

1 The Grammar

SCOPE

$$m \parallel n := \begin{cases} m n & \text{if } m :: \alpha \rightarrow \beta, n :: \alpha \\ \lambda k. m(\lambda f. n(\lambda x. k(f \parallel x))) & \text{otherwise} \end{cases} \quad m \parallel n := \begin{cases} n m & \text{if } n :: \alpha \rightarrow \beta, m :: \alpha \\ \lambda k. m(\lambda x. n(\lambda f. k(x \parallel f))) & \text{otherwise} \end{cases} \quad m \parallel n := \begin{cases} \lambda x. m x \wedge n x & \text{if } n :: \alpha \rightarrow \beta, m :: \alpha \\ \lambda k. m(\lambda x. n(\lambda f. k(f \parallel x))) & \text{otherwise} \end{cases}$$

BINDING

$$\eta x := \lambda gh. \{\langle x, g, h \rangle \mid x \neq \mathbf{F}\} \quad m^\star := \lambda kgh. \bigcup \{k x g' h' \mid \langle x, g', h' \rangle \in m g h\} \quad m^\perp := m \eta \quad x^\uparrow := (\eta x)^\star = \lambda k. k x \quad m^{\triangleright u} := m^\star (\lambda xgh. \{\langle x, g^{u \mapsto x}, h \rangle\})$$

POSTSUPPOSITIONS

$$m^\sharp := (\lambda gh. \{\langle x, g', h \rangle \mid \langle \cdot, \cdot, h' \rangle \in G, \langle x, g', \cdot \rangle \in h' G\})^\star, \quad \text{where } G = m^\perp g \text{ id} \\ \text{true}_g m := \exists \langle \cdot, \cdot, h \rangle \in G. h G \neq \emptyset, \quad \text{where } G = m^\perp g \text{ id}$$

- Postsups are evaluated at reset boundaries and at closing time. A computation is true at an input g if it has a successful output *after its postsups have applied*. A computation is not fully reset until its postsuppositions have each had a whack at the set of outputs, at which point the survivors are collected, and the postsups flushed.

- Following Brasoveanu 2012, contexts are split into two components, an assignment function, and a “postsupposition”. Postsups here have the recursive type $C \equiv \{\alpha * \gamma * C\} \rightarrow \{\alpha * \gamma * C\}$, the type of filters on sets of outputs.
- The dynamic machinery now bears an intriguing resemblance to something Wadler (1994) calls the InputOutput Monad. Assignments play the role of input, postsups that of outputs, in the sense that they are simply accumulated (composed) over the course of the computation, and used to post-process the final result.

2 Basic Definite Descriptions

circle := circ

square := sq

in := in

$$\mathbf{the}_u := \lambda ckg h. \bigcup \{k x g' h' \mid \langle \mathbf{T}, g', h' \rangle \in c x g^{u \mapsto x} h^{\mathbf{1}_u}\}$$

$$\mathbf{1}_u \equiv \lambda G. \begin{cases} G & \text{if } |\{g u \mid \langle \cdot, g, \cdot \rangle \in G\}| = 1 \\ \emptyset & \text{otherwise} \end{cases}$$

RESET

$$\llbracket \text{the square} \rrbracket \rightsquigarrow \left(\frac{\mathbf{the}_v(\lambda y. \llbracket \square \rrbracket)}{y} \parallel \frac{\llbracket \square \rrbracket}{\mathbf{square}} \right)^{\downarrow \sharp} \rightsquigarrow (\mathbf{the}_v(\lambda y g. \{\langle \text{sq } y, g \rangle\}))^\sharp \\ \rightsquigarrow \left(\frac{\lambda gh. \bigcup \{\llbracket \square \rrbracket g^{v \mapsto y} h^{\mathbf{1}_u} \mid \text{sq } y\}}{y} \right)^\sharp \rightsquigarrow \frac{\lambda gh. |G_v| = 1. \bigcup \{\llbracket \square \rrbracket g^{v \mapsto y} h \mid \text{sq } y\}}{y}$$

- **the** _{u} is just **a** _{u} plus a uniqueness postsup, which restricts what the *global set of outputs* is allowed to look like: they must all agree on the value of u .
- Thus the uniqueness effect associated with **the** is effectively *delayed* until some program containing it is evaluated, similar to the way that other cardinality impositions have been argued to operate (Brasoveanu 2012, Henderson 2014).

- Note that resetting $\llbracket \text{the square} \rrbracket$ in the last reduction step here has no effect on its semantic shape, because it’s essentially $\llbracket \text{a square} \rrbracket$.
- But it does fix its postsup (which I’ve switched to representing as a *presup*, since it now just constrains the input). For any incoming g , the G of $\mathbf{1}_u^G$ will be equal to $\{\langle y, g^{v \mapsto y}, \mathbf{1}_u \rangle \mid \text{sq } y\}$. Nothing from this set will survive the $\mathbf{1}_u$ filter unless all the assignment funcs in G happen to map v to the same square. That is, unless there’s exactly one square available to assign v to in the first place.

ABSOLUTE DEFINITE READING

[[the circle in the square]]

$$\begin{aligned}
 & \left(\frac{\mathbf{the}_u(\lambda x. [])}{x} \parallel \left(\frac{[]}{\mathbf{circle}} \parallel \frac{[]}{\mathbf{in}} \parallel \left(\frac{\mathbf{the}_v(\lambda y. [])}{y} \parallel \frac{[]}{\mathbf{square}} \right)^{\downarrow \uparrow} \right) \right)^{\downarrow \uparrow} \\
 & \left(\frac{\mathbf{the}_u(\lambda x. [])}{x} \parallel \left(\frac{[]}{\mathbf{circle}} \parallel \frac{[]}{\mathbf{in}} \parallel \frac{\lambda gh: |G_v| = 1. \cup \{[] g^{v \mapsto y} h \mid \text{sq } y\}}{y} \right) \right)^{\downarrow \uparrow} \\
 & \left(\frac{\mathbf{the}_u(\lambda xgh: |G_v| = 1. \cup \{[] g^{v \mapsto y} h \mid \text{sq } y\})}{\text{circ } x \wedge \text{in } yx} \right)^{\downarrow \uparrow} \\
 & \left(\frac{\lambda gh: |G_v| = 1. \cup \left\{ [] g^{u \mapsto x} h^{1_u} \mid \text{sq } y, \text{ circ } x, \text{ in } yx \right\}}{x} \right)^{\uparrow} \\
 & \frac{\lambda gh: |G_v| = |G'_u| = 1. \cup \left\{ [] g^{u \mapsto x} h \mid \text{sq } y, \text{ circ } x, \text{ in } yx \right\}}{x}
 \end{aligned}$$

RELATIVE DEFINITE (HADDOCK) READING

[[the circle in the square]]

$$\begin{aligned}
 & \left(\frac{\mathbf{the}_u(\lambda x. [])}{x} \parallel \left(\frac{[]}{\mathbf{circle}} \parallel \frac{[]}{\mathbf{in}} \parallel \left(\frac{\mathbf{the}_v(\lambda y. [])}{y} \parallel \frac{[]}{\mathbf{square}} \right)^{\downarrow} \right) \right)^{\downarrow \uparrow} \\
 & \left(\frac{\mathbf{the}_u(\lambda x. [])}{x} \parallel \left(\frac{[]}{\mathbf{circle}} \parallel \frac{[]}{\mathbf{in}} \parallel \frac{\lambda gh. \cup \{[] g^{v \mapsto y} h^{1_v} \mid \text{sq } y\}}{y} \right) \right)^{\downarrow \uparrow} \\
 & \left(\frac{\mathbf{the}_u(\lambda xgh. \cup \{[] g^{v \mapsto y} h^{1_v} \mid \text{sq } y\})}{\text{circ } x \wedge \text{in } yx} \right)^{\downarrow \uparrow} \\
 & \left(\frac{\lambda gh. \cup \left\{ [] g^{u \mapsto x} (h^{1_u})^{1_v} \mid \text{sq } y, \text{ circ } x, \text{ in } yx \right\}}{x} \right)^{\uparrow} \\
 & \lambda gh: |G_v| = |G_u| = 1. \cup \left\{ [] g^{u \mapsto x} h \mid \text{sq } y, \text{ circ } x, \text{ in } yx \right\} x
 \end{aligned}$$

- The inner definite is reset, freezing its presupposition as above. When the input assignment g is eventually inserted, we will have $G = \{\langle y, g^{v \mapsto y}, \mathbf{1}_v \rangle \mid \text{sq } y\}$, and the presupposition will guarantee that $g'v$ is constant across the outputs.
- As with the inner DP, the host DP's presupposition is fixed when it is reset. This time, we have $G' = \left\{ \left\langle x, g^{u \mapsto x}, \mathbf{1}_u \right\rangle \mid \text{sq } y, \text{ circ } x, \text{ in } yx \right\}$, where g is whatever the input happens to be. In particular, all the outputs will now need to agree on the value of u in addition to v , which will only be possible if there's exactly one circle in the square that all outputs assign to v .
- The only difference here is that we do not reset the inner DP, which staves off its postsup until more information is accumulated in its scope.
- So when the outer DP is eventually reset, it ends up fixing the postsup of both definites to the same set of potential outputs.
- For any input g , the set of outputs G that $\mathbf{1}_u$ and $\mathbf{1}_v$ will target is given below. All of the outputs in this set will map u to some circle and v to some square containing the circle they map to u .

$$G = \left\{ \left\langle x, g^{u \mapsto x}, \mathbf{1}_u^{1_v} \right\rangle \mid \text{sq } x, \text{ circ } y, \text{ in } yx \right\}.$$
- Requiring that there be exactly one such v and one such u is tantamount to requiring that there be exactly one square *that has a circle in it* and exactly one circle *in that square*. In other words, there should be exactly one pair $\langle x, y \rangle$ in $\text{circ} \times \text{sq}$ such that in yx .

3 Plurals

$$\begin{aligned} \text{-s} &:= \lambda Px. x \in \{\oplus P' \mid P' \subseteq P\} \\ \mathbf{the}_u &:= \lambda ckg h. \bigcup \{k x g' h' \mid \langle \mathbf{T}, g', h' \rangle \in c x g^{u \mapsto x} h^{1_u \circ \mathbf{M}_u}\} \\ \mathbf{M}_u &:= \lambda G. \{\langle \cdot, g, \cdot \rangle \in G \mid \neg \exists \langle \cdot, g', \cdot \rangle \in G. g' u \sqsupset g u\} \end{aligned}$$

- **-s** is a boilerplate plural morpheme that builds sums from the atoms in its complement.
- \mathbf{M}_u is a kind of maximization operator on outputs (Brasoveanu 2012, Charlow 2014). It filters out those assignments in $g \in G$ that are strictly dominated, in the sense that they assign u to a value that is a proper part of something assigned to u by some other $g' \in G$.

DERIVATIONS

circles, squares, etc. ...

4 DP-Internal Superlatives

$$\begin{aligned} \mathbf{older} &:= \lambda xy. \text{age } x > \text{age } y \\ \mathbf{the}_u &:= \lambda Mckgh. \bigcup \{k x g' h' \mid \langle \mathbf{T}, g', h' \rangle \in c x g^{u \mapsto x} h^{1_u \circ M}\} \mathbf{M}_u^f \\ \mathbf{est}_u &:= \lambda f. \mathbf{M}_u^f \\ \mathbf{oldest}_u &= \mathbf{est}_u \mathbf{older} = \lambda G. \{\langle \cdot, g, \cdot \rangle \in G \mid \neg \exists \langle \cdot, g', \cdot \rangle \in G. \text{age } (g' u) > \text{age } (g u)\} \end{aligned} \quad := \lambda G. \{\langle \cdot, g, \cdot \rangle \in G \mid \neg \exists \langle \cdot, g', \cdot \rangle \in G. f(g' u)(g u)\}$$

- **est** abstracts over the ordering function that \mathbf{M} uses of pluralities, \mathbf{M}_u filters away any output that assign have (ie., a smaller sum than one of the other outputs). \mathbf{M}_u filters outputs that assign u to a *younger* individual.
- And now here's the swim move: **the** absorbs the max for the ride. "Absolute" definite become absolute superlatives. This part is my favorite.

SUPERLATIVE RESET

[[the oldest squirrel]]

$$\begin{aligned} &\sim \left(\frac{\mathbf{the}_v \mathbf{M}_v^{\text{ag}} (\lambda y. [\])}{y} \parallel \frac{[\]}{\mathbf{squirrel}} \right)^{\downarrow \uparrow} \sim (\mathbf{the}_v \mathbf{M}_v^{\text{ag}} (\lambda ygh. \{\langle \text{sq } y, g, h \rangle\}))^{\uparrow} \\ &\sim \left(\frac{\lambda gh. \bigcup \{[\] g^{v \mapsto y} h^{1_v \circ \mathbf{M}_v^{\text{ag}}} \mid \text{sq } y\}}{y} \right)^{\uparrow} \sim \frac{\lambda gh. |G_v| = 1. \bigcup \{[\] g^{v \mapsto y} h \mid \text{sq } y, \forall z: \text{sq}. \neg \text{older } z y\}}{y} \end{aligned}$$

- Resetting the superlative DP has two effects: (1) it fixes the set of individuals that the superlative operator \mathbf{M} compares, in this case squirrels with respect to age; and (2) it freezes the postsup associated with the definite article.
- Here $G = \{\langle y, g^{v \mapsto y}, \mathbf{1}_v \circ \mathbf{M}_v \rangle \mid \text{sq } y, \forall z: \text{sq}. \neg \text{older } z y\}$. This set could in principle have multiple winners (multiple squirrels with the same greatest age), but the definite determiner rules that out; all the $g \in G$ need to point v to *the same* squirrel.

ABSOLUTE DP-INTERNAL SUPERLATIVES

[[the circus with the oldest squirrel]]

$$\begin{aligned}
 & \left(\frac{\mathbf{the}_u(\lambda x. [])}{x} \parallel \left(\frac{[]}{\mathbf{circus}} \parallel \frac{[]}{\mathbf{with}} \parallel \left(\frac{\mathbf{the}_v \mathbf{M}_v^{\text{ag}}(\lambda y. [])}{y} \parallel \frac{[]}{\mathbf{squirrel}} \right)^{\downarrow \uparrow} \right)^{\downarrow \uparrow} \right)^{\downarrow \uparrow} \\
 & \left(\frac{\mathbf{the}_u(\lambda x. [])}{x} \parallel \left(\frac{[]}{\mathbf{circus}} \parallel \frac{[]}{\mathbf{with}} \parallel \frac{\lambda gh: |G_v| = 1. \cup \left\{ [] g^{v \mapsto y} h \mid \text{sq } y, \forall z: \text{sq}. \neg \text{older } z y \right\}}{y} \right)^{\downarrow \uparrow} \right)^{\downarrow \uparrow} \\
 & \left(\frac{\mathbf{the}_u(\lambda xgh: |G_v| = 1. \cup \{ [] g^{v \mapsto y} h \mid \text{sq } y, \forall z: \text{sq}. \neg \text{older } z y \})}{\text{circ } x \wedge \text{with } y x} \right)^{\downarrow \uparrow} \\
 & \left(\frac{\lambda gh: |G_v| = 1. \cup \left\{ [] g^{u \mapsto x} h^{1_u} \mid \text{sq } y, \text{circ } x, \text{with } y x, \forall z: \text{sq}. \neg \text{older } z y \right\}}{x} \right)^{\uparrow} \\
 & \frac{\lambda gh: |G'_u| = |G_v| = 1. \cup \left\{ [] g^{u \mapsto x} h \mid \text{sq } y, \text{circ } x, \text{with } y x, \forall z: \text{sq}. \neg \text{older } z y \right\}}{x}
 \end{aligned}$$

RELATIVE DP-INTERNAL SUPERLATIVES

[[the circus with the oldest squirrel]]

$$\begin{aligned}
 & \left(\frac{\mathbf{the}_u(\lambda x. [])}{x} \parallel \left(\frac{[]}{\mathbf{circus}} \parallel \frac{[]}{\mathbf{with}} \parallel \left(\frac{\mathbf{the}_v \mathbf{M}_v^{\text{ag}}(\lambda y. [])}{y} \parallel \frac{[]}{\mathbf{squirrel}} \right)^{\downarrow \uparrow} \right)^{\downarrow \uparrow} \right)^{\downarrow \uparrow} \\
 & \left(\frac{\mathbf{the}_u(\lambda x. [])}{x} \parallel \left(\frac{[]}{\mathbf{circus}} \parallel \frac{[]}{\mathbf{with}} \parallel \frac{\lambda gh. \cup \{ [] g^{v \mapsto y} h^{1_v \circ \mathbf{M}_v^{\text{ag}}} \mid \text{sq } y \}}{y} \right)^{\downarrow \uparrow} \right)^{\downarrow \uparrow} \\
 & \left(\frac{\mathbf{the}_u(\lambda xgh. \cup \{ [] g^{v \mapsto y} h^{1_v \circ \mathbf{M}_v^{\text{ag}}} \mid \text{sq } y \})}{\text{circ } x \wedge \text{with } y x} \right)^{\downarrow \uparrow} \\
 & \left(\frac{\lambda gh. \cup \left\{ [] g^{u \mapsto x} (h^{1_u})^{1_v \circ \mathbf{M}_v} \mid \text{sq } y, \text{circ } x, \text{with } y x \right\}}{x} \right)^{\uparrow} \\
 & \frac{\lambda gh: |G_u| = |G_v| = 1. \cup \left\{ [] g^{u \mapsto x} \mid \text{sq } y, \text{circ } x, \text{with } y x \right\}}{x}
 \end{aligned}$$

- When the inner DP is reset, it has the effect just demonstrated: the set of outputs is restricted to those that map v to the (unique) oldest squirrel.
- The outer definite has no explicit maximization operator. For simplicity, for now, I assume that the \mathbf{M} argument to **the** is optional. But even max-less, the definite still imposes a uniqueness condition on the values assigned to u .
- Altogether, after the host DP has been reset, we will have an essentially nondeterministic (continuized) individual with two presuppositions: (1) there is exactly one squirrel with no elders; and (2) there is exactly one circus that he is with.

- The only difference here is that we do not reset the inner DP, which staves off its presupposition until more information is accumulated in its scope
- But now when the outer DP is reset, it sets the presuppositions of *both* definites
- For any input g , $G = \left\{ \left\langle x, g^{u \mapsto x} \right\rangle \mid \text{sq } x, \text{circ } y, \text{in } y x \right\}$ is the set of outputs that map u onto a circle in some square that it maps to v .
- So requiring that there be exactly one such v is tantamount to requiring that there be exactly one square *that has a circle in it* and exactly one circle *in that square*. In other words, there should be exactly one pair $\langle x, y \rangle$ in $\text{circ} \times \text{sq}$ such that in $y x$.

5 Focus

$$\begin{aligned}\mathbf{M}_u^f &= \lambda G. \{ \langle \alpha, g, h \rangle \mid \langle \langle \alpha, \cdot \rangle, g, h \rangle \in G, \neg \exists \langle \langle \cdot, \beta \rangle, g', \cdot \rangle \in G. \forall \beta \wedge f(gu)(g'u) \} \\ \mathbf{oldest}_u &= \mathbf{est}_u \mathbf{older} \\ &= \lambda G. \left\{ \langle \mathbf{T}, g, h \rangle \mid \begin{array}{l} \langle \langle \mathbf{T}, \cdot \rangle, g, h \rangle \in G, \\ \neg \exists \langle \langle \cdot, \beta \rangle, g', \cdot \rangle \in G. \forall \beta \wedge \text{age}(gu) < \text{age}(g'u) \end{array} \right\}\end{aligned}$$

- Note that **the** is exactly as before. It is neutral between focus-sensitive and focus-sensitive “maximizers”.

- Superlatives are apparently focus-sensitive. ‘John sold MARY the most expensive painting’ means he sold Mary a more expensive painting than he sold anyone else.
- To prepare for this, the entry for **est** here expects to see outputs with focus-pairs. It filters out the assignments that either fail to satisfy the “ordinary” component of their prejacent, or fail to assign u to a value whose f -degree is greater than that of every other acceptable assignment’s choice of u .

DERIVATIONS

$$\frac{\frac{j^F \star (\lambda j. [])}{j^D \star (\lambda x. [])}}{x} \parallel \frac{\frac{[]}{\text{drew}}}{y} \parallel \frac{\lambda g. \mathbf{M}_u \cup \{ [] g^{u \mapsto y} \mid \text{sq } y \}}{y}$$

John drew the largest square

$$\frac{\frac{A \mid A}{S \mid S}}{\lambda x. x} \quad \frac{\mathbf{the}_v (\lambda y. [])}{y}$$

$$\begin{aligned}\mathcal{F}\alpha &:= \sigma \rightarrow \{\alpha * \sigma\} * \{\alpha * \sigma\} \\ \eta x &:= \lambda g. \langle \{ \langle x, g \rangle \}, \{ \langle x, g \rangle \} \rangle \\ m \star f &:= \lambda g. \left\langle \bigcup \{ (f \ x \ g')_1 \mid \langle x, g' \rangle \in (mg)_1 \}, \bigcup \{ (f \ x \ g')_2 \mid \langle x, g' \rangle \in (mg)_2 \} \right\rangle\end{aligned}$$

- Rolling focus and state simultaneously to grease the wheels

6 Adjectival Exclusives

$$\begin{aligned}\mathbf{true} &:= \lambda G. \bigvee \{ \alpha \mid \langle \alpha, g \rangle \in G \} \\ \mathbf{not} &:= \lambda mg. \eta (\neg(mg)) g \\ \mathbf{only}_u &:= \lambda G: \mathbf{true} \ G. \mathbf{M}_u \ G \\ \mathbf{the}_u &:= \lambda Mckg. |G_u| = 1. G, \\ &\quad \text{where } G = \mathbf{M}_u \bigcup \{ k \ x \ g' \mid \langle \mathbf{T}, g' \rangle \in c \ x \ g^{u \mapsto x} \}\end{aligned}$$

- true** and **not** are bog standard dynamic booleans: an update is “true” iff it generates at least one successful output
- Adjectival **only** is just **M** plus a presupposition (cf. Coppock and Beaver 2012, 2014)!
- So here’s the swim move: feed **only** into **the**, and then let the scope of the exclusivity ride the scope of the determiner. This will predict relative readings for adj **only**.

John sold the only cars

Anna didn’t give the only good talk