}(z)) & \rightarrow & \text{q4}(y,z) \\
\text{predP}(\text{q3}(x),\text{q4}(y,z)) & \rightarrow & \text{q5}(x,y,z) \\
\text{q5}(x,y,z) & \rightarrow & \text{q6}(\phi(x,y),z) \\
\text{q6}(x,y) & \rightarrow & \text{q3}(\phi(x,y))
\end{array}$$

. . . Then compositions are fast, but complexity multiplies

1-normal forms and compositions aren't for human reading!
They conspire to exactly implement the generalization.








At SCiL, we left the states out of the rules, added back in red.
This 1 rule has 3 input symbols, 2 output symbols, and 4 states.
(Not too complicated – syn/prosody research can work at this level)

But we implement that rule with a set of simpler rules.
To handle complex transducer i/o: 1-normal form.

Then to compose, intuitively: when first transducer outputs a symbol,
it can immediately be given to second transducer.
(multiplies rules. . . implementation-oriented, only for psycholinguists & al)

NB: No one of these 1-normal rules can be justified independently of the others. It is truly the extended rule (= the whole conspiracy of 1-normal rules) that is supported by the evidence.

## Factoring the tree transductions

Standard tree automata not quite what we need, so dissect a bit. . .

Tree automata recursively 'fold' classification, construction steps through tree traversal (e.g. Bahr'12)

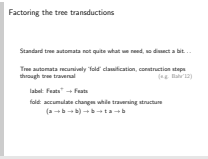
label: $\text{Feats}^{+} \rightarrow \text{Feats}$

fold: accumulate changes while traversing structure

$(\text{a} \rightarrow \text{b} \rightarrow \text{b}) \rightarrow \text{b} \rightarrow \text{t a} \rightarrow \text{b}$








Tree transducers not quite what we need. . .

First, consider what they are

Haskell gives this type for its 'right' fold,
but fold is really a family of higher order functions

## Refactor and adjust 1: *-extensions

- Multiple successors: left, right, etc?
  But then how to express unranked generalizations?
- Better: replace (parent,child) relation with (parent, next gen) where next gen is unbounded.
  E.g. `data Tree a = Node a [Tree a]`,
  with regular *-sequence [Tree a].
- OK, but some components still too complex:
  esp. labeling with k 'movers' to emulate sharing





With *-extension, the parent-child (successor) relation is no longer basic

Refactor and adjust 1: *-extensions
- Multiple successors: left, right, etc?
  But then how to express unranked generalizations?
- Better: replace (parent,child) relation with (parent, next gen) where next gen is unbounded.
  E.g. data Tree a = Node a [Tree a],
  with regular *-sequence [Tree a].
- OK, but some components still too complex:
  esp. labeling with k 'movers' to emulate sharing

## Refactor and adjust 2: DAG transducers

Labeling simplified when sharing is real: not trees, DAGs

⇒ Split structure up, for arity k labels
(Fowlie'11)

Fold generalizes to DAGs (Priese'07, Bahr&Axelsson'17)
E.g. `data Dag a = {(Node a, [Edges])}`

DAG automata recursively 'fold' classification, construction steps through graph traversal

Weighted search methods extend to *-DAG grammars
(Höfner&Möller'12, Kidney&Wu'21, Gibbons&al'22)





Refactor and adjust 2: DAG transducers
Labeling simplified when sharing is real: not trees, DAGs
⇒ Split structure up, for arity k labels
(Fowlie'11)
Fold generalizes to DAGs (Priese'07, Bahr&Axelsson'17)
E.g. data Dag a = {(Node a, [Edges])}
DAG automata recursively 'fold' classification, construction steps through graph traversal
Weighted search methods extend to *-DAG grammars
(Höfner&Möller'12, Kidney&Wu'21, Gibbons&al'22)

# In progress

- **Appropriate modularity** over
  **structures with empirical support**
  - decreases specification complexity (better theory!), and
  - exposes intermodular dependencies,
    some of which fold through structures recursively,
  - suggests specialized automata for psych modeling
- High-level generalizations → lower-level conspiracies:
  multiple processses, different granularities, in concert
- Tree/dag transducer composition 'platform', in prep –
  a library with functions for basic fold, labeling, composition ops



2023-10-08

└ In progress

In progress

- **Appropriate modularity** over
  **structures with empirical support**
  - decreases specification complexity (better theory!), and
  - exposes intermodular dependencies,
    some of which fold through structures recursively,
  - suggests specialized automata for psych modeling
- High-level generalizations → lower-level conspiracies:
  multiple processes, different granularities, in concert
- Tree/dag transducer composition 'platform', in prep –
  a library with functions for basic fold, labeling, composition ops

Conspiracies happen, but not in the theoretical statement,
which is optimized for simplicity, intelligibility,
and clear relevance to evidential support/learnability

Patrick Bahr. 2012.
Modular tree automata.
In *Mathematics of Program Construction*, pages 263–299. Springer LNCS 7432.

Patrick Bahr and Emil Axelsson. 2017.
Generalizing tree traversals and tree transformations to DAGs: Exploiting sharing without the pain.
*Science of Computer Programming*, 137.

Ryan Bennett, Emily Elfner, and James McCloskey. 2016.
Lightest to the right: An apparently anomalous displacement in Irish.
*Linguistic Inquiry*, 47(2):169–234.

Joost Engelfriet, Eric Lilin, and Andreas Maletti. 2009.
Extended multi bottom–up tree transducers: Composition and decomposition.
*Acta Informatica*, 46(8):561–590.

Marina Ermolaeva and Greg Kobele. 2023.
Formal properties of agreeing minimalist grammars.
Society for Computation in Linguistics 2023.

Meaghan Fowlie. 2011.
Multidominant minimalist grammars.
UCLA Master's thesis.

Jeremy Gibbons, Donnacha Oisín Kidney, Tom Schrijvers, and Nicolas Wu. 2022.
Breadth-first traversal via staging.
In *Mathematics of Program Construction, Procs 14th International Conference*. Springer LNCS 13544.

Peter Höfner and Bernhard Möller. 2012.
Dijkstra, Floyd and Warshall meet Kleene.
*Formal Aspects of Computing*, 24:459–476.

Tim Hunter, Miloš Stanojević, and Edward P. Stabler. 2019.
The active-filler strategy in a move-eager left-corner minimalist grammar parser.
In *Procs. Workshop on Cognitive Modeling and Computational Linguistics*, pages 1–10.



Dimitrios Kalemis. 2014.
Trees as monads.
*Blog post.*

Donnacha Oisín Kidney and Nicolas Wu. 2021.
Algebras for weighted search.
*Procs. of the ACM on Programming Languages*, 5(72).

Lutz Priese. 2007.
Finite automata on unranked and unordered DAGs.
In *Procs 11th International Conference, Developments in Language Theory*. Springer LNCS 4588.

Edward P. Stabler and Kristine M. Yu. 2023.
Unbounded recursion in two dimensions, where syntax and phonology meet.
Society for Computation in Linguistics 2023.

Miloš Stanojević. 2017.
Minimalist grammar transition-based parsing.
In *International Conference on Logical Aspects of Computational Linguistics, LACL*, pages 273–290.
Springer LNCS 10054.

Mai Ha Vu, Aniello De Santo, and Hossep Dolatian. 2022.
Logical transductions for the typology of ditransitive prosody.
In *Procs SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*.