Chapter 17

$\star$-CCG

$\star$-CCG represents an extension of CCG achieved through the integration of a continuation monad Wadler (1992); Shan (2001). This framework distinguishes itself by orchestrating semantic composition such that the predicate-arguments structures are calculated at the "ground-floor" level, while the interactions between quantifiers is meticulously mediated by continuation. This architectural choice aligns with the ideas of continuized grammar, wherein quantification is treated as a *side-effect* of the broader semantic compositional process.

17.1 Continuation Monad

As a preparatory step in formal definition of $\star$-CCG, we introduce a specific set of DTS functions structured as a *continuation monad.* This monadic construction is foundational to the subsequent development. Firstly, a type shifter $\boldsymbol{M}$ is defined as a endofunctor mapping terms within UDTT to UDTT terms. Specifically, for any UDTT type A, this functor yields its relative doulbe negation.

Definition 366. $\boldsymbol{M}A \overset{def}{\equiv} (A \to \mathsf{type}) \to \mathsf{type}$

The type $\boldsymbol{M}A$ denotes a type A augmented with a *continuation* of type $A \to \mathsf{type}$ where type serves as the fixed answer type. While its computational underpinnings will be elaborated in Section **??**, its linguistic intuition is that $\boldsymbol{M}A$ serves as the type for an A-object endowed with quantification. Consider, for instance, a proper name such as *John*, assigned an LF of type e. In stark contrast, a quantificational expression like *everyone* receives an LF of type $\boldsymbol{M}(\mathsf{e})$, which unpacks to $(\mathsf{e} \to \mathsf{type}) \to \mathsf{type}$. Notably, this semantic type is standardly ascribed to a type-raised noun phrase ($NP^{\uparrow}$) within conventional semantic theories, given that type effectively plays the role of type t (the type of truth values).

Secondly, we introduce four UDTT functions to facilitate the manipulate of the continuation monad.

Definition 367 (Monadic operators in UDTT)**.**

$$\eta \overset{def}{\equiv} \lambda x.\lambda\kappa.\kappa x$$
$$\mu \overset{def}{\equiv} \lambda f.\lambda\kappa.f(\lambda g.g(\kappa))$$
$$f^{\dagger} \overset{def}{\equiv} \lambda a.\lambda\kappa.f(\lambda m.a(\lambda n.\kappa(m(n))))$$
$$f^{\star} \overset{def}{\equiv} (\eta f)^{\dagger}$$

$$\frac{NP}{T/(T\backslash NP)}{>}T$$

Each function is assigned the following type.

john
entity

η(john)
(entity $\to$ type) $\to$ type

$\boxed{\boxed{\boxed{\text{john}}}}$

Theorem 368.

$$\begin{array}{rcl} & \vdash & \eta : A \to \boldsymbol{M}A \\ & \vdash & \mu : \boldsymbol{M}\boldsymbol{M}A \to \boldsymbol{M}A \\ f : \boldsymbol{M}(A \to B) & \vdash & f^{\dagger} : \boldsymbol{M}A \to \boldsymbol{M}B \\ f : A \to B & \vdash & f^{\star} : \boldsymbol{M}A \to \boldsymbol{M}B \end{array}$$

The functoin η, termed the *unit*, serves to embed any preterm of type A into $\boldsymbol{M}A$. Given that $\boldsymbol{M}$ is conceived as a relativised double negation of A, η can be interpreted as the proof term for double negation introduction, an intuitionistic theorem.

Conversely, μ, designatedas the *join* operatoin, functions to eliminate a single instance of the monadic operator $\boldsymbol{M}$ from the double application $\boldsymbol{M}\boldsymbol{M}A$. Consequently, μ is to be understood as as a doubly-negated version of double negation elimination, which similarly holds as an intuitionistic theorem.

Furthermore, we introduce the operation $(-)^{\dagger}$, which maps a monadic function f of type $\boldsymbol{M}(A \to B)$ to its corresponding lifted function $f^{\dagger}$, from $\boldsymbol{M}A$ to $\boldsymbol{M}B$.

Finally, $(-)^{\star}$, referred as *lift*, transforms a plain function f from A to B into its lifted counterpart $f^{\star}$, mapping from $\boldsymbol{M}A$ to $\boldsymbol{M}B$. This transformation is defined through the combination of both η and $\dagger$. It is pertinent to note the intentional notational overloading of $\star$; its distinct uses in syntactic types within $\star$-CCG and in semantic representations within DTS should obviate any ambiguity.

The formal proofs of Theorem 368 are detailed in Subsection **??**. Theorem 368 establishes the requisite theoretical groundwork for the subsequent theorems.

Theorem 369. $\langle \eta; \star; \mu \rangle$ is a monad, namely, the following monad laws hold.

$$\mu \circ \mu^{\star} =_{\beta} \mu \circ \mu$$
$$\mu \circ \eta^{\star} =_{\beta} \mu \circ \eta \ =_{\beta}\ id$$
$$\eta \circ f =_{\beta} f^{\star} \circ \eta$$
$$\mu \circ f^{\star\star} =_{\beta} f^{\star} \circ \mu$$

The proof of Theorem 369, alongside its crucial application to normal form parsing, will be presented in Subsection **??**.

17.2 ⋆-CCG

We now formally define ⋆-CCG. The collection of base syntactic types for ⋆-CCG are directly inherited from those established in CCG.

Definition 370. A collection of *syntactic types* of ⋆-CCG is recursively defined as follows:[*1]

1. Base ~~categories~~ syntactic types are syntactic types.
2. If X and Y are syntactic types, then both X/Y and $X\backslash Y$ are syntactic types.
3. If X is a syntactic type, then $\boxed{X^\star}$ is a syntactic type.

The novel ⋆-operator, which transforms a syntactic type X into $X^\star$, repreesnts the *continuation monad* (cf. Shan (2001))[*2].

Definition 371. ~~Syntax-semantics~~ Type correspondence map from syntactic types of ⋆-CCG to preterms of UDTT is recursively defined as follows:

$$\begin{aligned}\boxed{\lceil X^\star \rceil} &\overset{def}{\equiv} M\lceil X \rceil \\ &\equiv \boxed{(\lceil X \rceil \to \mathsf{type})} \to \mathsf{type}\end{aligned}$$

$$\dfrac{\dfrac{\text{John}}{NP}}{NP^\star}\uparrow \qquad \dfrac{\dfrac{\text{John}}{\text{john}}}{\lambda K.\,K(\text{john})}\uparrow$$

$\uparrow$ continuation.

Given that the sort **type** denotes the type of the semantic representation of a sentence – that is, the semantic type corresponding to the syntactic type S – the present definition carries the following implication: if $X \equiv NP$, then $X^\star$ functions as a syntactic type for quantifiers within the framework standard formal semantics. This equivalence mirrors the continuation-passing style (CPS) transformation of X in functional programming and, moreover, corresponds to a relative double negation of X in logic.

To accommodate the ⋆ operator, the definition of $\boldsymbol{T}$-reducible syntactic types, as presented in Definition 324, necessitates the following extention.

Definition 372. For any syntactic type X, $X^\star$ is a S-reducible syntactic type.

Consequently, any syntactic type of the form $X^\star$ may be subject to conjunction by means of $\langle\Phi\rangle_{X^\star}$ rule. As a direct illustration, consider the syntactic type $S^\star$. Its

[*1] *The categorial assumption*, as articulated in Steedman (2024), posits that in CCG, "[a]ll linguistic categories are defined syntactically and semantically as FUNCTIONS and/or ARGUMENTS." In ⋆-CCG, this fundamental principle necessitates a minor refinement; the last clause must be amended to "FUNCTIONS, ARGUMENTS, and/or their MONADIC extentions" in accordance with Definition 370.

[*2] Monads, as presented by (Wadler, 1992) within type theory, serves, inter alia, to encapsulate and manage *side-effects* arising from computation. A salient application of continuations in the context of categorical grammars involves reinterpriting covert movement operations, such as quantifier raising (May, 1977) as side effects of semantic composition. This elegant reconception obviates the intricate problem of *type mismatch* that quantifiers encounter (Heim and Kratzer, 1998). This particular approach has been adopted and further developed by Barker and Shan (2014).

conjunctive application yields a semantic interpretation derivable from the established definition, as follows.

Definition 373. $(\!|\langle\Phi_{S^\star}\rangle|\!) \overset{def}{\equiv} \lambda f.\lambda c.\lambda g.\lambda\kappa.((c(f\kappa))g(\kappa))$

We proceed to define the combinatory rules of ⋆-CCG. This is achieved by augmenting the standard set of CCG combinatory rule with three distinct unary rules (**Unit**, **Join**, and **Reset**), and the introduction of a recursive rule, designated **Lift**.[*3]

Definition 374 (Combinatory Rules of ⋆-CCG)**.** Let X, Y be syntactic types. Combinatory rules of ⋆-CCG is defined as follows:

1. All the CCG combinatory rules are ⋆-CCG combinatory rules.
2. The following rules are ⋆-CCG combinatory rules, where R is a combinatory rule of ⋆-CCG except $\langle\Phi\rangle$.[*4]

Unit	**Join**	**Reset**	**Lift**
$\dfrac{X}{X^\star}\uparrow$	$\dfrac{X^{\star\star}}{X^\star}\downarrow$	$\dfrac{S^\star}{S}\checkmark$	$\dfrac{X_1^\star \quad \cdots \quad X_n^\star}{Y^\star}{\scriptstyle R^\star} \quad \text{if} \quad \dfrac{X_1 \quad \cdots \quad X_n}{Y}{\scriptstyle R}$

The semantic import of these combinatory rules is elucidated by their respective semantic interpretations.

Definition 375. Semantic interpretation of combinatory rules are recursively defined as follows.[*5]

$$
\begin{aligned}
(\!|\uparrow|\!) &\overset{def}{\equiv} \eta \\
(\!|\downarrow|\!) &\overset{def}{\equiv} \mu \\
(\!|\checkmark|\!) &\overset{def}{\equiv} \lambda f.f(id) \\
(\!|R^\star|\!) &\overset{def}{\equiv} \lambda f_1.\cdots\lambda f_n.(\cdots((\!|R|\!)^\star f_1)^\dagger\cdots)^\dagger f_n
\end{aligned}
$$

(ToDo: Explain the intuition behind the above definition.)

[*3] The **Unit**, **Reset**, and **Lift** rules find their precise analogues within continuation-based grammar (Barker and Shan, 2014): these correspond to the *lift*, *lower*, and the *lifted* version of R, respectively. The **Join** rule posesses no direct counterpart in continuation-based grammar. The **Reset** rule, as its appellation suggests, serves to *reset* the continuation.

[*4] The **Lift** rule, by its exception clause, does not apply to $\langle\Phi\rangle$ (coordination) rule. This exception is empirically motivated: it crucially prevents quantificatoinal expressions embedded within conjuncts from taking inverse scope over the conjunction itself. This observation, in turn, provides compelling grounds for treating coordination as an independent rule.

[*5] We may definie $(\!|R^\star|\!)$ for the general R with an n-arguments as follows (if necessary).

Remark 376. We present here a generalized definition of $(\!|R^\star|\!)$ for an arbitrary rule R of arity n.

$$\dfrac{\dfrac{\eta : (A \to B) \to \boldsymbol{M}(A \to B) \quad (\!|R|\!) : A \to B}{\eta(\!|R|\!) : \boldsymbol{M}(A \to B)}\,{\scriptstyle (\Pi E)}}{(\eta(\!|R|\!))^\dagger : \boldsymbol{M}A \to \boldsymbol{M}B}\,{\scriptstyle (\dagger)}$$

In the following discourse, however, our focus will be confined to instances of $(\!|R\star|\!)$ with $n = 1$ and $n = 2$, derived as follows.

$$(\!|R^\star|\!) \overset{def}{\equiv} \begin{cases} \lambda f_1.\lambda f_2.((\!|R|\!)^\star f_1)^\dagger f_2 & \textit{where } R \text{ is a binary rule.} \\ (\!|R|\!)^\star & \textit{where } R \text{ is a unary rule.} \end{cases}$$

The semantic interpretation of $\star$-CCG is formally constructed such that each rule possesses an intrinsic semantic interpretation, as a higher-order function to which the interpretations of the premises are passed. This architechture is necessitated by the requirement that the interpretation of the **Lift** $(R^\star)$ rule must inherently refer to the interpretation of (R). Notwithstanding this, for the CCG fragment, the combinatorial rules and their associated semantic interpretations are equivalent to the original CCG formulation.

17.3 Box Notation

The *box* notation is herein introduced as a novel notational convention, superseding the tower notation within the framework of continuation-based grammar. Its formal definition is presented as follows.

Definition 377 (Box Notation[*6]).

$$f\boxed{a} \overset{def}{\equiv} \lambda\kappa.f(\kappa(a))$$

Thus, $f\boxed{a}$ corresponds to the following tower.

$$\frac{f[\;]}{a}$$

The tower notation, whilst serviceable in certain contexts, suffers from a lack of robustness. Specifically, it proves inadequate for representing all conceivable semantic forms that emerge in a continuation-based grammar. This limitation becomes salient when semantic representations involve multiple occurrences of the continuation variable κ. Consider, for instance, the semantic representation of (166a), where conjunction induces a copy of κ within a coordinated structure of two quantifiers. This is expressible in flat notation as (166b).

(166) a. every boy and some girl
b. $\lambda\kappa.(\forall x(\mathbf{boy}(x) \to \kappa x)) \wedge (\exists y(\mathbf{girl}(y) \to \kappa(y)))$

However, (166b) finds no corresponding representation in tower notation. Although one might attempt to represent it as (167), the constraint of tower notation – that only a singular object (here, x or y) may occupy each floor – precludes such a representation.

(167) $$\frac{(\forall x(\mathbf{boy}(x) \to [\,])) \wedge (\exists y(\mathbf{girl}(y) \to [\,]))}{x \qquad y}$$

The ⋆-CCG's box notation remedies this deficiency. Utilising this convention, Example (166a) is perspicuously rendered as (168).

(168) $\forall x(\mathbf{boy}(x) \to \boxed{x}) \wedge (\exists x(\mathbf{girl}(x) \to \boxed{x}))$

The hierarchical nesting inherent in the box notation mirrors a tree structure, reflecting the semantic representation where continuations can indeed be duplicated. For grammars that explicitly mediate continuation in semantic composition, such as continuation-based grammar and ⋆-CCG, this novel notation offers a robust and comprehensive means of capturing every relevant structural detail.

Definition 378. A semantic representation of the structure $f\,\boxed{m}$ is designated a *level1-box*, wherein f resides at the first level and m at the ground level. Extending this convention, a semantic representation of the form $f\,\boxed{g\,\boxed{m}}$ constitutes a *level2-box*, with f at the second level, g at the first, and m on the ground.

$\lambda k.\,k(john)$
$\shortparallel$
$\boxed{john}$

Employing this box notation, the three unary rules established within ⋆-CCG, complete with their semantic interpretations, are presented below.

$\cancel{\lambda k}.f(\cancel{k}(m))$

$$\frac{X : m}{X^\star : \boxed{m}}\uparrow \qquad \frac{X^{\star\star} : f\,\boxed{g\,\boxed{m}}}{X^\star : ~~f\,\boxed{g(m)}~~}\downarrow \qquad \frac{S^\star : f\,\boxed{m}}{S : f(m)}\checkmark$$

$f(g\boxed{x})$

id

Theorem 379. For any unary ⋆-CCG rule R, the following equation holds.

$$(\!|R^\star|\!)(f\,\boxed{m}) = f(\,\boxed{(\!|R|\!)m}\,)$$

Proof.

$$\begin{aligned}
\eta(\!|R|\!) &=_\beta \lambda\kappa.\kappa((\!|R|\!)) \\
(\eta(\!|R|\!))^\dagger &=_\beta (\lambda f.\lambda a.\lambda k.f(\lambda m.(a(\lambda n.(k(m(n)))))))(\lambda\kappa.\kappa((\!|R|\!))) \\
&=_\beta \lambda a.\lambda k.(\lambda\kappa.\kappa((\!|R|\!)))(\lambda m.(a(\lambda n.(k(mn))))) \\
&=_\beta \lambda a.\lambda k.(a(\lambda n.(k((\!|R|\!)n)))) \\
(\eta(\!|R|\!))^\dagger(f\,\boxed{m}) &=_\beta (\lambda a.\lambda k.(a(\lambda n.(k((\!|R|\!)n)))))(\lambda\kappa.f(\kappa(m))) \\
&=_\beta \lambda k.((\lambda\kappa.f(\kappa(m)))(\lambda n.(k((\!|R|\!)n)))) \\
&=_\beta \lambda k.f(k((\!|R|\!)m))
\end{aligned}$$

□

Theorem 380. For any binary ⋆-CCG rule R, the following equation holds.

$$(\!|R^\star|\!)(f\,\boxed{m})(g\,\boxed{n}) = f(g(\,\boxed{((\!|R|\!)m)n}\,))$$

Proof.

$$\begin{aligned}
((\eta\llparenthesis R\rrparenthesis)^\dagger(f\boxed{m}))^\dagger &=_\beta (\lambda f.\lambda a.\lambda k.f(\lambda m.(a(\lambda n.(k(m(n)))))))(\lambda k.f(k(\llparenthesis R\rrparenthesis m)))\\
&=_\beta \lambda a.\lambda k.(\lambda k.f(k(\llparenthesis R\rrparenthesis m)))(\lambda m.(a(\lambda n.(k(m(n))))))\\
&=_\beta \lambda a.\lambda k.f(a(\lambda n.(k((\llparenthesis R\rrparenthesis m)n))))
\end{aligned}$$

$$\begin{aligned}
\llparenthesis R^\star\rrparenthesis(f\boxed{m})(g\boxed{n}) &=_\beta ((\eta\llparenthesis R\rrparenthesis)^\dagger(f\boxed{m}))^\dagger(g\boxed{n})\\
&=_\beta (\lambda a.\lambda k.f(a(\lambda n.(k((\llparenthesis R\rrparenthesis m)n)))))(\lambda\kappa.g(\kappa n))\\
&=_\beta \lambda k.f((\lambda\kappa.g(\kappa n))(\lambda n.(k((\llparenthesis R\rrparenthesis m)n))))\\
&=_\beta \lambda k.f(g(k((\llparenthesis R\rrparenthesis m)n)))
\end{aligned}$$

□

Corollary 381. The following rules are derivable in ⋆-CCG.

$$\frac{(X/Y)^\star : f\boxed{m} \quad Y^\star : g\boxed{n}}{X^\star : f(g\boxed{m(n)})}\,{<}^\star \qquad \frac{Y^\star : f\boxed{m} \quad (X\backslash Y)^\star : g\boxed{n}}{X^\star : f(g\boxed{n(m)})}\,{<}^\star$$

$$\frac{X/Y : f\boxed{m} \quad Y/Z : g\boxed{n}}{X/Z^\star : f(g\boxed{\lambda x.m(n(x))})}\,{>}B^\star \qquad \frac{Y\backslash Z : f\boxed{m} \quad X\backslash Y : g\boxed{n}}{X\backslash Z : f(g\boxed{\lambda x.n(m(x))})}\,{<}B^\star$$

$$\frac{X/Y : f\boxed{m} \quad Y\backslash Z : g\boxed{n}}{X\backslash Z : f(g\boxed{\lambda z.m(n(z))})}\,{>}B^\star_\times \qquad \frac{Y/Z : f\boxed{m} \quad X\backslash Y : g\boxed{n}}{X/Z : f(g\boxed{\lambda z.n(m(z))})}\,{<}B^\star_\times$$

$$\frac{X/Y : f\boxed{m} \quad Y/Z\,|\,W : g\boxed{n}}{X/Z\,|\,W : f(g\boxed{\lambda w.\lambda z.m(n(w)(x))})}\,{>}B^{2\star} \qquad \frac{Y\backslash Z\,|\,W : f\boxed{m} \quad X\backslash Y : g\boxed{n}}{X\backslash Z\,|\,W : f(g\boxed{\lambda w.\lambda z.n(m(w)(x))})}\,{<}B^{2\star}$$

$$\frac{X : f\boxed{x}}{Y/(Y\backslash X) : f\boxed{\lambda p.px}}\,{>}T^\star \qquad \frac{X : f\boxed{x}}{Y\backslash(Y/X) : f\boxed{\lambda p.px}}\,{<}T^\star$$

$$\frac{X/Y\backslash Z : f\boxed{m} \quad Y\backslash Z : g\boxed{n}}{X\backslash Z : f(g\boxed{\lambda z.(m(z))(n(z))})}\,{>}S_\times{}^\star \qquad \frac{Y/Z : f\boxed{m} \quad X\backslash Y/Z : g\boxed{n}}{X/Z : f(g\boxed{\lambda z.(n(z))(m(z))})}\,{<}S_\times{}^\star$$

$$\frac{X : f\boxed{m}}{X^\star : f\boxed{\boxed{m}}}\,\uparrow^\star \qquad \frac{X^{\star\star} : f\boxed{g\boxed{m}}}{X^\star : f\boxed{g(m)}}\,\downarrow^\star$$

The semantic rules for other lifted rules are derived in the same way.

Theorem 382. For any unary ⋆-CCG rule R, the following equation holds.

$$[\![R^{\star\star}]\!]\ (f\,\boxed{h\,\boxed{m}}) = f\,\boxed{h\,\boxed{(\!|R|\!)m}}$$

And for any binary ⋆-CCG rule R, the following equation holds.

$$[\![R^{\star\star}]\!]\ (f\,\boxed{h\,\boxed{m}})(g\,\boxed{i\,\boxed{n}}) = f(g(\boxed{h(i(\boxed{((\!|R|\!)m)n}}))))$$

Proof. Immediate from Theorem 379 and Theorem 380. □

17.4 Benchmarks

We now proceed to illustrate how ⋆-CCG reproduces the basic analyses characteristic of continuation-based grammars.

17.4.1 Canonical scope readings

Our initial benchmark involves demonstrating the derivation of both canonical scope ($\exists > \forall$) and inverse scope ($\forall > \exists$) for (169).

(169) Someone loves everyone

$S/(S\backslash NP)$

The crucial lexical items for *someone* and *everyone* within the ⋆-CCG framework are specified as follows.

(170) a. $[\![\text{someone} \vdash NP^\star]\!] \overset{def}{\equiv} \left(u : \begin{bmatrix} x : \mathsf{e} \\ \textbf{human}(x) \end{bmatrix}\right) \times \boxed{\pi_1 u}$

b. $[\![\text{everyone} \vdash NP^\star]\!] \overset{def}{\equiv} \left(v : \begin{bmatrix} y : \mathsf{e} \\ \textbf{human}(y) \end{bmatrix}\right) \to \boxed{\pi_1 v}$

These lexical items represent level-1 structures (signified by a single internal box), wherein quantification consistently occupies the first level, with the argument situated at the ground level.

The derivation of the canonical scope is straightforward. As depicted in the derivation (171), the syntactic structure of the original CCG merely undoergoes the additoin of the ⋆ operator to each syntactic type.

(171) Syntactic structure of (169) in ⋆-CCG: Canonical scope

$$\cfrac{\cfrac{\dfrac{\text{someone}}{NP^\star} \qquad \cfrac{\cfrac{\dfrac{\text{loves}}{S\backslash NP/NP}}{(S\backslash NP/NP)^\star}\uparrow \qquad \dfrac{\text{everyone}}{NP^\star}}{(S\backslash NP)^\star}>^\star}{S^\star}<^\star}{S}\ \checkmark$$

$$\dfrac{X/Y \qquad Y}{X}>$$

$$\Downarrow$$

$$\dfrac{(X/Y)^\star \qquad Y^\star}{X^\star}>^\star$$

(172) Semantic composition of (169) in ⋆-CCG: Canonical scope

$$\dfrac{\dfrac{\dfrac{\text{someone}}{\begin{bmatrix} x:\mathrm{e} \\ \boxed{x}\end{bmatrix}} \qquad \dfrac{\dfrac{\dfrac{\text{loves}}{\lambda y.\lambda x.\mathbf{love}(x,y)}\uparrow}{\boxed{\lambda y.\lambda x.\mathbf{love}(x,y)}} \qquad \dfrac{\text{everyone}}{(y:\mathrm{e})\to\boxed{y}}}{(y:\mathrm{e})\to\boxed{\lambda x.\mathbf{love}(x,y)}}>^{\star}}{\begin{bmatrix} x:\mathrm{e} \\ (y:\mathrm{e})\to\boxed{\mathbf{love}(x,y)}\end{bmatrix}}<}{\begin{bmatrix} x:\mathrm{e} \\ (y:\mathrm{e})\to\mathbf{love}(x,y)\end{bmatrix}}\checkmark$$

id

Semantic composition, within DTS, operates at each level of the box. Specifically, quantification is composed in left to right order, while the argument structure, consistent with the original CCG, is computed internally within the box. The resultant DTS semantic representation invariably exhibits a scopal ordering of $\Sigma > \Pi$, thereby accurately reflecting the intended interpretation.

A more detailed examination of this compositional process, with specific to (169), is undertaken as follows.

(173)
$$\begin{aligned}
& (\!|>^{\star}|\!)\ [\![\texttt{loves}]\!]\ [\![\texttt{everyone}]\!] \\
=_{\beta}\ & (\!|>^{\star}|\!)(id\,\boxed{\mathbf{love}})((y:\mathrm{e})\to\boxed{y}) \\
=_{\beta}\ & id((y:\mathrm{e})\to\boxed{([\![>]\!]\ \mathbf{love})y}) \\
=_{\beta}\ & (y:\mathrm{e})\to\boxed{((\lambda f.\lambda x.fx)\mathbf{love})y} \\
=_{\beta}\ & (y:\mathrm{e})\to\boxed{(\mathbf{love}y)}
\end{aligned}$$

(174)
$$\begin{aligned}
& (\!|<^{\star}|\!)\ [\![\texttt{someone}]\!]\ ((\!|>^{\star}|\!)\ [\![\texttt{loves}]\!]\ [\![\texttt{someone}]\!]) \\
=_{\beta}\ & (\!|<^{\star}|\!)((x:\mathrm{e})\times\boxed{x})((y:\mathrm{e})\to\boxed{(\mathbf{love}y)}) \\
=_{\beta}\ & (x:\mathrm{e})\times(y:\mathrm{e})\to\boxed{([\![<]\!]\ (x)(\mathbf{love}y))}
\end{aligned}$$

17.4.2 Inverse scope readings

To "lifet" a quantifier such as *everyone* from the level-1 structure to the level-2 structure, two distinct operations are available: the unit rule ($\uparrow$) and the lifted unit rule ($\uparrow^{\star}$).

The unit rule ($\uparrow$) superimposes a vacuous second level upon the entire existing semantic representation.

(175)
$$\dfrac{\left(v:\begin{bmatrix} y:\mathrm{e} \\ \mathbf{human}(y)\end{bmatrix}\right)\to\boxed{\pi_1 v}}{\boxed{\left(v:\begin{bmatrix} y:\mathrm{e} \\ \mathbf{human}(y)\end{bmatrix}\right)\to\boxed{\pi_1 v}}}\uparrow$$

This second level is populated by the identity function, ensurig that the original first and ground levels remain unchanged. In contrast, the lifted unit rule ($\uparrow^{\star}$) effects a more complex structural rearrangement. It elevates the original ground level to a new first level, thereby displacing the erstwhile first level to the second. Consequently, the resulting structure features Π-type at the second level, an empty (or identity) first level, and the original argument $\pi_1 u$ occupying the ground level.

Within the framework of ⋆-CCG, the derivation of inverse scope mirrors the principles established in continuation-based grammar. This is achieved by applying the lifted unit ($\uparrow^\star$) rule to the object, thereby elevating it to a level-2 structure.

(176)
$$\dfrac{\left(v : \begin{bmatrix} y : \mathsf{e} \\ \mathbf{human}(y) \end{bmatrix}\right) \to \boxed{\pi_1 v}}{\left(v : \begin{bmatrix} y : \mathsf{e} \\ \mathbf{human}(y) \end{bmatrix}\right) \to \boxed{\boxed{\pi_1 v}}}\,\uparrow^\star$$

Concomitantly, the remaining constituents, specifically the subject and verb in this instance, are commensurately upgraded to level-2 through the application of the standard Unit ($\uparrow$) rule.[*7]

The mechanism for deriving inverse scope hinges upon adopting an analysis akin to continuation-based grammar. This involves the application of a lifted unit rule ($\uparrow^\star$) to the universal quantifier, *everyone* – an operation analogous to the "lifted version of lift" in continuation-based grammar, taking place "inside the box."

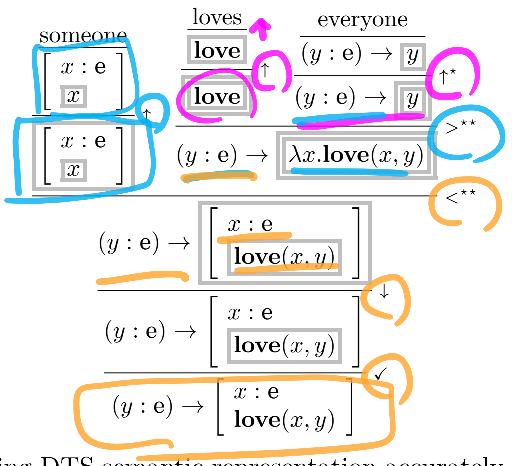
(177) Syntactic structure of (169) in ⋆-CCG: Inverse scope

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{someone}}{NP^\star}}{NP^{\star\star}}\uparrow \quad \dfrac{\dfrac{\dfrac{\text{loves}}{(S\backslash NP/NP)^\star}\,\uparrow}{(S\backslash NP/NP)^{\star\star}}\uparrow \quad \dfrac{\dfrac{\text{everyone}}{NP^\star}}{NP^{\star\star}}\uparrow^\star}{(S\backslash NP)^{\star\star}}>^{\star\star}}{S^{\star\star}}<^{\star\star}}{S^\star}\downarrow}{S}\,\checkmark$$

Consequently, the universal quantification associated with *everyone* is situated at the level-2 continuation. This necessitates the concomitant lifting of both *someone* and *loves* to level-2 continuation, achieved through the application of the standard unit rule ($\uparrow$) (not its lifted variant ($\uparrow^\star$)). As a direct consequence, the existential quantification corresponding to *someone* remains embedded in the first level.

[*7] It is noteworthy that, across various scenarios – including the application of $\uparrow^\star$ to both subject and object, $\uparrow$ to both, or $\uparrow^\star$ to the subject alongside $\uparrow$ for the rest – the resultant representation consistently aligns with that of the canonical scope.

(178) Semantic composition of (169) in ⋆-CCG: Inverse scope



The resulting DTS semantic representation accurately reflects the intended inverse scope reading $\Pi > \Sigma$.

(179)
$$
\begin{aligned}
& [\![>^{\star\star}]\!]\,([\![\uparrow]\!]\,[\![\texttt{loves}]\!])((\!|\uparrow^\star|\!)\,[\![\texttt{everyone}]\!]) \\
=_\beta\ & (\!|>^{\star\star}|\!)(\boxed{\boxed{\mathbf{love}}})((y:\mathsf{e}) \to \boxed{\boxed{y}}) \\
=_\beta\ & id((x:\mathsf{e}) \to \boxed{(\!|>^{\star}|\!)\,\boxed{\mathbf{love}}\,\boxed{y}}) \\
=_\beta\ & (x:\mathsf{e}) \to \boxed{id(id\,\boxed{((\lambda f.\lambda x.fx)\ \mathbf{love})\ y})} \\
=_\beta\ & (x:\mathsf{e}) \to \boxed{\boxed{\mathbf{love}\ y}}
\end{aligned}
$$

(180)
$$
\begin{aligned}
& [\![<^{\star\star}]\!]\,([\![\uparrow]\!]\,[\![\texttt{someone}]\!])([\![>^{\star\star}]\!]\,([\![\uparrow]\!]\,[\![\texttt{loves}]\!])([\![\uparrow^\star]\!]\,[\![\texttt{everyone}]\!])) \\
=_\beta\ & [\![<^{\star\star}]\!]\,\boxed{(x:\mathsf{e}) \times \boxed{x}}\,((x:\mathsf{e}) \to \boxed{\boxed{\mathbf{love}\ y}}) \\
=_\beta\ & id((x:\mathsf{e}) \to \boxed{[\![<^{\star}]\!]\,\boxed{x}\,\boxed{\mathbf{love}\ y}}) \\
=_\beta\ & (x:\mathsf{e}) \to \boxed{id(id\,\boxed{([\![<]\!]\ x)\ (\mathbf{love}\ y)})} \\
=_\beta\ & (x:\mathsf{e}) \to \boxed{\boxed{((\lambda x.\lambda f.fx)\ x)\ (\mathbf{love}\ y)}} \\
=_\beta\ & (x:\mathsf{e}) \to \boxed{\boxed{\mathbf{love}(x,y)}}
\end{aligned}
$$

History and Further Readings

- Early work on monads for natural language semantics: Shan (2001)
- Early work on continuation for natural language semantics: Barker (2001, 2002, 2004), Bekki and Asai (2010), Cong (2014)
- Development of monadic semantics: Ogata (2008), Bekki and Masuko (2012), Unger (2011), Charlow (2014), Grudzińska and Zawadowski (2017)
- Continuation-based grammar (with tower notations) Barker and Shan (2008, 2014)
- Continuized CCG: White (2019), Matsuoka et al. (2023)