Formalizing Agree*
Part I: Complexity

Dan Milway
University of Toronto

12 August 2015

1 Introduction

- Much, maybe most, work in formal syntax now includes two operations in the narrow syntax:
  - Merge
  - Agree
- While Merge has been analysed from a formal perspective (Collins 2014), Agree has not.
- Since Agree is the de facto workhorse in many analyses we should understand its properties.
- An analysis of Agree shows that a narrow syntax that includes Agree is in a higher complexity class than one that excludes it.

1.1 Plan

- Starting point and assumptions (largely from Collins and Stabler 2011)
- Defining Agree
- Assessing Agree's complexity
- Conclusion

2 Assumptions

2.1 Collins and Stabler (2011)

2.1.1 The Grammar

Definition 1. Universal Grammar
Universal Grammar (UG) is a 6-tuple: ⟨PHON, SYN, SEM, Select, Merge, Transfer⟩

- PHON, SYN, and SEM are sets of universal features.
- Select, Merge, and Transfer are operations.

Definition 2. An I-Language is a pair ⟨LEX, UG⟩

- The Lexicon, LEX, is a set of lexical items

Definition 3. A *lexical item* is a triple of three sets of features,
LI = ⟨Sem, Syn, Phon⟩
where Sem ⊂ SEM, Syn ⊂ SYN, Phon ⊂ PHON.

*Presented at the 2015 Dog Days Syntax Workshop at The University of Toronto.

2.1.2 The Derivation

- A derivation is a sequence of stages.

Definition 4. A *stage (of a derivation)* is a pair S = $\langle$LA, W$\rangle$, where LA is a lexical array and W is a set of syntactic objects called the *workspace* of S.

Definition 5. A *lexical array* is a finite set of lexical item tokens.

Definition 6. A *lexical item* token LI_k is a pair $\langle$LI, k$\rangle$ where LI is a lexical item and k is an integer.

- The inclusion of indicies is to distinguish copies from tokens.
  - *He$_j$ saw him$_{i/*j}$*.
- The workspace is a set of syntactic objects

Definition 7. X is a *syntactic object* iff
  i X is a lexical item token, or
  ii X is a set of syntactic objects.

- One stage (S_1) derives another (S_2) by either Merge or Select.

Definition 8. Let X and Y be two (distinct) syntactic objects.
Merge(X,Y) = $\{$X, Y$\}$.

Definition 9. Let S be a stage in a derivation S = $\langle$LA, W$\rangle$.
If $\mathrm{LI}_k \in$ LA, then Select(LI, S) = $\langle \mathrm{LA} - \{\mathrm{LI}\}, \mathrm{W} \cup \{\mathrm{LI}\} \rangle$

- Operations, then, can be understood as relations between stages of a derivation.

(1) Derivation of "John fell"
$\mathrm{S0} = \langle \{past_1, \mathrm{John}_2, \mathrm{fall}_3\}, \{\} \rangle$
$\mathrm{S1} = \langle \{past_1, \mathrm{John}_2\}, \{\mathrm{fall}_3\} \rangle$
$\mathrm{S2} = \langle \{past_1\}, \{\mathrm{John}_2, \mathrm{fall}_3\} \rangle$
$\mathrm{S3} = \langle \{past_1\}, \{\{\mathrm{John}_2, \mathrm{fall}_3\}\} \rangle$
$\mathrm{S4} = \langle \{\}, \{past_1, \{\mathrm{John}_2, \mathrm{fall}_3\}\} \rangle$
$\mathrm{S5} = \langle \{\}, \{\{past_1, \{\mathrm{John}_2, \mathrm{fall}_3\}\}\} \rangle$
$\mathrm{S6} = \langle \{\}, \{\{\mathrm{John}_2, \{past_1, \{\mathrm{John}_2, \mathrm{fall}_3\}\}\}\} \rangle$

- S0 and S1 are related by Select(fall_3)
- S2 and S3 are related by Merge(John_2, fall_3)
- S5 and S6 are related by Merge($\{past_1, \{\mathrm{John}_2, \mathrm{fall}_3\}\}$, John_2)

2.2 Equality and Identity matter

- Close only counts in horseshoes and handgrenades.
- Partial similarity is not good enough for identity in syntax
  - *she* $\neq$ *her* even though they differ only by case.
  - *They like you* $\neq$ *They liked you* even though they differ only in tense.
  - $\{\mathrm{A}, \{\mathrm{B}, \mathrm{C}\}\} \neq \{\mathrm{A}, \{\mathrm{B}', \mathrm{C}\}\}$ even though they only differ by a $'$ symbol.
- If two objects (*e.g.*, lexical items, stages, derivations, features) differ even slightly, then they are distinct.
- There are no degrees of distinctness.

2.3 Upwards or downwards doesn't matter

- I assume that Standard (upwards valuing) and Reverse (downwards valuing) Agree are equivalent. [1]
- I use Reverse Agree (Wurmbrand 2012) only for expository purposes.

3 Defining Agree

3.1 Features

- Since Agree is usually taken to operate between features, we need to formalize features.
- The standard way to represent features is [uF_], [iF*val*], *etc.* (Wurmbrand 2012; Pesetsky and Torrego 2007; Zeijlstra 2012)

Definition 10. A Feature F is a triple $\langle \tau, f, v \rangle$ where,

i $\tau \in \{0, 1\}$ is an interpretability value,

ii f is a dimension (*e.g.*, person, tense, *etc.*), and

iii v is a value (=$\emptyset$ if unvalued).

- $\langle 0, \pi, \emptyset \rangle$ represents an uninterpretable unvalued person feature.
- $\langle 1, \mathrm{T}, pst \rangle$ represents an interpretable past tense feature.

3.2 Agree

- Agree is usually taken to be a compound operation (Probe, Match, Value).
  - Probe can be stated as a structural condition (X must c-command/be c-commanded by Y)
  - Match can be stated as an identity condition ($\mathrm{F_Y} = \langle 0, f, \emptyset \rangle$ where $\mathrm{F_X} = \langle 1, f, val \rangle$)
  - The two can be combined to make an Agree relation (Agree(X, Y, F) iff Probe(X, Y) and Match($\mathrm{F_X}$, $\mathrm{F_Y}$)
- Value makes Agree and operation.

Definition 11. Value($\langle \tau, f, v \rangle$, $\langle$LI, k$\rangle$)=
$\langle$LJ, k$\rangle$ such that
$\mathrm{Syn_{LJ}} = (\mathrm{Syn_{LI}} - \{\langle \rho, f, \emptyset \rangle\}) \cup \{\langle \rho, f, v \rangle\}$

- Feature valuation replaces an unvalued feature with a valued feature ($\langle \rho, f, \emptyset \rangle \neq \langle \rho, f, v \rangle$).
- This requires replacing one Syn set with another ($\mathrm{Syn_{LI}} \neq \mathrm{Syn_{LJ}}$).
- Since lexical items are defined as triples of sets of features, two lexical items with distinct Syn sets are distinct (LI $\neq$ LJ).
- Since a lexical item token can have multiple occurrences in a given SO, valuation must replace each occurrence of LI with an occurrence of LJ.

[1] A logically possible version of Agree that doesn't seem to increase complexity is one in which Agree is restricted to valuing features on not yet merged heads. Agree would only occur before external Merge.

3.3 Agree in Derivations

Definition 12. UG_{Agree} is the 7-tuple $\langle$PHON, SYN, SEM, Select, Merge, Agree, Transfer$\rangle$

- If Agree is an operation like Merge and Select, it should be expressable as relation between stages.
- Consider two sequential stages derived by Agree.

(2) **Agree(X, $\langle \tau, \mathbf{f}, \mathbf{v}\rangle$)(S, S′)**
$S = \langle \text{LA}, \{\text{W}, \{\mathbf{X}, \{\text{Y}, \{\mathbf{X}, \text{Z}\}\}\}\}\rangle$
$S' = \langle \text{LA}, \big\{\text{W}, \big\{\mathbf{X}', \big\{\text{Y}, \big\{\mathbf{X}', \text{Z}\big\}\big\}\big\}\big\}\rangle$

- While, Merge and Select can be defined as simple set-theoretic statements, Agree must be defined as an algorithm that traverses the the workspace and replaces all instances of X with Value($\langle \tau, f, v\rangle$, X) or. . .

Definition 13. [2] For stages S^D_n, S^D_{n+1} in derivation D in UG_{Agree},
Agree(X, $\langle \tau, f, v\rangle$)($S^D_n$, S^D_{n+1}) iff
There is some derivation E in UG such that
$\text{LA}^E_1 = (\text{LA}^D_1 - \{\text{X}\}) \cup \{\text{Value}(\langle \tau, f, v\rangle, \text{X})\}$ and
$S^E_n = S^D_{n+1}$

- Agree, at stage n, is equivalent to an n-stage (partial) derivation using only Merge and Select.

4 A measure of complexity: Big-O

- Big-O notation expresses how a function reacts to the size of its input
- Commonly used to describe the runtime of an algorithm.
  - O(1): constant time. Runtime does not vary with input size.
    e.g. testing if the a number is even.
  - O(n): linear time. Runtime increases proportionally with the size of its input.
    e.g. searching an unsorted array.
  - O(n^2): quadratic time. Runtime increases proportionally to the square of the size of its input.
    e.g. many sorting algorithms
- The worst case scenario is used to assess runtime.
  - For a search algorithm: the case in which the desired object is the very last object in an array.
  - For a sorting algorithm: the case in which an array is sorted in reverse order.
- If we take the number of terminal nodes in a tree as n,
  - Merge and Select are O(1)
    * Selecting an LI_k from a 10-member LA takes as much time as Selecting an LI_k from a 20-member LA.
    * Merging an LI_k with an n=10 tree takes as long as Merging an LI_k with an n=20 tree.
  - A derivation in UG is O(n)
    * A derivation of an n-sized tree is ∼n iterations of Merge and Select
  - Agree in an n-sized tree, by hypothesis, is equivalent to UG derivation of that tree, therefore O(n)
  - Consider a derivation of an n-sized tree in which every instance of Merge is accompanied by an instance of Agree.[3]

[2] This definition is slightly imprecise. At stage S_n of a derivation, n = m+s+a+t, where m, s, a, and t are equal to the number of iterations of Merge, Select, Agree, and Transfer, repectively, up to that point. So, $S^E_{m+s+t} = S^D_{m+s+a+t+1}$.

[3] As is typical of a crash-proof syntax

        * The first Agree will be on a 2-element tree → 1 iteration of Merge to re-derive.
        * The second will be on a 3-element tree → 2 iterations of Merge to re-derive.
        * And so on...
        * To derive an n-element tree the number of iterations of Merge that occur in re-derivations is: $1+2+3+\cdots+(n-2)+(n-1) =$ $\sum_{i=1}^{n-1} k = \frac{(n-1)(n)}{2}$ (The sum is a *triangular* number (Weisstein n.d.))
    - A derivation of this type belongs to the class of O(n^2) algorithms.
- Derivations in UG_{Agree} are in a higher complexity class than those of UG.

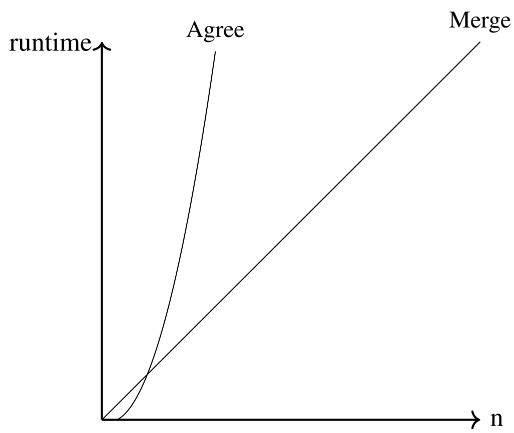

Figure 1: The runtimes of derivations under UG vs UG+Agree as a function of the size of the SO derived

5 Conclusion and questions

- Whether or not we think we can eliminate Agree from the narrow syntax, we can't ignore its complexity.
- Feature valuation as a part of Transfer does not seem have the same kind of increase in complexity.

5.1 Some questions for Parts II – N

- Are all the proposed versions of Agree even formulable in this system?
- How do we model feature geometries? Articulated probes? Cyclic Agree?

References

Collins, Chris (2014). “Merge$(X,Y) = \{X,Y\}$”. In: *Labels and Roots*. Ed. by Leah Bauke, Andreas Blümel, and Erich Groat. De Gruyter Mouton. URL: http://ling.auf.net/lingbuzz/002186. Submitted.

Collins, Chris and Edward Stabler (2011). “A formalization of minimalist syntax”. ms., NYU and UCLA.

Epstein, Samuel David, Hisatsugu Kitahara, and T. Daniel Seely (2011). “Derivation(s)”. In: *The Oxford Handbook of Linguistic Minimalism*. Ed. by Cedric Boeckx. Oxford University Press.

Pesetsky, David and Esther Torrego (2007). “The syntax of valuation and the interpretability of features”. In: *Phrasal and clausal architecture: Syntactic derivation and interpretation*. Ed. by Simin Karimi, Vida Samiian, and Wendy K. Wilkins. Amsterdam: John Benjamins, pp. 262–294.

Weisstein, Eric W. *Triangular Number From MathWorld–A Wolfram Web Resource*. Last visited 7 August 2015. URL: http://mathworld.wolfram.com/TriangularNumber.html.

Wurmbrand, S. (2012). “The Merge Condition: A syntactic approach to selection”. In: *To appear. Minimalism and Beyond: Radicalizing the interfaces*. Ed. by Peter Kosta et al. Amsterdam: John Benjamins.

Zeijlstra, Hedde (2012). “There is only one way to agree”. In: *The Linguistic Review* 29.3, pp. 491–539.