Against successive cyclicity: A proof-theoretic account of extraction pathway marking

Yusuke Kubota
NINJAL

Robert Levine
Ohio State University

December 26, 2023

Abstract

This paper proposes a novel analysis of extraction pathway marking in Type-Logical Grammar, taking advantage of proof-theoretic properties of logical proofs whose empirical application has so far been underexplored. The key idea is to allow certain linguistic expressions to be sensitive to the intermediate status of a syntactic proof. The relevant conditions can be stated concisely as constraints at the level of the proof term language, which is formally a special type of λ-calculus. The proposed analysis does not have any direct analog to either of the two familiar techniques in the syntactic literature for analyzing extraction pathway marking, namely, successive cyclic movement in derivational syntax and the SLASH feature percolation in HPSG. It nonetheless captures the relevant empirical patterns at least equally successfully. This new analysis is conceptually revealing as well: on the 'meaning-centered' perspective that emerges naturally from this approach, extraction pathway marking essentially boils down to a strategy that certain languages employ for overtly flagging the existence of a semantic variable inside a partially derived linguistic expression whose interpretation is dependent on a higher-order operator located in a structurally distant position.

1 Introduction

A widely entertained—but seldom questioned—assumption in generative syntax holds that the long-distance movement operation is 'successively cyclic' (Chomsky 1973, 1977). Of all the arguments that have been adduced to this notion, perhaps the clearest type of evidence comes from the so-called *extraction pathway marking* phenomena exhibited by typologically diverse languages (Kayne and Pollock 1978; McCloskey 1979; Chung 1982; Zaenen 1983; Borsley 2010; van Urk and Richards 2015), in which a syntactically displaced expression induces overtly visible effects at the intermediate landing sites of a chain of movement linking the filler and the gap. Unlike purely theory-internal arguments, such evidence calls for an explanation in any approach to natural language syntax (for a compact and readable summary of this complex and rich literature, see van Urk (2020)).[1] In fact, a non-movement-based analysis has been worked out in detail in HPSG (Bouma et al. 2001), exploiting the feature percolation mechanism native to such nonderivational approaches.

In the present paper, we question this long-held assumption in the syntactic literature, arguing that an account of extraction pathway marking is possible without recognizing either successively cyclic movement or feature percolation. This stands in a stark contrast to a recent claim made by van Urk and Richards (2015) and van Urk (2020) that *both* successively cyclic movement and feature percolation (or feature checking) are needed to capture the entire patterns of extraction pathway marking across different languages. The surprising result that neither of the ostensibly well-motivated and well-understood mechanisms is needed comes from trying to analyze extraction pathway marking in a theory in which neither device is native to the underlying architecture: a proof-theoretic variant of categorial grammar known as Type-Logical Grammar (TLG). In fact, previous literature on long-distance dependencies in categorial grammar (in both CCG and TLG) has remained moot on the question of successive cyclicity, with the central focus being on how to capture island sensitivity in narrow syntax (see, e.g., Steedman (2000), Morrill (1994) and Morrill (2017)).[2] The only exception we are aware of is the recent proposal by Kubota and Levine (2020), which, as the authors themselves admit, is essentially a clumsy rendering of the HPSG-style feature percolation analysis by Bouma et al. (2001) within TLG.

The new analysis we advocate in this paper takes the proof-theoretic perspective on natural language syntax inherent to TLG as its starting point, and our goal is to argue that extraction pathway marking in fact provides novel empirical evidence for such an architecture of grammar. In TLG, proofs represent histories of the combinatoric process in which larger linguistic sings are assembled from its parts. Just like any standard system of logic, proofs are themselves structured objects, but crucially, they don't reflect phrase structural constituency directly (this is the main difference between proof trees in TLG and linguistic trees in mainstream syntax).

The key claim of the present paper is that extraction pathway marking can be best understood as a 'strategy' that the grammar of some languages employ in making the intermediate (or 'incomplete') status of such proofs visible in surface syntax, in a sense to be made precise below. In this approach, nothing 'moves' literally (let alone in

[1] See Pullum (1992) for a insightful and critical survey of the theoretical status of the notion of 'transformational cycle' in the history of generative grammar. An initial motivation for this notion came from constraints on rule ordering. Yet 'bafflingly' (in Pullum's (1992, 212) words), the notion of cycle was retained after the motivating idea of rule ordering was abandoned (a decisive factor in this latter theoretical move came from Chomsky and Lasnik's (1977) proclamation that all transformational rules were unordered), with the alleged justification—worked out in detail in Chomsky (1973)—that it provided a basis for a principled explanation for Ross's (1967) island constraints (for an updated perspective on the status of syntactic islands, see footnote 2).

[2] There is now a growing body of literature challenging this classical consensus about the status of island constraints, offering various alternative explanations based on pragmatic or processing-oriented factors (some important work includes Deane (1992), Kluender (1992, 1998), Hofmeister and Sag (2010) and Chaves and Putnam (2020)). See Newmeyer (2016) for a concise overview, and Kubota and Levine (2020, Chapter 10) for a discussion of this literature on extragrammatical factors on island constraints from a contemporary TLG perspective.

a successive cyclic way), nor is there any structure-manipulation operation or feature percolation of any sort. As noted above, this was precisely the source of the apparent impasse that this phenomenon poses for TLG, but this impasse is in fact an illusion and the reality is to the contrary: the proof theoretic perspective *predicts* the existence of extraction pathway marking in natural language, in the sense that the phenomenon exploits just as much as what the grammar offers as available resource, in a conceptually simple way.

We believe that this unexpected result is of interest to many syntacticians and semanticists. For this reason, we have decided to write this paper as a self-contained piece, with section 2 consisting of a quick introduction to Hybrid Type-Logical Grammar (Hybrid TLG; Kubota and Levine 2020), the version of TLG we adopt. This section also introduces a new notation for derivations using a dedicated λ-calculus for writing syntactic proofs. This proof term notation not only makes the sometimes tedious TLG derivations easier to read, but it also plays a crucial role in the formulation of the technical analysis of extraction pathway marking. Section 3 then presents the analysis of extraction pathway marking in full detail, taking the patterns of complementizer marking in Irish from McCloskey (1979) as a main target of analysis. After presenting the core formal analysis in section 3.2, section 3.3 addresses a recent claim by van Urk and Richards (2015) and van Urk (2020) that a proper analysis of extraction pathway marking requires both movement and feature checking. We argue that that the facts from Dinka and other languages that lead these authors to argue for their 'two-component' analysis of successive cyclicity are amenable to a *single*-component *non*-cyclic analysis of *wh*-dependency, once we take into account the interactions between the proof-theoretic properties of extraction and the independently motivated semantic properties of the relevant phenomena. This is followed by a brief comparison with a feature percolation analysis in HPSG in section 3.4, and section 4 concludes the paper.

2 Long-distance dependencies in Hybrid TLG

This section is meant to serve two purposes: to introduce Hybrid TLG as a syntactic framework and to illustrate its workings with an analysis of pied-piping in relative clauses in English. The choice of the empirical phenomenon is motivated by the fact that pied-piping exhibits properties of both 'overt' and 'covert' movement in derivational syntax. A recasting of the movement-based analysis from mainstream generative syntax in Hybrid TLG—building on an earlier analysis by Morrill (1994)—illustrates clearly the way in which TLG handles complex mapping between form and meaning. There is already substantial literature on linguistic applications of TLG (see, e.g., Morrill (1994); Carpenter (1997); Kubota and Levine (2020)), and readers are encouraged to refer to these sourcs for more information about TLG as a syntactic framework. Handbook articles such as Moortgat (2011, 2014) and Kubota (2021) are also useful sources of reference.

The full system of Hybrid TLG comprises three logical connectives $/$, $\backslash$, and $\upharpoonright$, and has Elimination and Introduction rules for all these. However, since the linguistic phenomena we deal with in this paper do not involve hypothetical reasoning with the directional slashes / and \, our presentation below focuses on the way in which the directional slashes / and \ are used for licensing local function-argument structures and on the use of the $\upharpoonright$ connective for modeling 'movement' operations (this corresponds to the system introduced in Section 2.3 of Kubota and Levine (2020)). The more complex Introduction rules for / and \ are discussed only briefly in section 2.4.

2.1 AB grammar

We start with a simple fragment called the *AB grammar* (Ajdukiewicz 1935; Bar-Hillel 1953), consisting of just two syntactic rules in (1):

(1) a. Forward Slash Elimination

$$\frac{\mathsf{a};\ A/B \quad \mathsf{b};\ B}{\mathsf{a} \bullet \mathsf{b};\ A}/\mathrm{E}$$

b. Backward Slash Elimination

$$\frac{\mathsf{b};\ B \quad \mathsf{a};\ B\backslash A}{\mathsf{b} \bullet \mathsf{a};\ A}\backslash\mathrm{E}$$

With a somewhat minimal lexicon in (2), we can license a simple transitive verb sentence (3) as in (4). The two slashes / and \ are used to form complex syntactic categories, or, syntactic types, indicating valence information: the transitive verb *loves* is assigned the syntactic type (NP\S)/NP since it first combines an NP to its right (i.e. the direct object) and then another NP to its right (i.e. the subject).

(2) a. $\mathsf{john};\ \mathrm{NP}$
b. $\mathsf{mary};\ \mathrm{NP}$
c. $\mathsf{ran};\ \mathrm{NP}\backslash\mathrm{S}$
d. $\mathsf{loves};\ (\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP}$

(3) John loves Mary.

(4)

$$\dfrac{\mathsf{john};\ \mathrm{NP} \qquad \dfrac{\mathsf{mary};\ \mathrm{NP} \quad \mathsf{loves};\ (\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP}}{\mathsf{loves} \bullet \mathsf{mary};\ \mathrm{NP}\backslash\mathrm{S}}/\mathrm{E}}{\mathsf{john} \bullet \mathsf{loves} \bullet \mathsf{mary};\ \mathrm{S}}\backslash\mathrm{E}$$

Syntactic types are defined recursively. For the AB grammar, this can be concisely written using the so-called 'BNF notation' as follows (the exact choice of the set of basic types is an empirical question):

(5) $\mathcal{A} := \{\ \mathrm{S, NP, N, \ldots}\}$ (atomic type)
$\mathcal{T} := \mathcal{A} \mid \mathcal{T}\backslash\mathcal{T} \mid \mathcal{T}/\mathcal{T}$ (type)

In words, anything that is an atomic type is a type, and any complex expression of form $A\backslash B$ or A/B where A and B are both types is a type.

As should already be clear in the above illustration, categorial grammar lexicalizes the valence (or subcategorization) properties of linguistic expressions, and this is transparently represented in the syntactic types of functional expressions (such as verb lexical entries). Here are some more sample lexical entries:

(6) a. $\mathsf{ran};\ \mathrm{NP}\backslash\mathrm{S}$

b. $\mathsf{read};\ (\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP}$

c. $\mathsf{introduces};\ (\mathrm{NP}\backslash\mathrm{S})/\mathrm{PP}/\mathrm{NP}$

2.2 Syntax-semantics interface

Assuming the standard recursive definition of semantic types as in (7) (with basic types e (individuals) and t (truth values) for an extensional fragment), we can define the function Sem that returns, for each syntactic type given as input, its semantic type, as in (8) and (9).

(7) a. $\mathcal{A}_\sigma := \{\ e, t\ \}$ (atomic semantic type)

b. $\mathcal{T}_\sigma := \mathcal{A}_\sigma \mid \mathcal{T}_\sigma \to \mathcal{T}_\sigma$ (semantic type)

(8) (Base Case)

a. $\mathsf{Sem}(\mathrm{NP}) = \mathsf{Sem}(\mathrm{PP}) = e$

b. $\mathsf{Sem}(\mathrm{N}) = e \to t$

c. $\mathsf{Sem}(\mathrm{S}) = t$

(9) (Recursive Clause)

For any complex syntactic type of the form A/B (or $B\backslash A$),

$\mathsf{Sem}(A/B)\ (= \mathsf{Sem}(B\backslash A)) = \mathsf{Sem}(B) \to \mathsf{Sem}(A)$

For example, assuming that VP adverbs such as *quickly* are of type $(\mathrm{NP}\backslash\mathrm{S})\backslash(\mathrm{NP}\backslash\mathrm{S})$, we can determine their semantic type based on the syntactic type by following the definitions in (7)–(9):

(10)
$$\begin{aligned}&\mathsf{Sem}((\mathrm{NP}\backslash\mathrm{S})\backslash(\mathrm{NP}\backslash\mathrm{S}))\\ &= \mathsf{Sem}(\mathrm{NP}\backslash\mathrm{S}) \to \mathsf{Sem}(\mathrm{NP}\backslash\mathrm{S})\\ &= (\mathsf{Sem}(\mathrm{NP}) \to \mathsf{Sem}(\mathrm{S})) \to (\mathsf{Sem}(\mathrm{NP}) \to \mathsf{Sem}(\mathrm{S}))\\ &= (e \to t) \to (e \to t)\end{aligned}$$

In other words, the syntactic type $(\mathrm{NP}\backslash\mathrm{S})\backslash(\mathrm{NP}\backslash\mathrm{S})$ transparently represents the semantic type of a VP modifier as an $e \to t$ property modifier.

Syntactic rules with semantics can then be written as in (11) (where the semantic effect of these rules is *function application*) and a sample derivation with semantic annotation is given in (12).

(11) a. Forward Slash Elimination

$$\frac{\mathsf{a};\ \boldsymbol{\mathcal{F}};\ A/B \quad \mathsf{b};\ \boldsymbol{\mathcal{G}};\ B}{\mathsf{a} \bullet \mathsf{b};\ \boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{G}});\ A}/\mathrm{E}$$

b. Backward Slash Elimination

$$\frac{\mathsf{b};\ \boldsymbol{\mathcal{G}};\ B \quad \mathsf{a};\ \boldsymbol{\mathcal{F}};\ B\backslash A}{\mathsf{b} \bullet \mathsf{a};\ \boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{G}});\ A}\backslash\mathrm{E}$$

(12)
$$\dfrac{\begin{matrix}\mathsf{john};\\ \mathbf{j};\mathrm{NP}\end{matrix} \quad \dfrac{\dfrac{\mathsf{chased};\ \mathbf{chase};\ (\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP} \quad \mathsf{mary};\ \mathbf{m};\ \mathrm{NP}}{\mathsf{chased} \bullet \mathsf{mary};\ \mathbf{chase}(\mathbf{m});\ \mathrm{NP}\backslash\mathrm{S}}/\mathrm{E} \quad \begin{matrix}\mathsf{patiently};\\ \mathbf{patiently};\\ (\mathrm{NP}\backslash\mathrm{S})\backslash(\mathrm{NP}\backslash\mathrm{S})\end{matrix}}{\mathsf{chased} \bullet \mathsf{mary} \bullet \mathsf{patiently};\ \mathbf{patiently}(\mathbf{chase}(\mathbf{m}));\ \mathrm{NP}\backslash\mathrm{S}}\backslash\mathrm{E}}{\mathsf{john} \bullet \mathsf{chased} \bullet \mathsf{mary} \bullet \mathsf{patiently};\ \mathbf{patiently}(\mathbf{chase}(\mathbf{m}))(\mathbf{j});\ \mathrm{S}}\backslash\mathrm{E}$$

2.3 Adding the vertical slash for 'movement'

The AB grammar introduced above deals with local licensing of arguments via the Elimination rules for / and \. This roughly corresponds to simple phrase structure grammar (or context-free grammar) without 'movement' operations. In order to model phenomena that involve both 'covert' and 'overt' movement (in the derivational terminology), we need to extend the underlying logical formalism. Hybrid TLG follows a relatively recent strand of research in the CG literature in addressing this issue by employing functional expressions in the prosodic representations of linguistic signs written as λ-terms (Oehrle 1994; de Groote 2001; Muskens 2003; Mihaliček and Pollard 2012).

Building on this tradition, we introduce into our system a new connective $\upharpoonright$ called the *vertical slash*, for order-insensitive mode of implication (as with /, we write the argument to the right for $\upharpoonright$). For this connective, we posit the following two rules:

(13) a. Vertical Slash Introduction

$$\dfrac{\begin{array}{ccc} \vdots & \underline{[\varphi;\ x;\ A]^n} & \vdots \\ \vdots & \vdots & \vdots \\ \hline & \mathsf{b};\ \mathcal{F};\ B & \end{array}}{\lambda\varphi.\mathsf{b};\ \lambda x.\mathcal{F};\ B\upharpoonright A}\ {\upharpoonright}\mathrm{I}^n$$

b. Vertical Slash Elimination

$$\dfrac{\mathsf{a};\ \mathcal{F};\ A\upharpoonright B \quad \mathsf{b};\ \mathcal{G};\ B}{\mathsf{a}(\mathsf{b});\ \mathcal{F}(\mathcal{G});\ A}\ {\upharpoonright}\mathrm{E}$$

As an illustration of how this extended system works, we start with a simple analysis of English relative clauses. The key idea here is that the new rules we have just introduced enable us to 'reason about' linguistic expressions in which some material is missing inside. For example, in (14), the body of the relative clause *Bill criticized* __ is analyzed as S$\upharpoonright$NP, a sentence missing an NP inside.

(14) the guy who Bill criticized __

As a starter, we posit the following entry for the relative pronoun *who* in which both the semantics and the prosody are higher-order functions.

(15) $\lambda\sigma.\mathsf{who} \bullet \sigma(\boldsymbol{\epsilon});\ \lambda P\lambda Q\lambda u.Q(u) \wedge P(u);\ (\mathrm{N}\backslash\mathrm{N})\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP})$

We can then license (16) for (14) (the dotted lines in (16) just show the β-reduction steps for the prosodic term, and are not part of the syntactic derivation; in what follows, VP is an abbreviation for NP\S).

(16)

$$\dfrac{\begin{array}{l}\mathsf{guy};\\ \mathbf{guy};\\ \mathrm{N}\end{array} \qquad \begin{array}{c} \dfrac{\begin{array}{l}\lambda\sigma.\mathsf{who} \bullet \sigma(\boldsymbol{\epsilon});\\ \lambda P\lambda Q\lambda u.Q(u) \wedge P(u);\\ (\mathrm{N}\backslash\mathrm{N})\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP})\end{array} \quad ① \to \dfrac{\dfrac{\begin{array}{l}\mathsf{bill};\\ \mathbf{b};\\ \mathrm{NP}\end{array} \quad \dfrac{\begin{array}{l}\mathsf{criticized};\\ \mathbf{criticize};\mathrm{VP/NP}\end{array} \quad \begin{bmatrix}\varphi_0;\\ x;\mathrm{NP}\end{bmatrix}^1}{\mathsf{criticized} \bullet \varphi_0;\ \mathbf{criticize}(x);\ \mathrm{VP}}\,/\mathrm{E}}{\mathsf{bill} \bullet \mathsf{criticized} \bullet \varphi_0;\ \mathbf{criticize}(x)(\mathbf{b});\ \mathrm{S}}\,\backslash\mathrm{E}}{\lambda\varphi_0.\mathsf{bill} \bullet \mathsf{criticized} \bullet \varphi_0;\ \lambda x.\mathbf{criticize}(x)(\mathbf{b});\ \mathrm{S}\upharpoonright\mathrm{NP}}\,{\upharpoonright}\mathrm{I}^1}{\lambda\sigma[\mathsf{who} \bullet \sigma(\boldsymbol{\epsilon})](\lambda\varphi_0.\mathsf{bill} \bullet \mathsf{criticized} \bullet \varphi_0);\ \lambda Q\lambda u.Q(u) \wedge \mathbf{criticize}(u)(\mathbf{b});\ \mathrm{N}\backslash\mathrm{N}}\,{\upharpoonright}\mathrm{E} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \mathsf{who} \bullet \lambda\varphi_0[\mathsf{bill} \bullet \mathsf{criticized} \bullet \varphi_0](\boldsymbol{\epsilon});\ \lambda Q\lambda u.Q(u) \wedge \mathbf{criticize}(u)(\mathbf{b});\ \mathrm{N}\backslash\mathrm{N} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \mathsf{who} \bullet \mathsf{bill} \bullet \mathsf{criticized} \bullet \boldsymbol{\epsilon};\ \lambda Q\lambda u.Q(u) \wedge \mathbf{criticize}(u)(\mathbf{b});\ \mathrm{N}\backslash\mathrm{N} \end{array}}{\mathsf{guy} \bullet \mathsf{who} \bullet \mathsf{bill} \bullet \mathsf{criticized} \bullet \boldsymbol{\epsilon};\ \lambda u.\mathbf{guy}(u) \wedge \mathbf{criticize}(u)(\mathbf{b});\ \mathrm{N}}\,\backslash\mathrm{E}$$

The derivation in (16) can be paraphrased in prose as follows.

- The NP with prosody φ_0 is a hypothetically assumed NP (the angle brackets around it indicate its status as such). With this hypothesis, we derive a complete S corresponding to the body of the relative clause *Bill criticized* __ (immediately above ①).
- The crucial step is the next one (①). At this point, the hypothesis is *withdrawn* with the $\upharpoonright$-Introduction rule. This yields an S$\upharpoonright$NP, a sentence containing an NP-type gap. The string position of the gap is kept track of by λ-binding the prosodic variable φ_0.
- The relative pronoun, with the lexical specification in (15), then takes this gapped sentence as its first argument and returns a backward nominal modifier of type N\N. (Semantically, the relative pronoun dentoes an intersective modifier of two properties.)

The final step where the relative pronoun takes a gapped sentence as argument perhaps requires some comment. The key point here is that the prosodic specification of the relative pronoun is a higher-order function that combines strings in a particular way. Specifically, its first argument σ is the gapped sentence (itself a function of type $\mathbf{st} \to \mathbf{st}$, that is, a function that maps a string into another string). It feed an empty string $\boldsymbol{\epsilon}$ to σ, thereby filling in the embedded gap position, and concatenates the string who in front of the string thus obtained. For the purpose of exposition, the relevant β-reduction steps are explicitly shown in the dotted line part in (16).

An important property of this analysis is that the gap can be deeply embedded inside the relative clause. Hypothetical reasoning with the vertical slash works exactly in the same way in the simple example above in which the gap corresponds to a local argument position and the more complex example in (17) in which the gap is located in an embedded clause with multiple levels of embedding.

(17) the guy who John thinks Mary said Bill criticized __ yesterday

The derivation for (17) is shown in (18).

(18)

$$
\dfrac{
\begin{array}{l}\mathsf{john};\\ \mathbf{j};\\ \mathrm{NP}\end{array}
\quad
\dfrac{
\begin{array}{l}\mathsf{thinks};\\ \mathbf{think};\\ \mathrm{VP/S}\end{array}
\quad
\dfrac{
\begin{array}{l}\mathsf{mary};\\ \mathbf{m};\\ \mathrm{NP}\end{array}
\quad
\dfrac{
\begin{array}{l}\mathsf{said};\\ \mathbf{say};\\ \mathrm{VP/S}\end{array}
\quad
\dfrac{
\dfrac{
\begin{array}{l}\mathsf{bill};\\ \mathbf{b};\\ \mathrm{NP}\end{array}
\quad
\dfrac{
\begin{array}{l}\mathsf{criticized};\\ \mathbf{criticize};\\ \mathrm{VP/NP}\end{array}
\quad
\left[\begin{array}{l}\varphi_0;\\ x;\\ \mathrm{NP}\end{array}\right]^1
}{
\begin{array}{l}\mathsf{criticized} \bullet \varphi_0;\\ \mathbf{criticize}(x);\mathrm{VP}\end{array}
}/\mathrm{E}
}{
\begin{array}{l}\mathsf{bill} \bullet \mathsf{criticized} \bullet \varphi_0;\\ \mathbf{criticize}(x)(\mathbf{b});\mathrm{S}\end{array}
}\backslash\mathrm{E}
\quad
\begin{array}{l}\mathsf{yesterday};\\ \mathbf{yest};\\ \mathrm{S\backslash S}\end{array}
}{
\begin{array}{l}\mathsf{bill} \bullet \mathsf{criticized} \bullet \varphi_0 \bullet \mathsf{yesterday};\\ \mathbf{yest}(\mathbf{criticize}(x)(\mathbf{b}));\mathrm{S}\end{array}
}\backslash\mathrm{E}
}{
\begin{array}{l}\mathsf{said} \bullet \mathsf{bill} \bullet \mathsf{criticized} \bullet \varphi_0 \bullet \mathsf{yesterday};\\ \mathbf{say}(\mathbf{yest}(\mathbf{criticize}(x)(\mathbf{b})));\mathrm{VP}\end{array}
}/\mathrm{E}
}{
\begin{array}{l}\mathsf{mary} \bullet \mathsf{said} \bullet \mathsf{bill} \bullet \mathsf{criticized} \bullet \varphi_0 \bullet \mathsf{yesterday};\\ \mathbf{say}(\mathbf{yest}(\mathbf{criticize}(x)(\mathbf{b})))(\mathbf{m});\mathrm{S}\end{array}
}\backslash\mathrm{E}
}{
\begin{array}{l}\mathsf{thinks} \bullet \mathsf{mary} \bullet \mathsf{said} \bullet \mathsf{bill} \bullet \mathsf{criticized} \bullet \varphi_0 \bullet \mathsf{yesterday};\\ \mathbf{think}(\mathbf{say}(\mathbf{yest}(\mathbf{criticize}(x)(\mathbf{b})))(\mathbf{m}));\mathrm{VP}\end{array}
}/\mathrm{E}
}{
\dfrac{
\begin{array}{l}\mathsf{john} \bullet \mathsf{thinks} \bullet \mathsf{mary} \bullet \mathsf{said} \bullet \mathsf{bill} \bullet \mathsf{criticized} \bullet \varphi_0 \bullet \mathsf{yesterday};\\ \mathbf{think}(\mathbf{say}(\mathbf{yest}(\mathbf{criticize}(x)(\mathbf{b})))(\mathbf{m}))(\mathbf{j});\mathrm{S}\end{array}
}{
\begin{array}{l}\lambda\varphi_0.\mathsf{john} \bullet \mathsf{thinks} \bullet \mathsf{mary} \bullet \mathsf{said} \bullet \mathsf{bill} \bullet \mathsf{criticized} \bullet \varphi_0 \bullet \mathsf{yesterday};\\ \lambda x.\mathbf{think}(\mathbf{say}(\mathbf{yest}(\mathbf{criticize}(x)(\mathbf{b})))(\mathbf{m}))(\mathbf{j});\mathrm{S{\upharpoonright}NP}\end{array}
}{\upharpoonright}\mathrm{I}^1
}\backslash\mathrm{E}
$$

The key point here is that a gapped sentence of type S↾NP can be derived for the body of the relative clause *John thinks Mary said Bill criticized* __ *yesterday* in exactly the same way as above, with hypothetical reasoning with the vertical slash.

The extension of the AB grammar with the new connective ↾ necessitates a revision to the definition of syntactic types and the mapping from syntactic types to semantic types. In addition, the grammar now recognizes not just simple strings (of type **st**) but also functions that compose such strings in particular ways as admissible prosodic representations of linguistic expressions. We therefore need to define the mapping from syntactic types to prosodic types as well. The new definitions are given in (19)–(23).

Syntactic types:

(19) $\mathcal{A} := \{\ \mathrm{S}, \mathrm{NP}, \mathrm{N}, \ldots\}$ (atomic type)
$\mathcal{D} := \mathcal{A} \mid \mathcal{D}\backslash\mathcal{D} \mid \mathcal{D}/\mathcal{D}$ (directional type)
$\mathcal{T} := \mathcal{D} \mid \mathcal{T}{\upharpoonright}\mathcal{T}$ (type)

Semantic types:

(20) (Base Case)

a. $\mathsf{Sem}(\mathrm{NP}) = \mathsf{Sem}(\mathrm{PP}) = e$

b. $\mathsf{Sem}(\mathrm{N}) = e \to t$

c. $\mathsf{Sem}(\mathrm{S}) = t$

(21) (Recursive Clause)
For any complex syntactic type of the form A/B (or $B\backslash A$, $A{\upharpoonright}B$),
$\mathsf{Sem}(A/B)\ (= \mathsf{Sem}(B\backslash A) = \mathsf{Sem}(A{\upharpoonright}B)\ ) = \mathsf{Sem}(B) \to \mathsf{Sem}(A)$

Prosodic types:

(22) (Base Case)
For any directional type $\mathcal{D}$, $\mathsf{Pros}(\mathcal{D}) = \mathbf{st}$ (with **st** for 'strings').

(23) (Recursive Clause)
For any complex syntactic type $A{\upharpoonright}B$ involving $\upharpoonright$,
$\mathsf{Pros}(A{\upharpoonright}B) = \mathsf{Pros}(B) \to \mathsf{Pros}(A)$.

Note that $\mathcal{D}$ in (19) replaces $\mathcal{T}$ in the earlier definition of syntactic types in (5). The set of syntactic types is defined on top of the set of directional types (i.e., the complete set of syntactic types in the earlier definition) as in the final clause in (19). This ensures that a vertical slash cannot occur under a directional slash. Thus, S/(S↾NP) is not a well-formed syntactic type. One way to make sense of this restriction is to think of it as a 'filter' on uninterpretable prosodic objects. An expression with syntactic type X/(Y↾Z) would have to concatenate a string to the left of a function of type **st** into **st**, but that obviously doesn't make sense.

As the asymmetry between (21) and (23) should make clear, the three slashes /, \ and ↾ are all functional in the semantic domain, but only ↾ is functional in the prosodic domain. This asymmetry corresponds to the fact that lambda binding is involved in the prosody only for the Introduction rule for ↾ (see section 2.4 for the Introduction rules for / and \).

2.4 Hypothetical reasoning with the directional slashes

The key notion involved in the analysis of English relative clauses above is hypothetical reasoning, which is essentially a theoretical machinery for 'reasoning about' complex linguistic expressions in which some material is missing from where it is supposed to appear. In the full version of Hybrid TLG, hypothetical reasoning is generalized to the directional slashes / and \ as well. For the sake of completeness of presentation, we show below the Introduction rules for / and \, and briefly discuss linguistic application of these rules.

The Slash Introduction rules for / and \ are formulated as in (24).

(24) a. Forward Slash Introduction

$$\dfrac{\dfrac{\vdots \quad \dfrac{[\varphi;\, x;\, A]^n}{\vdots} \quad \vdots}{\mathsf{b} \bullet \varphi;\, \boldsymbol{\mathcal{F}};\, B}}{\mathsf{b};\, \lambda x.\boldsymbol{\mathcal{F}};\, B/A}/\mathrm{I}^n$$

b. Backward Slash Introduction

$$\dfrac{\dfrac{\vdots \quad \dfrac{[\varphi;\, x;\, A]^n}{\vdots} \quad \vdots}{\varphi \bullet \mathsf{b};\, \boldsymbol{\mathcal{F}};\, B}}{\mathsf{b};\, \lambda x.\boldsymbol{\mathcal{F}};\, A\backslash B}\backslash\mathrm{I}^n$$

The difference between the Introduction rule for the vertical slash introduced above in (13a) and these rules is that in (24), the prosodic variable φ for the hypothesis is simply thrown away (instead of being λ-bound). The position of the missing expression is instead recorded in the forward vs. backward slash distinction in the syntactic type.

This is useful when one wants to assign a directional slash type for some string of words in which some material is missing at the periphery, instead of analyzing such expressions with functional prosodic types. For example, for the string *John loves* in the Right-node Raising example in (25), we want to assign the type S/NP so that it is directly conjoinable with another string *Bill hates* of the same type.

(25) $[_{\mathrm{S/NP}}$ John loves$]$, and $[_{\mathrm{S/NP}}$ Bill hates$]$, $[_{\mathrm{NP}}$ Mary$]$.

The derivation for the string *John loves* in type S/NP is shown in (26).

(26)

$$\text{①}\!\to\dfrac{\dfrac{\text{john}; \mathbf{j}; \mathrm{NP} \qquad \dfrac{\text{loves}; \mathbf{love}; (\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP} \quad [\varphi; x; \mathrm{NP}]^1}{\text{loves} \bullet \varphi; \mathbf{love}(x); \mathrm{NP}\backslash\mathrm{S}}\,/\mathrm{E}}{\text{john} \bullet \text{loves} \bullet \varphi; \mathbf{love}(x)(\mathbf{j}); \mathrm{S}}\,\backslash\mathrm{E}}{\text{john} \bullet \text{loves}; \lambda x.\mathbf{love}(x)(\mathbf{j}); \mathrm{S/NP}}\,/\mathrm{I}^1$$

In prose:

- A complete sentence is formed with the hypothetical NP indexed 1. (This much is the same as in the earlier (16).)
- At the next step (①), the hypothesis is withdrawn just as in (16), but here the string variable φ is thrown away, and the derived type is S/NP (with type **st** prosody). It is this syntactic type that tells us that this is a sentence missing an NP on the right.

2.5 Proof term notation of derivations

To facilitate the ensuing discussion, we introduce here an alternative notation of derivations, one in which a derivation/proof can be written as a single formal object, specifically a lambda term. This corresponds to Abstract Syntax in Abstract Categorial Grammar (de Groote 2001). It exploits the theoretical result in TLG research building on the so-called Curry-Howard Isomorphism (Howard 1969), which states that there is a one-to-one correspondence between proofs and lambda terms in a simply typed lambda calculus. Essentially, an Elimination step (in natural deduction) in a proof corresponds to function application in the lambda calculus and an Introduction step corresponds to lambda abstraction. With Hybrid TLG, this lambda calculus for writing syntactic proofs needs to be extended to distinguish three types of function application ($\mathrm{app}_{/}$, $\mathrm{app}_{\backslash}$, and $\mathrm{app}_{\upharpoonright}$), and three types of lambda abstraction ($\lambda_{/}$, $\lambda_{\backslash}$, and $\lambda_{\upharpoonright}$), corresponding to the three slashes in the system.[3]

As an illustration, consider the derivation (27) (= (16) above) for a simple relative clause from the previous section.

[3] This lambda calculus can be thought of as an extension of the bidirectional lambda calculus for the Lambek calculus proposed by Buszkowski (1987) and Wansing (1992). Studying the formal properties of this lambda calculus is an interesting topic on its own, but we leave this task for another occasion.

(27)

$$\dfrac{\text{guy};\ \mathbf{guy};\ \mathrm{N} \qquad \dfrac{\begin{array}{l}\lambda\sigma.\text{who} \bullet \sigma(\epsilon);\\ \lambda P\lambda Q\lambda u.\\ \quad Q(u) \wedge P(u);\\ (\mathrm{N}\backslash\mathrm{N})\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP})\end{array} \qquad \dfrac{\dfrac{\begin{array}{l}\text{bill};\\ \mathbf{b};\\ \mathrm{NP}\end{array} \qquad \dfrac{\begin{array}{l}\text{criticized};\\ \mathbf{criticize}; \mathrm{VP/NP}\end{array} \qquad \begin{bmatrix}\varphi_0;\\ x; \mathrm{NP}\end{bmatrix}^1}{\begin{array}{l}\text{criticized} \bullet \varphi_0;\\ \mathbf{criticize}(x); \mathrm{VP}\end{array}}/\mathrm{E}}{\begin{array}{l}\text{bill} \bullet \text{criticized} \bullet \varphi_0;\\ \mathbf{criticize}(x)(\mathbf{b}); \mathrm{S}\end{array}}\backslash\mathrm{E}}{\begin{array}{l}\lambda\varphi_0.\text{bill} \bullet \text{criticized} \bullet \varphi_0;\\ \lambda x.\mathbf{criticize}(x)(\mathbf{b}); \mathrm{S}\upharpoonright\mathrm{NP}\end{array}}\upharpoonright\mathrm{I}^1}{\text{who} \bullet \text{bill} \bullet \text{criticized} \bullet \epsilon;\ \lambda Q\lambda u.Q(u) \wedge \mathbf{criticize}(u)(\mathbf{b});\ \mathrm{N}\backslash\mathrm{N}}\upharpoonright\mathrm{E}}{\text{guy} \bullet \text{who} \bullet \text{bill} \bullet \text{criticized} \bullet \epsilon;\ \lambda u.\mathbf{guy}(u) \wedge \mathbf{criticize}(u)(\mathbf{b});\ \mathrm{N}}\backslash\mathrm{E}$$

We first posit the following constants (written in small capitals) for each of the tripartite linguistic signs in the lexicon (in what follows, TV is an abbreviation for (NP\S)/NP):

(28) $\mathbf{CRITICIZED}_{\mathrm{TV}} = \text{criticized};\ \mathbf{past}(\mathbf{criticize});\ \mathrm{TV}$
$\mathbf{WHO}_{(\mathrm{N}\backslash\mathrm{N})\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP})} = \lambda\sigma.\text{who} \bullet \sigma(\epsilon);\ \lambda P\lambda Q\lambda u.Q(u) \wedge P(u);\ (\mathrm{N}\backslash\mathrm{N})\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP})$
$\mathbf{BILL}_{\mathrm{NP}} = \text{bill};\ \mathbf{b};\ \mathrm{NP}$
$\mathbf{GUY}_{\mathrm{N}} = \text{guy};\ \mathbf{guy};\ \mathrm{N}$

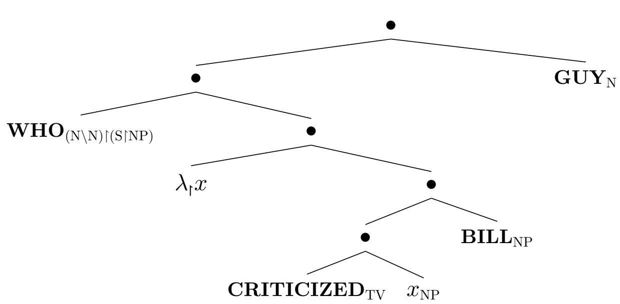
Then, by replacing Slash Elimination by function application and Slash Introduction by lambda abstraction in (27), we obtain the following complex lambda term whose syntactic form is isomorphic (i.e., stands in a one-to-one relation) to the natural deduction proof in (27) (the variety of application rule is omitted, since this information is unambiguously recoverable from the syntactic type of the function):

(29) $\mathbf{WHO}_{(\mathrm{N}\backslash\mathrm{N})\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP})}(\lambda_{\upharpoonright}x.\mathbf{CRITICIZED}_{\mathrm{TV}}(x_{\mathrm{NP}})(\mathbf{BILL}_{\mathrm{NP}}))(\mathbf{GUY}_{\mathrm{N}})$

In effect, (29) displays the entire proof narrative exhibited in (27) as a single object: the function corresponding to *criticized* is saturated, with its variable argument undergoing abstraction, yielding an eligible argument for the relative pronoun *who*. Note here that the variable x_{NP} in (29) is a variable in the syntactic logic and is thus unrelated to the x in the semantic component of the hypothesis in (27); we use the same variable letter only for expository convenience.

To make it clear that (29) represents underlying semantic composition, and to enhance readability, here is an alternative notation for the term (29) in the form of a binary tree:

(30)

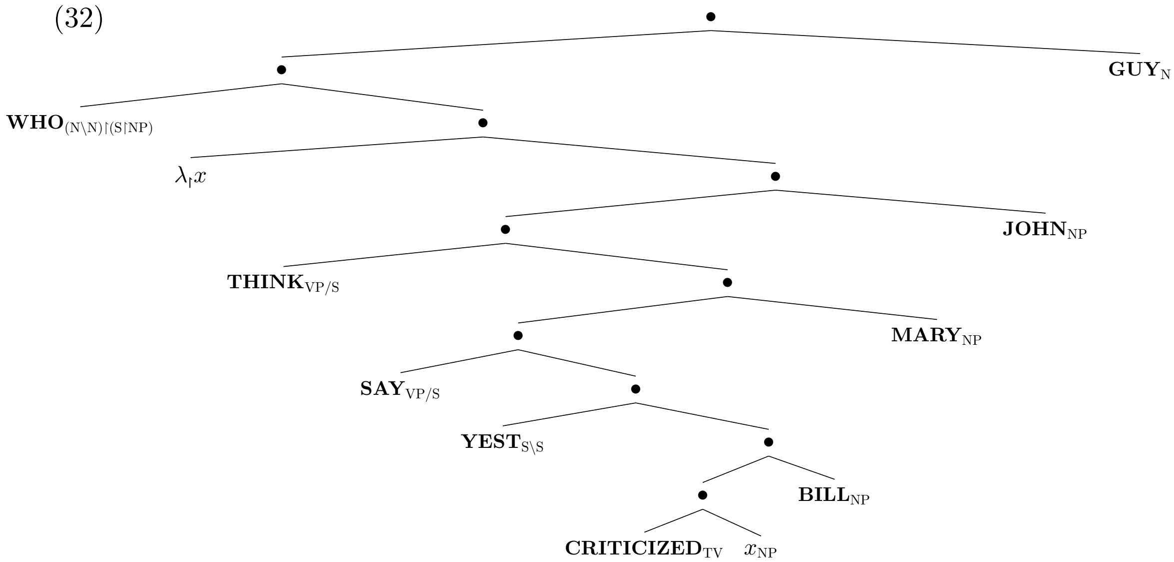


Readers familiar with derivational approaches to syntax might see a resemblance to LF structure. The correspondence to the natural deduction proof tree in (27) should also be easier to see in this format.

The proof term notation is a compact representation of derivations that represents the underlying combinatorics transparently. As we demonstrate below with pied-piping, this is especially useful in the analyses of complex empirical phenomena involving hypothetical reasoning with the vertical slash (roughly corresponding to 'syntactic movement') extensively. Before moving on, here is just another illustration, for the long-distance extraction example in (17).

(31) $\mathbf{WHO}_{(\mathrm{N}\backslash\mathrm{N})\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP})}(\lambda_{\upharpoonright}x.\mathbf{THINK}_{\mathrm{VP/S}}(\mathbf{SAY}_{\mathrm{VP/S}}$
$(\mathbf{YEST}_{\mathrm{S}\backslash\mathrm{S}}(\mathbf{CRITICIZED}_{\mathrm{TV}}(x_{\mathrm{NP}})(\mathbf{BILL}_{\mathrm{NP}})))(\mathbf{MARY}_{\mathrm{NP}}))(\mathbf{JOHN}_{\mathrm{NP}}))(\mathbf{GUY}_{\mathrm{N}})$

(32)



Note that 'extraction' of the NP corresponds to the lambda abstraction of the variable x_{NP}. This variable is embedded deeply inside a complex lambda term, and this indicates that this example involves long-distance dependency between the gap and the filler *wh*-phrase that takes the gapped sentence as an argument.

2.6 Pied-piping as 'overt and covert' movement

In the analysis of English relative clauses above, the semantic and syntactic linkage between the extracted material, the *wh* pronoun and the rest of the sentence is in effect built into the higher-order operator entry for the *wh* word of type $(\mathrm{N}\backslash\mathrm{N})\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP})$. In this section, we consider how this simple analysis can be extended to deal with pied-piping.

Pied-piping, whimsically named in Ross (1967, 24), is a species of extraction in which a *wh* pronoun does not directly correspond to a gap within the relative clause but is itself a subconstituent of a larger fronted constituent corresponding to the gap. The following data exemplify the most basic kinds of pied-piping:

(33) a. the guy [to **whom**] John spoke __ yesterday

b. the guy [to **whose** office] John walked __ yesterday

c. the guy [to **whose** sister] John spoke __ yesterday

More elaborate cases can be found, including Ross' famous example:

(34) the reports [[the height of the lettering on the covers of **which**] [the government prescribes __]]

This example makes it clear that the *wh*-phrase can be embedded inside the fronted expression arbitrarily deeply.

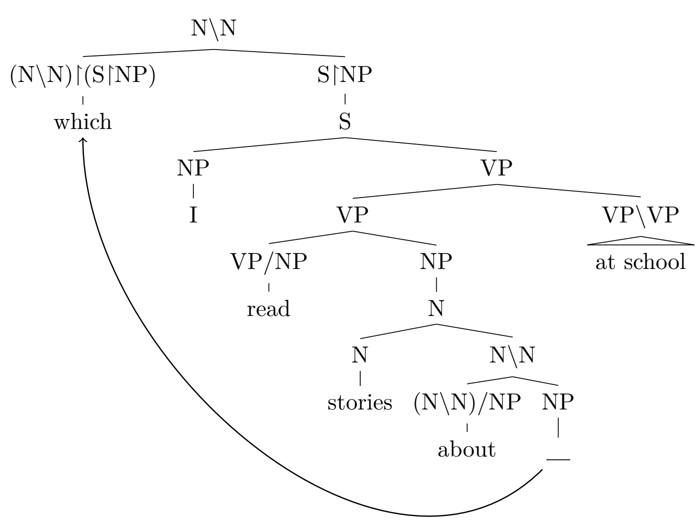
By comparing the pied-piping example in (35a) to its non-pied-piped counterpart in (35b), it should be clear that the semantic interpretation of pied-piping examples is exactly the same as the corresponding simpler examples in which only the *wh*-word is displaced.

(35) a. Castle Combe is the town [stories about **which**] I read __ at school.

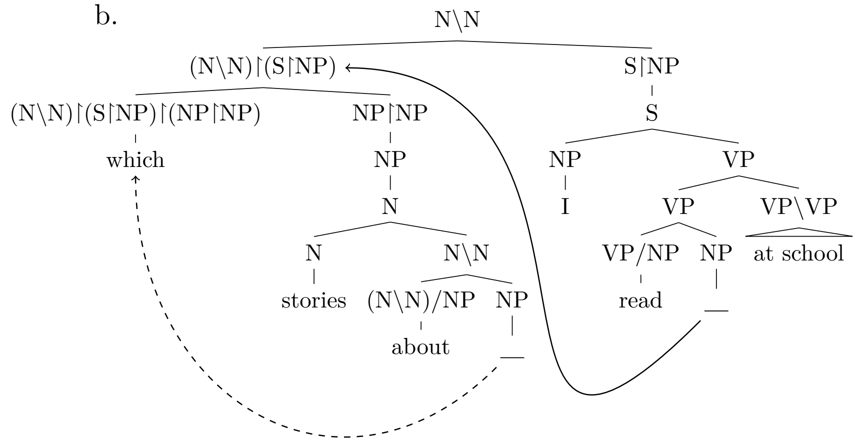
b. Castle Combe is the town **which** I read stories about __ at school.

This correspondence can be graphically represented in the following informal pictures (where the solid line indicates 'overt' movement and the dashed line indicates 'covert' movement):

(36) a.



b.



In the case of non-pied-piped relativization (36a), the filler and the gap have the same syntactic type. In contrast, in the pied-piping example (36b), the *wh*-pronoun that triggers relativization is embedded inside the filler, and it is this entire filler phrase that 'binds' the gap in the body of the relative clause. Here, as alluded to by the use of different types of 'movement arrows', the correspondence between the gap and the filler is a case of 'overt movement', just as with non-pied-piped relativization. By contrast, the identification of the whole *wh*-phrase that contains the *wh*-word as the 'operator' that triggers relativization is mediated by a 'covert movement'-like operation. In the latter, the string of the *wh*-word is embedded inside the filler phrase.

This can be formalized precisely by modifying the lexical entry for the *wh* operator as in (37) (the key idea here is due to Morrill (1994)).

(37) $\lambda\sigma_1\lambda\sigma_2.\sigma_1(\mathsf{whom}) \bullet \sigma_2(\epsilon)$;
$\lambda F\lambda P\lambda Q\lambda x.P(F(x)) \wedge Q(x); (\mathrm{N\backslash N})\upharpoonright(\mathrm{S}\upharpoonright\mathrm{X})\upharpoonright(\mathrm{X}\upharpoonright\mathrm{NP})$

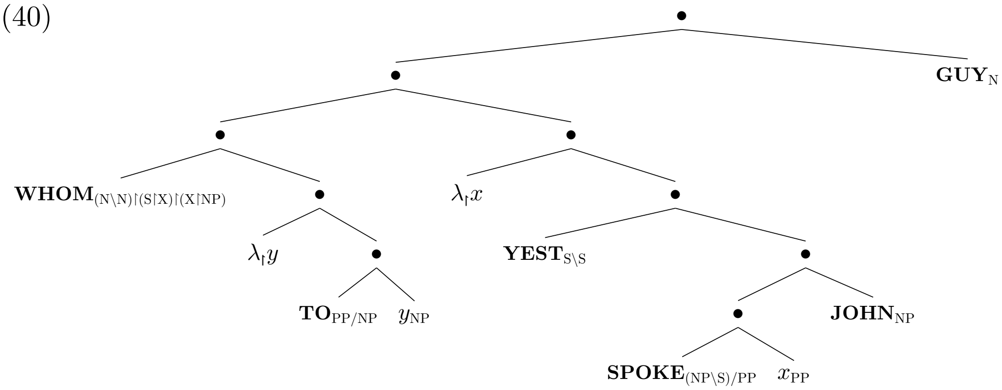
This says that the relative pronoun takes two arguments, some expression of type X missing an NP inside itself and an S missing an X, and then becomes a nominal modifier. A sample derivation for (33a) using this entry is shown in (38) (in natural deduction) and (39) (in the proof term format). Here, since the fronted phrase is a PP, X is instantiated as PP.

(38)

$$\dfrac{\dfrac{\begin{array}{l}\lambda\sigma_1\lambda\sigma_2.\sigma_1(\mathsf{whom}) \bullet \sigma_2(\epsilon);\\ \lambda F\lambda P\lambda Q\lambda x.P(F(x)) \wedge Q(x);\\ (\mathrm{N\backslash N})\upharpoonright(\mathrm{S}\upharpoonright\mathrm{X})\upharpoonright(\mathrm{X}\upharpoonright\mathrm{NP})\end{array} \quad \dfrac{\dfrac{\begin{array}{l}\mathsf{to};\\ \lambda x.x; \mathrm{PP/NP}\end{array} \quad \begin{bmatrix}\varphi_2;\\ y; \mathrm{NP}\end{bmatrix}^2}{\mathsf{to} \bullet \varphi_2; y; \mathrm{PP}}/\mathrm{E}}{\begin{array}{l}\lambda\varphi_2.\mathsf{to} \bullet \varphi_2;\\ \lambda y.y; \mathrm{PP}\upharpoonright\mathrm{NP}\end{array}}\upharpoonright\mathrm{I}^2}{\begin{array}{l}\lambda\sigma_2.\mathsf{to} \bullet \mathsf{whom} \bullet \sigma_2(\epsilon);\\ \lambda P\lambda Q\lambda x.P(x) \wedge Q(x); (\mathrm{N\backslash N})\upharpoonright(\mathrm{S}\upharpoonright\mathrm{PP})\end{array}}\upharpoonright\mathrm{E} \quad \dfrac{\begin{array}{c}\begin{bmatrix}\varphi_1;\\ x; \mathrm{PP}\end{bmatrix}^1\\ \vdots\end{array}}{\begin{array}{l}\lambda\varphi_1.\mathsf{john} \bullet \mathsf{spoke} \bullet\\ \quad \varphi_1 \bullet \mathsf{yesterday};\\ \lambda x.\mathbf{yest}(\mathbf{speak}(x)(\mathbf{j}));\\ \mathrm{S}\upharpoonright\mathrm{PP}\end{array}}\upharpoonright\mathrm{I}^1}{\mathsf{to} \bullet \mathsf{whom} \bullet \mathsf{john} \bullet \mathsf{spoke} \bullet \mathsf{yesterday};\ \lambda Q\lambda x.\mathbf{yest}(\mathbf{speak}(x)(\mathbf{j})) \wedge Q(x);\ \mathrm{N\backslash N}}\upharpoonright\mathrm{E}$$

(39) $\mathbf{WHOM}_{(\mathrm{N\backslash N})\upharpoonright(\mathrm{S}\upharpoonright\mathrm{X})\upharpoonright(\mathrm{X}\upharpoonright\mathrm{NP})}$
$(\lambda_{\upharpoonright}y.\mathbf{TO}_{\mathrm{PP/NP}}(y_{\mathrm{NP}}))(\lambda_{\upharpoonright}x.\mathbf{YEST}_{\mathrm{S\backslash S}}(\mathbf{SPOKE}_{(\mathrm{NP\backslash S})/\mathrm{PP}}(x_{\mathrm{PP}})(\mathbf{JOHN}_{\mathrm{NP}})))$

(40)



Note that this analysis involves two instances of hypothetical reasoning, corresponding to the 'overt' and 'covert' movement operations in the informal diagram in (36b). The hypothetical reasoning with the PP (indexed 1 in (38) and x_{PP} in (39)/(40)) is for forming a gapped sentence of type $\mathrm{S}\upharpoonright\mathrm{PP}$ that serves as the body of the relative clause. The hypothetical reasoning involving the NP hypothesis (indexed 2 in (38) and y_{NP} in (39)/(40)) is for identifying the location of the relative pronoun inside the fronted constituent *to whom*. The relativization operator defined in (37) fills in an empty string and the string of the relative pronoun (i.e., the string *whom*) in the positions of the two lambda-bound variables φ_1 and φ_2, reflecting the 'overt' and 'covert' movement statuses of the two hypothetical reasoning steps involved. In Hybrid TLG, 'covert' and 'overt' movement are handled by the same formal mechanism, and the difference between the two merely consists in whether an overt string is substituted for the bound variable position in the prosodic function that is given as an argument to the higher-orderr operator.

Since the 'in-situ' operator relationship between the relative pronoun and the fronted expression containing it is mediated by $\upharpoonright$, we predict that the *wh*-pronoun can be embedded inside the fronted constituent arbitrarily deeply. Thus, Ross's (1967) example can be accounted for in the same way as the simpler PP pied-piping example in (37) above. We show the derivation in proof term notation:

(41)
$$
\begin{aligned}
&\mathbf{WHICH}_{(\mathrm{N}\backslash\mathrm{N})\upharpoonright(\mathrm{S}\upharpoonright\mathrm{X})\upharpoonright(\mathrm{X}\upharpoonright\mathrm{NP})}\\
&\quad(\lambda_{\upharpoonright}y.\mathbf{THE}_{\mathrm{NP/N}}(\mathbf{HEIGHT}_{\mathrm{N/PP}}(\mathbf{OF}_{\mathrm{PP/NP}}\\
&\qquad(\mathbf{THE}_{\mathrm{NP/N}}(\mathbf{ON}_{(\mathrm{N}\backslash\mathrm{N})/\mathrm{NP}}(\mathbf{THE}_{\mathrm{NP/N}}(\mathbf{COVERS}_{\mathrm{N/PP}}(\mathbf{OF}_{\mathrm{PP/NP}}(y_{\mathrm{NP}}))))\\
&\qquad\qquad(\mathbf{LETTERING}_{\mathrm{N}}))))))\\
&\quad(\lambda_{\upharpoonright}x.\mathbf{PRESCRIBES}_{(\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP}}(x_{\mathrm{NP}})(\mathbf{THE}_{\mathrm{NP/N}}(\mathbf{GVT}_{\mathrm{N}})))\\
=\;&\mathsf{the} \bullet \mathsf{height} \bullet \mathsf{of} \bullet \mathsf{the} \bullet \mathsf{lettering} \bullet \mathsf{on} \bullet \mathsf{the} \bullet \mathsf{covers} \bullet \mathsf{of} \bullet \mathsf{which} \bullet\\
&\mathsf{the} \bullet \mathsf{government} \bullet \mathsf{prescribes};\\
&\lambda x.\mathbf{prescribe}(\mathbf{the}(\mathbf{height}(\mathbf{the}(\mathbf{on}(\mathbf{the}(\mathbf{covers}(x)))\\
&(\mathbf{lettering})))))(\mathbf{the}(\mathbf{gvt})); \mathrm{N}\backslash\mathrm{N}
\end{aligned}
$$

Here, X is instantiated as NP. The question of which syntactic type can be pied-piped is a rather thorny issue. As noted by Arnold and Godard (2021), even a descriptively correct generalization is unclear for well-studied languages such as English. We do not attempt to address this issue in this paper, since the analysis of pied-piping is not itself our central goal here.

3 Extraction pathway marking as proof structure marking

3.1 Irish complementizer marking

The syntactic literature has documented the existence of languages which distinctively mark the syntactic domains intervening between fillers and their associated gaps. Such facts were first identified as a cross-linguistic pattern in Zaenen (1983) for Icelandic, and have been reported to exist in typologically unrelated languages such as Chamorro (Chung 1982), French (Kayne and Pollock 1978), Icelandic (Zaenen 1983), Irish (McCloskey 1979) and Welsh (Borsley 2010).

The phenomenon can be illustrated most clearly by the choice of different types of complementizers in Irish reported in McCloskey (1979). For expository convenience, we illustrate the relevant empirical pattern here by a pseudo-language called Iringlish, which is like Irish in having the relevant distinction of two complementizers but otherwise is identical to English in all other respects.

We start with clausal embedding without any extraction. In this case, as shown in (42)–(43), the complementizers (the counterpart of *that* in English) are all realized as *goN*.

(42) I thought **goN** [he would be there].

(43) I said **goN** [I thought **goN** [he would be there]].

Iringlish has another form of complementizer *aL*, which is essentially in complementary distribution to *goN*. The distribution of *aL* is restricted to clauses that contain an undischarged gap site in one of its argument or adjunct positions. Thus, for example, in the following (44), the lower clause is marked by *goN*, but the higher clause, which is missing its subject argument corresponding to the head noun in the relative clause, is marked by *aL*:

(44) the man **aL** [__ thought **goN** [he would be there]]

The examples in (45)–(46), with a multiple chain of *aL* complementation, show that the linkage between the filler and the gap is registered over an arbitrary number of structural levels.

(45) the man **aL** [I thought **aL** [__ would be there]]

(46) the man **aL** [I said **aL** [I thought **aL** [__ would be there]]]

Regardless of the depth of the extraction, as soon as the gap site is identified, all lower clauses which themselves are not associated with an extraction will be marked by *goN*, a point illustrated in (44) and at still greater structural depth in (47).

(47) the man **aL** [he said **aL** [__ thought **goN** [he would be there]]]

An important property of extraction pathway marking, whose theoretical relevance is discussed in detail by Hukari and Levine (1995), is that it provides clear evidence for the existence of extraction of adjuncts—whose gap sites are harder to identify than argument gap sites. As shown by (48) and (49), extraction of adjuncts is mediated by exactly the same mechanism of extraction pathway marking as with extraction of arguments in Iringlish (and in Irish):

(48) the day **aL** [we were in Derry __]

(49) What time **aL** [did they come home __]?

Furthermore, when an adjunct is extracted from a multiply embedded position, all the clause boundaries between the extraction site and the filler are marked by *aL*:

(50) It was in Bethlehem **aL** [the prophecies said **aL** [the Saviour would be born __]].

3.2 Accounting for extraction pathway marking

The pattern displayed by Iringlish is quite simple: the form of the complementizer is sensitive to the existence of an unbound gap in the complement clause. But how can we encode this restriction? In mainstream syntax, such facts have been taken to provide evidence for the notion of successive cyclic movement. An HPSG analysis by Bouma et al. (2001) makes use of the SLASH feature propagation mechanism. A recent claim by van Urk and Richards (2015) and van Urk (2020) holds that *both* cyclic movement and feature checking are needed to account for the entire patterns of extraction pathway marking documented in the literature. We return to this specific claim in section 3.3 below, but for the time being we focus on the task of formulating an explicit analysis of the Iringlish pattern.

The apparent dilemma here is that neither cyclic movement nor feature percolation is native to the architecture of TLG. In the hypothetical reasoning-based analysis of extraction in TLG, the identification between the filler and the gap is mediated via a single instance of hypothetical reasoning. In this approach, nothing 'moves' literally (let alone in a successive cyclic way), nor is there any structure-manipulation operation or feature percolation of any sort. Note in particular that, under the proof theoretic setup, unlike tree representations in derivational syntax, brute-force structure manipulation is not an option to begin with (that would correspond to mechanically moving a premise in a well-fomed proof tree from one position to another on the basis of purely structural conditions). Thus, the empirical generalization of extraction pathway marking seems simply unstable. Is there a way out of this impasse?

It turns out that there is, and the analysis available in TLG is conceptually quite different from either the movement-based analysis in derivational syntax or the feature percolation analysis in nonderivational syntax. But we have to pay its price: we need to throw away one influential dogma that has dominated CG research over the past several decades. What we need to give up is the idea that the grammar cannot make access to the internal structures of syntactic proofs. While the origin of this idea is unclear, it is likely to stem from the idea in classical Montague Grammar that the translation language is an intermediate step that is in principle eliminable (see, e.g., Dowty et al. (1981) and Cooper (1983)). In the more contemporary version of TLG, syntactic derivations are literally proofs, and one normally doesn't think of structures of proofs to be meaningful objects of study (unless one is a proof theorist). So, this tradition continued to be influential in contemporary CG research up until present.

But once we look beyond linguistics, there is nothing wrong or dubious about studying the structures of proofs in a formal setting. On the contrary, this is one of the central topics in the proof-theoretic/type-theoretic tradition in logic and computer science (see, e.g., von Plato (2018) for a an overview; see also the linguist-friendly linear logic introduction by Crouch and van Genabith (2000)). It is straightforward to incorporate some of the well-understood techniques from this literature into Hybrid TLG. Specifically, we argue in what follows that there is an empirical application of such proof theoretic techniques, and that one concrete case comes from extraction pathway marking (see Asudeh and Crouch (2002) for an important precursor in the application of a proof theoretic technique to a linguistic problem—Fox's (2000) Scope Economy in their case). Interestingly, the analysis we propose achieves the same empirical coverage as the P&P or HPSG analyses, but it dispenses with the dedicated theoretical mechanisms in each in capturing the relevant empirical patterns. In our analysis, all we need is the independently motivated hypothetical reasoning-based analysis of filler-gap linkage and an assumption that certain functional words in some languages are sensitive to the intermediate statuses of syntactic proofs (that is, whether they contain an unwithdrawn hypothesis).

The proof term notation of derivations introduced above enables a concise formulation of the extraction pathway marking patterns exhibited by the Iringlish (or Irish) data from section 3.1. We illustrate this point with a fragment of Iringlish with the lexicon in (51).

(51) $\mathbf{WBT}_{\mathrm{NP}\backslash\mathrm{S}} = \mathsf{would} \bullet \mathsf{be} \bullet \mathsf{there};\ \lambda x.\mathbf{exist}(x, \mathbf{there});\ \mathrm{NP}\backslash\mathrm{S}$
$\mathbf{MAN}_{\mathrm{N}} = \mathsf{man};\ \mathbf{man};\ \mathrm{N}$
$\mathbf{THOUGHT}_{(\mathrm{NP}\backslash\mathrm{S})/\mathrm{S}'} = \mathsf{thought};\ \mathbf{thought};\ (\mathrm{NP}\backslash\mathrm{S})/\mathrm{S}'$
$\mathbf{SAID}_{(\mathrm{NP}\backslash\mathrm{S})/\mathrm{S}'} = \mathsf{said};\ \mathbf{said};\ (\mathrm{NP}\backslash\mathrm{S})/\mathrm{S}'$
$\mathbf{AL}_{\mathrm{S}'/\mathrm{S}} = \mathsf{aL};\ \lambda p.p;\ \mathrm{S}'/\mathrm{S}$
where for any α, $\mathbf{AL}(\alpha)$ is defined only if $fv_{\mathrm{X}_{+wh}}(\alpha) \neq \emptyset$
$\mathbf{GON}_{\mathrm{S}'/\mathrm{S}} = \mathsf{goN};\ \lambda p.p;\ \mathrm{S}'/\mathrm{S}$
where for any α, $\mathbf{GON}(\alpha)$ is defined only if $fv_{\mathrm{X}_{+wh}}(\alpha) = \emptyset$
$\mathbf{REL}_{(\mathrm{N}\backslash\mathrm{N})\upharpoonright(\mathrm{S}'\upharpoonright\mathrm{NP}_{+wh})} =$
$\lambda\sigma_2\lambda\varphi_2.\varphi_2 \bullet \sigma_2(\epsilon);\ \lambda P\lambda Q\lambda y.Q(y) \wedge P(y);\ (\mathrm{N}\backslash\mathrm{N})\upharpoonright(\mathrm{S}'|\mathrm{NP}_{+wh})$

The key components of this analysis are the restrictions imposed on *aL* and *goN* that refer to the structures of the terms given as their (first) arguments. fv_{Φ} is the standard, inductively defined function that returns all free variables contained in a term, except that it filters the output of the general purpose *fv* to type Φ (see the examples below). The feature $\pm wh$ on NPs is introduced here to distinguish NPs that are bound by extraction operators (relativizers, *wh*-question operatos, etc.) and ordinary, non-*wh*-NPs. The relativizer in (51) specifically looks for gapped sentences in which the gap NP has type NP_{+wh}, and restrictions on *aL* and *goN* also make reference to the existence of a free variable (or unwithdrawn hypothesis) corresponding to a *wh*-gap, not just any hypothesis in hypothetical reasoning.

With the lexicon in (51), the derivation for the topmost relative clause in (52) goes as in (53).

(52) the man **aL** [I said **aL** [I thought **aL** [__ would be there]]]

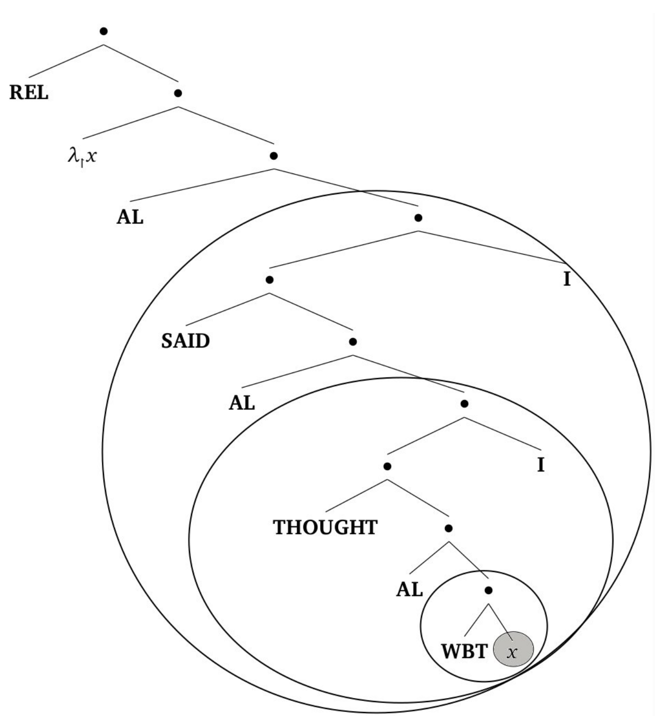
(53) $\mathbf{REL}_{(\mathrm{N}\backslash\mathrm{N})\upharpoonright(\mathrm{S}'\upharpoonright\mathrm{NP}_{+wh})}$

$$(\lambda_{\upharpoonright}x.\mathbf{AL}_{\mathrm{S}'/\mathrm{S}}(\mathbf{SAID}_{(\mathrm{NP}\backslash\mathrm{S})/\mathrm{S}'}(\mathbf{AL}_{\mathrm{S}'/\mathrm{S}}(\mathbf{THOUGHT}_{(\mathrm{NP}\backslash\mathrm{S})/\mathrm{S}'}(\mathbf{AL}_{\mathrm{S}'/\mathrm{S}}(\mathbf{WBT}_{\mathrm{NP}\backslash\mathrm{S}}(x_{\mathrm{NP}_{+wh}})))(\mathbf{I}_{\mathrm{NP}})))(\mathbf{I}_{\mathrm{NP}})))$$

Here, each token of *aL* applies to a clausal complement containing a free NP_{+wh} variable, and hence is legal.

This is perhaps clearer to see in the tree format than in the linear, logical-formula format in (53). As in the following tree, for each embedded clause, x remains free (i.e., not bound by the λ-operator) in the expression that is the argument of *aL* (the three circled subterms).

(54)

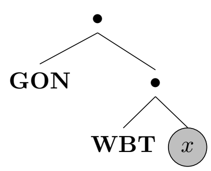


The ungrammaticality of the examples in (55) also follows immediately. In the case of (55a), *goN* is used instead of *aL* in the subproof corresponding to the innermost clause. This violates the constraint $fv_{X_{+wh}}(\alpha) = \emptyset$ on the first argument of *goN*. Similarly, in (55b), *goN* replaces the first *aL* in the subproof corresponding to the innermost clause. Here again, the relevant 'no unbound $+wh$ hypothesis' constraint on *goN* is violated.

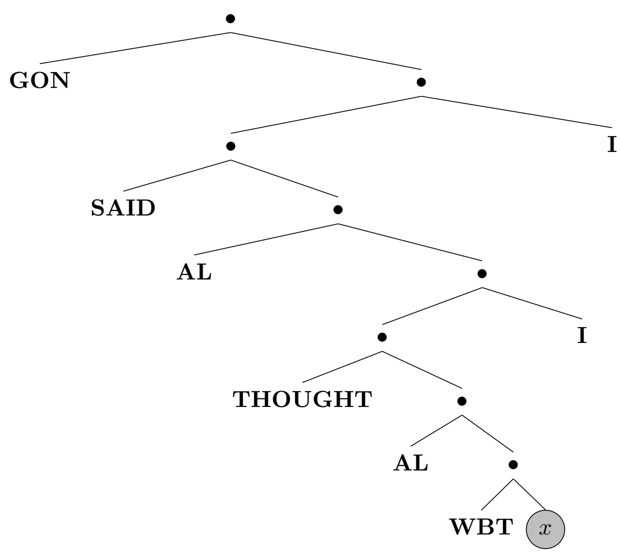
(55) a. *the man **aL** [I said **aL** [I thought **goN** [__ would be there]]]
b. *the man **goN** [I said **aL** [I thought **aL** [__ would be there]]]

The offending subterms in the proofs for (55a,b) are shown in (56).

(56) a.



b.



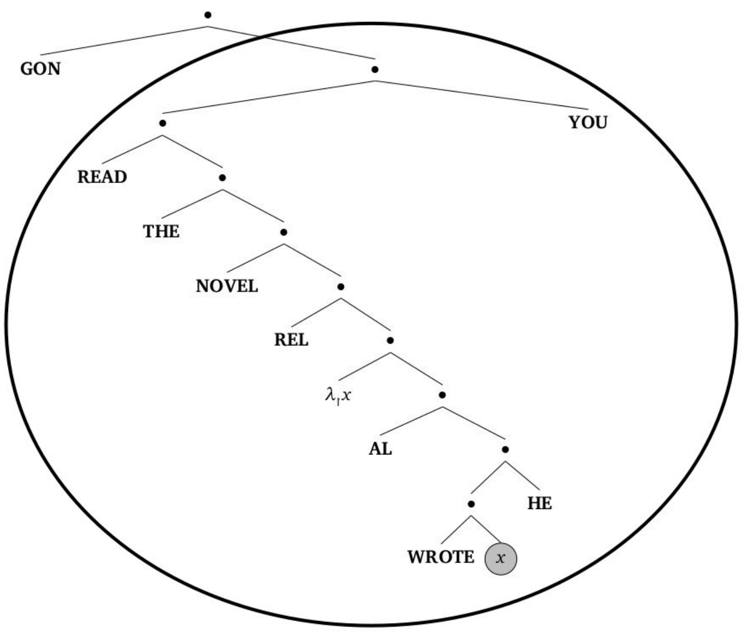
A further prediction of this approach is that when extraction terminates in an embedded clause, the complementizer in a higher structure will be *goN*, rather than *aL*. Such a case is attested in Irish with the following example (reproduced in Iringlish) involving a resumptive pronoun in the most embedded clause (originally (32) in McCloskey (1979, 164)):

(57) Which$_i$ poet **aL** [did $\_\__i$ you say **goN** [you read the novel$_i$ **aL** [he wrote $\_\__j$]]]?

The embedded *aL* is licensed by the filler-gap linkage between the object position gap in the lowest clause and the relativization operator. The intermediate clause *you read the novel he wrote* __ doesn't contain any unbound gap, so it is marked by *goN*. Finally, assuming that *which poet* is licensed as a 'correlate' of the embedded resumptive pronoun in the matrix clause and gets promoted to the sentence initial position by the *wh*-operator, the *aL*-marking in the matrix clause is explained by the same mechanism as in other *wh*-questions.

The following subterm illustrates the structure under *goN*:

(58)



The variable x is bound inside the circled subterm. Thus, the argument of *goN* doesn't contain any occurrence of a free variable.

Finally, we show that adjunct extraction cases are completely parallel to cases involving extraction of arguments. Adjunct extraction poses an interesting theoretical issue in lexicalist theories of syntax such as HPSG and CG (see, e.g., Hukari and Levine (1995)), since in such theories, there is an asymmetry between arguments and adjuncts in that the former is an argument of a lexical verb but the latter is a function that takes a verbal projection as an argument. Thus, if extraction is analyzed by a mechanism that targets the valence information of a lexical verb (as in certain versions of HPSG), then adjunct extraction necessitates a nontrivial theoretical move in which adjuncts are 'reanalyzed' as arguments of verbs (Bouma et al. (2001) in fact makes such a proposal in HPSG). Our analysis of extraction in TLG involves a general process of hypothetical reasoning, and no such complex reanalysis of the status of adjuncts is needed.

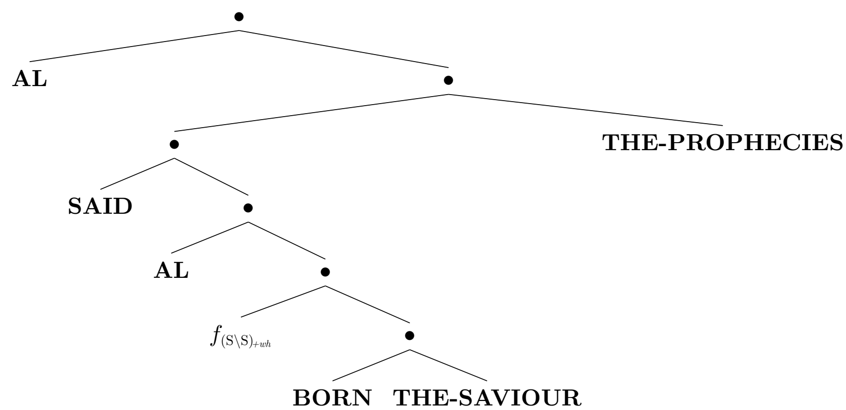
In an example such as (59) (= (50)), the extracted adjunct semantically modifies the embedded clause. Thus, a hypothetical clausal modifier of type $(\mathrm{S}\backslash\mathrm{S})_{+wh}$ is posited in the lower clause as in (60)/(61).[4]

[4] Admitting the syntactic type $(\mathrm{S}\backslash\mathrm{S})_{+wh}$ necessitates a move in the underlying theory in which not just atomic types but complex types can be specified for (at least certain) syntactic features as well. This may involve some major reworking of the feature system in TLG, but we leave this task for future work.

(59) It was in Bethlehem **aL** [the prophecies said **aL** [the Saviour would be born __]]

(60) $\mathbf{AL}_{\mathrm{S'/S}}(\mathbf{SAID}_{(\mathrm{NP\backslash S})/\mathrm{S'}}$
$(\mathbf{AL}_{\mathrm{S'/S}}(f_{(\mathrm{S\backslash S})_{+wh}}(\mathbf{BORN}_{\mathrm{NP\backslash S}}(\mathbf{THE\text{-}SAVIOUR}_{\mathrm{NP}}))))(\mathbf{THE\text{-}PROPHECIES}_{\mathrm{NP}}))$

(61)



Here again, until the variable f (of type $(\mathrm{S\backslash S})_{+wh}$) is bound, the right form of the complementizer is *aL*, so it is correctly predicted that the two occurrences of *aL* in (59) cannot be replaced by *goN*.

3.3 Extraction pathway marking in other languages

Having provided an analysis of the basic patterns of extraction pathway marking, we now turn to the question of whether this analysis is fully general. Specifically, we critically examine the recent claim by van Urk and Richards (2015) and van Urk (2020) that *both* successively cyclic movement and feature percolation are needed to capture the entire patterns of extraction pathway marking. The crucial piece of evidence comes from the patterns displayed by Dinka. According to van Urk and Richards (2015), the apparent violation of the V2 word order in the language exceptionally observed at *wh*-extraction pathways provides evidence for actual movement of the *wh*-phrase. However, the 'long-distance' plural agreement cannot be accounted for by movement alone, and requires a feature checking (or feature percolation) mechanism of some sort. van Urk (2020) summarizes facts from a wider range of languages for each type of evidence.

To state the conclusion first, while we agree with these authors that these phenomena call for some mechanism in the grammar for keeping track of the identity of the gap before the filler-gap linkage is established, the relevant facts can be analyzed adequately by what we have already proposed, together with independently motivated properties of the specific morpho-syntactic phenomena that exhibit extraction pathway marking effects. Among the two types of alleged evidence for distinct mechanisms, the 'feature checking' evidence can be dealt with by a slight extension of the analysis of the Irish complementizer marking pattern. We briefly demonstrate this point in section 3.3.1. After that, we turn to the main task in this section in sections 3.3.2 and 3.3.3, focusing on two types of 'movement evidence' reported in van Urk and Richards (2015) and van Urk (2020), specifically, Dinka word order and Ulster English floating quantifier *all*.

3.3.1 A brief note on 'agreement' type extraction pathway marking effects

Cases of extraction pathway marking in which the marking is sensitive to some particular syntactic or semantic feature of the extracted expression, such as the plural marking morphology in Dinka reported in van Urk and Richards (2015), perhaps requires some discussion, before we tackle the main issue of the movement-type evidence for extraction pathway marking. Here, we show that such cases can be analyzed essentially by the same approach we proposed for Irish complementizer marking, together with the feature-based account of agreement standardly assumed in lexicalist syntax (including TLG).

For the purpose of illustration, suppose that Iringlish had morphological indication of the plurality of the extracted item realized as reduplicative morphology in the form of an intermediate verb . Agreement is handled via features encoded in syntactic categories in lexicalist theories of syntax. Using this feature-based analysis of agreement, a plural-gap variant of the verb *think* can be defined as follows:

(62) $\mathbf{THOUGHT\text{-}PL}_{\mathrm{VP/S'}} = \mathsf{thought\text{-}thought}; \mathbf{thought}; \mathrm{VP/S'}$
where for any α, $\mathbf{THOUGHT\text{-}PL}(\alpha)$ is defined only if $fv_{\mathrm{X}_{+wh}}(\alpha) \neq \emptyset$
and the singleton element of $fv_{\mathrm{X}_{+wh}}(\alpha)$ has type NP_{+pl}

(63) the $\left\{\begin{matrix}\text{a.*man} \\ \text{b.men}\end{matrix}\right\}$ **aL** [I thought-thought **aL** [__ would be there]]

Since the gap NP and the head noun are required to agree in number by the relativization operator, in (63a) the gap NP has type NP_{+pl} and in (63b) it has type NP_{-pl}, yielding the subterms in (64a) and (64b), respectively, as arguments to (62). Only the latter satisfies the definedness condition for (62), correctly capturing the pattern in (63).

(64) a. $\mathbf{AL}_{\mathrm{S'/S}}(\mathbf{WBT}_{\mathrm{NP\backslash S}}(x_{\mathrm{NP}_{+pl}}))$
b. $\mathbf{AL}_{\mathrm{S'/S}}(\mathbf{WBT}_{\mathrm{NP\backslash S}}(x_{\mathrm{NP}_{-pl}}))$

3.3.2 V2 word order in Dinka

van Urk and Richards (2015) present the following pattern of extraction pathway marking reflected in V2 word order in Dinka as evidence for an actual movement of a copy of the *wh*-phrase in successive cyclicity. We reproduce the relevant pattern in Dinklish, another hypothetical dialect of English which mimicks Dinka syntax with an English lexicon.

First, (65) shows that embedded clauses exhibit the V2 word order, and that leaving the preverbal position empty is not allowed.

(65) a. Bill$_j$ thinks $\_\__j$ ke [Mary$_i$ bought $\_\__i$ the book].
'Bill thinks that Mary bought the book.'

b. *Bill$_j$ thinks $\_\_{}_j$ ke [__ bought Mary the book].

But there is a systematic exception to this V2 word order requirement. The preverbal position can, and in fact must, be empty when it is crossed by a *wh*-dependency chain. This is demonstrated by (66).

(66) a. Who thought John ke [said Mary ke [criticized Bill __]]?
'Who did John think Mary said Bill criticized __?'

b. *Who$_i$ thought John ke [**Mary**$_j$ said $\_\_{}_j$ ke [criticized Bill $\_\_{}_i$]]?

c. *Who$_i$ thought John ke [said ke [**Bill**$_j$ criticized $\_\_{}_j$ $\_\_{}_i$]]?

d. *Who$_i$ thought John ke [**Mary**$_k$ said $\_\_{}_k$ ke [**Bill**$_j$ criticized $\_\_{}_j$ $\_\_{}_i$]]?

(66a) is grammatical since the preverbal position in the most embedded and intermediate clauses are both left unoccupied. By contrast, in (66b–d), either the preverbal position in the lowest or the intermediate clause (or both) is occupied by an overt NP, and this leads to ungrammaticality.

van Urk and Richards (2015) characterize the preverbal position as Spec,CP. According to them, the pattern in (66) falls out immediately if Spec,CP is an intermediate landing site of the moved *wh*-phrase.

However, there is an alternative account of this distributional pattern that doesn't rely on actual movement of a *wh*-phrase (or an analogous mechanism in TLG), which identifies the semantic effect of extraction as one of the key components of the explanation. The key idea is that the preverbal position in Dinklish (or Dinka) corresponds to the 'variable' slot in the abstract predicate-argument structure underlying the topic/comment structure in ordinary sentences and the focus/background structure in *wh*-questions. Specifically, we make the following assumptions:

(67) a. Every clause must be associated with at most one 'most prominent' element.

b. The preverbal position is the designated position for the prominent element, and is licensed through $\upharpoonleft$.

c. As a consequence of (67a,b) when $\upharpoonleft$-Introduction applies to produces a predicate-argument structure underlying V2 syntax, there has to be exactly one unwithdrawn hypothesis (corresponding to the element carrying prominence).

To see how this works, consider first the following simple 'Dinklish' sentence with local topicalization:

(68) Bill$_i$ gave $\_\_{}_i$ Mary the book.

(69)

$$
\dfrac{
\dfrac{
\begin{array}{l}\lambda\varphi_1\lambda\sigma.\\ \quad \varphi_1 \bullet \sigma(\boldsymbol{\epsilon});\\ \lambda z\lambda R.R(z);\\ \mathrm{S}\upharpoonleft(\mathrm{S}\upharpoonleft\mathrm{NP})\upharpoonleft\mathrm{NP}\end{array}
\quad
\begin{array}{l}\mathsf{bill};\\ \mathbf{b};\\ \mathrm{NP}\end{array}
}{
\begin{array}{l}\lambda\sigma.\mathsf{bill} \bullet \sigma(\boldsymbol{\epsilon});\\ \lambda R.R(\mathbf{b});\\ \mathrm{S}\upharpoonleft\mathrm{NP}\end{array}
}\upharpoonleft\mathrm{E}
\quad
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\begin{array}{l}\mathsf{gave};\\ \lambda y\lambda x\lambda w.\\ \quad \mathbf{give}(x)(w)(y);\\ \mathrm{S/NP/NP/NP}\end{array}
\quad
\begin{bmatrix}\varphi_1;\\ v;\\ \mathrm{NP}\end{bmatrix}^1
}{
\begin{array}{l}\mathsf{gave} \bullet \varphi_1;\\ \lambda x\lambda w.\mathbf{give}(x)(w)(v);\mathrm{S/NP/NP}\end{array}
}/\mathrm{E}
\quad
\begin{array}{l}\mathsf{mary};\\ \mathbf{m};\\ \mathrm{NP}\end{array}
}{
\begin{array}{l}\mathsf{gave} \bullet \varphi_1 \bullet \mathsf{mary};\\ \lambda x.\mathbf{give}(\mathbf{m})(w)(v);\mathrm{S/NP}\end{array}
}/\mathrm{E}
\quad
\begin{array}{l}\vdots\\ \mathsf{the} \bullet \mathsf{book};\\ \iota(\mathbf{book});\\ \mathrm{NP}\end{array}
}{
\begin{array}{l}\mathsf{gave} \bullet \varphi_1 \bullet \mathsf{mary} \bullet \mathsf{the} \bullet \mathsf{book};\\ \mathbf{give}(\mathbf{m})(\iota(\mathbf{book}))(v);\mathrm{S}\end{array}
}/\mathrm{E}
}{
\begin{array}{l}\lambda\varphi_1.\mathsf{gave} \bullet \varphi \bullet \mathsf{mary} \bullet \mathsf{the} \bullet \mathsf{book};\\ \lambda v.\mathbf{give}(\mathbf{m})(\iota(\mathbf{book}))(v);\mathrm{S}\upharpoonleft\mathrm{NP}\end{array}
}\upharpoonleft\mathrm{I}^1
}{
\mathsf{bill} \bullet \mathsf{gave} \bullet \varphi \bullet \mathsf{mary} \bullet \mathsf{the} \bullet \mathsf{book};\ \mathbf{give}(\mathbf{m})(\iota(\mathbf{book}))(\mathbf{b});\ \mathrm{S}
}\upharpoonleft\mathrm{E}
$$

(70) $\mathbf{TOP}_{\mathrm{S}\upharpoonleft(\mathrm{S}\upharpoonleft\mathrm{NP})\upharpoonleft\mathrm{NP}}(\lambda_{\upharpoonleft}x.\mathbf{GAVE}_{\mathrm{S/NP/NP/NP}}(x_{\mathrm{NP}})(\mathbf{MARY}_{\mathrm{NP}})(\mathbf{THE\text{-}BOOK}_{\mathrm{NP}}))(\mathbf{BILL}_{\mathrm{NP}})$

At the step $\upharpoonleft$-Introduction applies, there is exactly one free variable v (corresponding to the unwithdrawn hypothesis indexed 1), so, the derivation succeeds. Since this hypothesis corresponds to the subject argument of the verb *gave*, we get a subject topicalization sentence.

Consider next the following minimal pair, which shows that an embedded topic position cannot remain empty:

(71) a. *Bill thinks __ ke [__ bought Mary the book].
'Bill thinks that Mary bought the book.'
b. Bill thinks __ ke [Mary bought __ the book].
'Bill thinks that Mary bought the book.'

To account for this pattern (and also the *wh*-dependency patterns below), we assume that the complementizer *ke* has the role of ensuring the condition (67a) above, which can be made explicit as in (72).

(72) *Ke* imposes the restriction that there is exactly one free variable in its complement.

As we show immediately below, in the normal topicalization example, after *ke* checks the existence of a free variable, the variable gets bound by $\upharpoonleft$-Introduction as usual, and the result is then fed to the topicalization operator; otherwise, that is, when there is a filler corresponding to an embedded gap in a higher clause, *ke* simply passes the free variable upstairs.

For (71), what goes wrong in (71a) is that at the point *ke* combines with the embedded clause, both of the argument positions are occupied by full NPs as in (73a). This violates the condition on *ke* in (72), hence the derivation fails. By contrast, in the case of the topicalization example (71b), the underlined subproof in (73b) satisfies (72), with the free variable x_{NP} which then gets bound by the topicalization operator that licenses the overt NP *Mary* in the clause initial position.

(73) a. $\mathbf{KE}_{\mathrm{S'/S}}(\mathbf{BOUGHT}_{\mathrm{S/NP/NP}}(\mathbf{THE\text{-}BOOK}_{\mathrm{NP}})(\mathbf{MARY}_{\mathrm{NP}}))$

b. $\mathbf{TOP}_{\mathrm{S}\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP})\upharpoonright\mathrm{NP}}(\lambda_{\upharpoonright}x.\mathbf{KE}_{\mathrm{S}'/\mathrm{S}}(\underline{\mathbf{BOUGHT}_{\mathrm{S/NP/NP}}(\mathbf{THE\text{-}BOOK}_{\mathrm{NP}})(x_{\mathrm{NP}})}))(\mathbf{MARY}_{\mathrm{NP}})$

Assuming that the same constraint is operative in more complex sentences involving long-distance extraction of a *wh*-phrase, the pattern in (66) falls out from the assumptions already made. As noted above, all the preverbal positions in intermediate clauses crossed by filler-gap linkage have to be empty:

(74) a. I wonder who thought John ke [said Mary ke [criticized Bill __]].
'I wonder who John thought Mary said Bill criticized __.'
b. *I wonder who$_i$ thought John ke [Mary$_j$ said __$_j$ ke [criticized Bill __$_i$]].
'I wonder who John thought Mary said Bill criticized __.'

We start with the analysis of the grammatical example (74a). Note first that the subproof for the most deeply embedded clause satisfies both (67) and (72), since it contains exactly one hypothesis x_{NP}.

(75) $\mathbf{KE}_{\mathrm{S}'/\mathrm{S}}(\underline{\mathbf{CRITICIZED}_{\mathrm{S/NP/NP}}(x_{\mathrm{NP}})(\mathbf{BILL}_{\mathrm{NP}})})$

The same process is repeated in the upstairs clause, yielding (76), again satisfying the relevant conditions at the intermediate clause headed by *said*:

(76) $\mathbf{KE}_{\mathrm{S}'/\mathrm{S}}(\mathbf{SAID}_{\mathrm{S/NP/S}'}(\mathbf{KE}_{\mathrm{S}'/\mathrm{S}}(\mathbf{CRITICIZED}_{\mathrm{S/NP/NP}}(x_{\mathrm{NP}})(\mathbf{BILL}_{\mathrm{NP}})))(\mathbf{MARY}_{\mathrm{NP}}))$

Finally, at the matrix level, the hypothesis is withdrawn to yield S↾NP, which is then given as an argument to the *wh*-operator:

(77) $\mathbf{WHO}_{\mathrm{Q}\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP})}(\lambda_{\upharpoonright}x.\mathbf{THOUGHT}_{\mathrm{S/S'/NP}}(\mathbf{JOHN}_{\mathrm{NP}})$
$(\mathbf{KE}_{\mathrm{S}'/\mathrm{S}}(\mathbf{SAID}_{\mathrm{S/NP/S}'}(\mathbf{KE}_{\mathrm{S}'/\mathrm{S}}(\mathbf{CRITICIZED}_{\mathrm{S/NP/NP}}(x_{\mathrm{NP}})(\mathbf{BILL}_{\mathrm{NP}})))(\mathbf{MARY}_{\mathrm{NP}}))))$

Turning now to the ungrammatical (74b), the offending structure is the subproof for the intermediate clause headed by *said*, where the preverbal position is occupied by the local subject *Mary* of that clause, instead of being left empty. As in the above example (71) (with derivation in (73b)) in order to license an overt NP in the topic position, we need to do hypothetical reasoning as in (78). But the underlined part violates the condition on *ke* in (72), since this subproof has two variables x_{NP} (corresponding to the *wh*-filler) and y_{NP} (for the local topic).

(78) $\mathbf{TOP}_{\mathrm{S}\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP})\upharpoonright\mathrm{NP}}(\lambda_{\upharpoonright}y.\mathbf{KE}_{\mathrm{S}'/\mathrm{S}}$
$\underline{(\mathbf{SAID}_{\mathrm{S/NP/S}'}(\mathbf{KE}_{\mathrm{S}'/\mathrm{S}}(\mathbf{CRITICIZED}_{\mathrm{S/NP/NP}}(x_{\mathrm{NP}})(\mathbf{BILL}_{\mathrm{NP}})))(y_{\mathrm{NP}})})))$

To summarize, the Dinka V2 word order pattern exhibited in (66) (in Dinklish) can be explained by an interaction of the topicalization operator and *wh*-extraction. Essentially, the ungrammatical cases all violate the constraint that there has to be exactly one 'prominent' element in a clause. Since both topicalization and *wh*-extraction exploit hypothetical reasoning at the syntax-semantics interface to identify a particular expression in the syntax as the 'prominent' element with respect to the respective constructions (where 'prominent' corresponds to focus in *wh*-extraction and topic in topicalization), we predict the same pattern as van Urk and Richards (2015), without identifying the preverbal position as a particular type of syntactic projection targeted by cyclic movement (or its analog in TLG).

3.3.3 Ulster English *all*

McCloskey (2000) argues that the Ulster subdialect of Irish English allows the extracted operator *what all* to jettison the quantifier-like *all* at various point along a Spec-to-Spec series of local extraction steps, giving tangible evidence that the extracted *wh*-phrase has passed through those steps to arrive at its final landing site. His evidence for this analysis includes the set of data in (79)–(81).

(79) a. What **all** did you get __ for Christmas?
  b. Who **all** did you meet __ when you were in Derry?

(80) a. What did you get **all** __ for Christmas?
  b. Who did you meet **all** __ when you were in Derry?

(81) a. What **all** did he say (that) he wanted __?
  b. What did he say (that) he wanted __ **all**?
  c. What did he say **all** (that) he wanted __?

On McCloskey's reasoning, the semantic identity of the floating and non-floating variants of *what/who all* sentences in (79) vs. (80) justifies an analysis in which *what/who all* is 'underlyingly' a unit. On the other hand, as illustrated in (81), the apparently free-floating *all* appears at exactly the points in the sentence that correspond either to the *wh* element's site of origin (as in (81b)) or to an intermediate Spec,CP position on the extraction pathway (as in (81c)). Mccloskey then takes the distribution of *all* as (at least indirect) evidence for cyclic movement.[5]

In what follows, we sketch an alternative explanation of these facts. We take *all* to be syntactically a VP adverb which imposes a certain semantic restriction on a free variable in its argument. This latter semantic effect is what gives rise to the apparent synonymy between the floating and non-floating variants of *what/who ... all*. Here again, our alternative account crucially makes reference to the intermediate status of the proof, in such a way that the semantic interpretation of the free variable (unwithdrawn hypothesis) plays a key role.

One piece of evidence for the assumption that stranded *all* is an adverb comes from data such as the following:

[5] As noted by an anonymous reviewer, an influential idea in the mainstream syntax holds that extraction pathway marking effects are exhibited at phase edges only. Whether such an assumption can be maintained depends largely on theory-internal assumptions. For example, as we discuss briefly in the main text, for the floating *all* in Ulster English McCloskey makes certain otherwise unmotivated assumptions in order to maintain the idea that it is a Spec,CP remnant. While we acknowledge that this is be a potentially controversial point, our own view is that trying to control the distribution of extraction pathway marking effects by the notion of phase is not particularly successful, and that a functional explanation along lines we speculate in section 4 would be more promising.

(82) ?What did you put in the drawer __ **all** (yesterday)?

On the VP modifier analysis, the position of *all* in (82) is naturally expected. By contrast, on McCloskey's (2000) movement-based analysis, (82) has to be analyzed as first involving a local movement of *what all* to the post-PP position (which is prohibited for overt, non-*wh*-NPs). However, such an analysis seems highly implausible given the lack of any independent evidence for the supposed movement operation.[6]

For the sake of exposition, we start with the analysis of non-floating (83b) and then extend it to the floating *all* in (83a).

(83) a. Who did Frank tell you **all** that they were after __?
b. Who **all** did Frank tell you that they were after __?

For the non-stranded case, we posit the following entry for *all*:

(84) $\lambda\rho\lambda\sigma.\rho(\lambda\varphi.\varphi) \bullet \mathsf{all} \bullet \sigma(\epsilon)$; $\lambda\mathscr{F}\lambda P\lambda x_C.\mathscr{F}(P)(x)$; $(\mathrm{Q}\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP}))\upharpoonright(\mathrm{Q}\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP}))$
defined only if the domain set C for x is above the
contextually relevant standard for high precision

This may look somewhat complex, but all it does is impose a certain restriction on the interpretation of the semantic variable x bound by the *wh*-operator. The semantic restriction imposed on x dictates that it be chosen from a domain set (i.e., contextually determined set of individuals) C which counts as sufficiently 'precise' in the context in

[6]Further support for the VP adverb analysis of *all* comes from the distributional parallel between the non-remnant adverb *precisely* and the floating *all*. Note first that *precisely* appears to have a very similar distribution as *all*, occuring in both the post-*wh* position and the 'in-situ' position:

(i) a. What **precisely** do you want __?
b. What do you want __ **precisely**?

However, as McCloskey himself notes, a closer inspection makes it clear that *precisely* cannot plausibly be analyzed as a *wh*-remnant:

(ii) a. *What did he say yesterday **precisely** that he wanted? [on the same reading as (iib)]
b. What **precisely** did he say yesterday that he wanted?

If *precisely* were a *wh*-remnant on a par with *all*, then (iia) should have a reading equivalent to (iib), with *precisely* being stranded at an intermediate landing site. However, (iia) clearly lacks such a reading.

Yet despite this clear difference in the *wh*-remnant status, *precisely* and *all* share a remarkable similarity in terms of their distributional properties as VP-internal adverbs syntactically, as shown by the following examples:

(iii) a. *What did he say {**precisely/all**} to {him/his students} that he wanted to buy __?
b. ?What did he say to {him/his students} {**precisely/all**} that he wanted to buy __?

This distributional parallel between *precisely* and *all* indicates that the pre-complementizer distribution of *all* that McCloskey takes as sufficient evidence for the Spec,CP remnant status of *all* can be accounted for equally naturally by simply assuming that it is a VP adverb that obeys the same word-order restrictions as an unequivocally non-remnant *precisely*.

question. By applying (84) to the *wh*-question operator *who* in (85), we obtain (86), which then licenses the semantics (87) for (83b).

(85) $\lambda\sigma.\mathsf{who} \bullet \sigma(\epsilon)$; $\lambda P \lambda x.\mathbf{wh_{person}}(x)(P)$; $\mathrm{Q}{\upharpoonright}(\mathrm{S}{\upharpoonright}\mathrm{NP})$

(86) $\lambda\sigma.\mathsf{who} \bullet \mathsf{all} \bullet \sigma(\epsilon)$; $\lambda P \lambda x_C.\mathbf{wh_{person}}(x)(P)$; $\mathrm{Q}{\upharpoonright}(\mathrm{S}{\upharpoonright}\mathrm{NP})$
defined only if the domain set C for x is above the contextually relevant standard for high precision

(87) $\lambda x_C.\mathbf{wh_{person}}(x)(\mathbf{tell}(\mathbf{you})(\mathbf{after}(x)(\mathbf{they}))(\mathbf{frank}))$
defined only if the domain set C for x is above the contextually relevant standard for high precision

The idea here is that by manipulating the domain set in the direction of increasing precision, things that are normally ignored enter into the domain of entities that the question sentence interrogates about. For example, suppose that a police officer is inquiring a witness in an investigation of an issue in which a foregn spy John died after having lunch with a suspicious person. In this situation, *What **all** did John eat?* is likely a more appropriate (and less ambiguous) question than *What did John eat?*, and it calls for a higher degree of precision and completeness for a proper answer.

Moving on to the floating *all*, we take this *all* to be syntactically a VP adverb which is reordered to the position immediately preceding the complement clause. This can be dealt with by some kind of surface reordering rule governing adverbs (see, e.g., Kubota (2014) for one approach in TLG), and it is motivated by the parallel distribution between *all* and the 'non-*wh*-remnant' adverb *precisely* noted in footnote 6. We can then take the combinatoric structure underlying the matrix VP in (83a) to be something like the following, where x is the free variable corresponding to the embedded gap:

(88) $\mathbf{ALL}_{\mathrm{VP/VP}}(\mathbf{TELL}_{\mathrm{VP/S'/NP}}(\mathbf{YOU}_{\mathrm{NP}})$
$(\mathbf{THAT}_{\mathrm{S'/S}}(\mathbf{WERE}_{\mathrm{VP/VP}}(\mathbf{AFTER}_{\mathrm{VP/NP}}(x_{\mathrm{NP}}))(\mathbf{THEY}_{\mathrm{NP}}))))$

Floating *all* then has the semantics analogous to the non-floating *all* in (84), with the only difference being that in the case of the floating *all*, the semantic variable that it targets is still *unbound* in the term that it takes as its argument as a VP adverb:

(89) $\mathbf{ALL}_{\mathrm{VP/VP}} = \mathsf{all}$; $\lambda P.P$; VP/VP
where $\mathbf{ALL}_{\mathrm{VP/VP}}$ is defined only if all elements $x_C \in \mathit{fv}(P)$ are such that the domain set C for x is above the contextually relevant standard for high precision

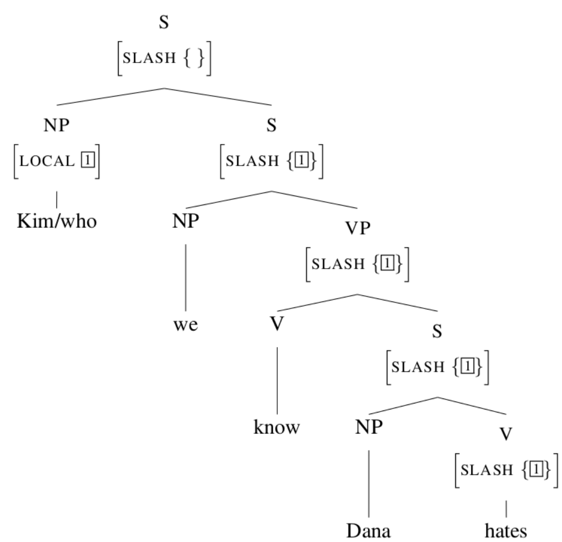
This imposes exactly the same restriction as the non-floating *all* on the variable x that the question operator ranges over. We thus obtain the same final translation for (83a) as for (83b), namely, (87). Thus, though the exact way in which *all* contributes its meaning in the compositional process is somewhat different in the two cases, we effectively get the same result as McCloskey (2000), preserving the key insight of his analysis that there is a tight connection between the *wh*-phrase and the stranded adverb *all*, but doing away with the undermotivated assumption that the latter is a movement remnant.

3.4 Comparison with a feature-percolation analysis of extraction pathway marking in HPSG

At this point, the key differences between our proof theoretic analysis and the successive cyclic analysis standard in derivational approaches should be clear. In the syntactic literature, an alternative analysis to the derivational analysis has been proposed by Bouma et al. (2001) in the constraint-based framework of HPSG that makes extensive use of the feature percolation mechanism of the framework. We briefly compare our approach with this HPSG approach in this section.

(90) illustrates the HPSG analysis of extraction.

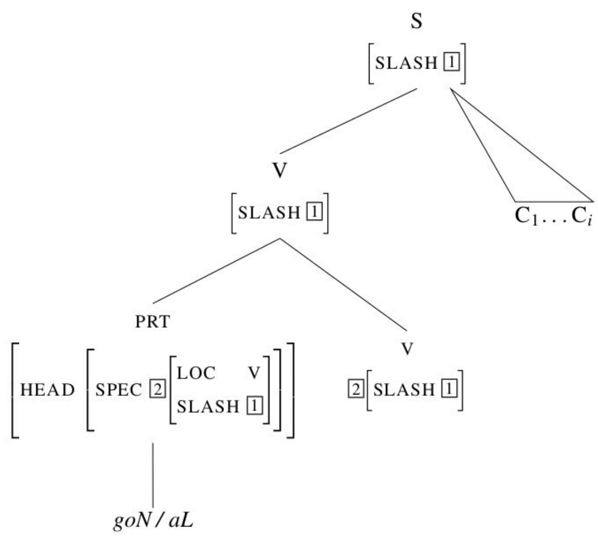
(90)



In HPSG, the SLASH feature is employed for indicating whether a phrase contains a gap position (in the object of the verb *hates* in (90)). As in (90), this information is successively inherited from daughter to mother via the feature percolation mechanism inherent to HPSG, until the corresponding filler is found (at the top S node).

Given this general analysis of extraction, in the case of complementizer marking in Irish, the choice of the morphological form of the complementizer can simply be made sensitive to the locally encoded value of the SLASH feature of the verbal projection that *goN/aL* directly combines with, since this feature indicates whether the clause in question contains a gap or not. This is schematically shown in (91).

(91)



The key difference, then, between the TLG analysis and this feature-percolation analysis in HPSG is the following. In the latter, the complementizer choice is dependent on the local syntactic information alone. This is in keeping with the locality condition in HPSG (see, e.g., Sag 2010) and it exploits the general SLASH inheritance mechanism that mediates nonlocal filler-gap linkage via a chain of local feature passing. By contrast, in our TLG analysis, the complementizer choice depends on the existence of an unwithdrawn hypothesis in the subproof (which may be deeply embedded). We have already noted in section 3.2 that this goes against the traditional CG dogma that proofs are not representational objects. The reader should now see a connection between HPSG and traditional CG: this CG dogma roughly corresponds to the locality condition in HPSG—indeed, they are likely to stem from ideas that shaped the common basic form of nonderivational syntactic theories in the 1980s.

While a casual cross-theoretic comparison can be misleading, there does seem to be a tradeoff about which part of the grammar needs to be made complex in the two approaches. Essentially, the HPSG approach abides by the locality principle by slightly enriching the local information encoded at each syntactic node. By contrast, the TLG approach does away with explicit feature percolation at the cost of violating the locality principle in a limited way—limited since all that is needed in this approach is identification of a free variable within a subterm (which conceptually corresponds to the 'tentative assumption' driving hypothetical reasoning in filler-gap linkage). Note that this doesn't involve complex manipulations ('transformations') of the structures of the subterms themselves, or anything that resembles the notion of 'phase' in minimalism (a proof-theoretic analog for this would be a set of meta-constraints imposing a certain 'structure' or ordering relations on proof strategy). In this sense, our proposal is structure-sensitive, but arguably *not* procedural, at least not in the same way that its derivational counterparts are.

As a final point of comparison with the constraint-based view of grammar embodied in HPSG, we would like to cautiously bring up possible implications for processing (we ourselves take the competence grammar and the theory of processing to be in principle distinct; see Kubota (2021, section 5) in this connection). One might initially think that processing-related considerations would favor the local licensing approach embodied in HPSG. However, note that the plausibility of this type of argument largely depends on the assumption that incremental parsing with complex data structures of the sort assumed in HPSG is cognitively realistic. By contrast, TLG embraces a much more indirect relationship between the grammar and processing. That being said, extraction pathway marking formalized as proof structure marking potentially illuminates a possible connection between grammar and processing that has largely been overlooked in the past literature. In proof-theoretic terms, establishing a filler-gap linkage corresponds to withdrawing a hypothesis at a certain point in a proof by finding a 'matching' premise (i.e., one that is looking to combine with a conditional statement derived from that hypothesis). Viewing syntactic parsing as proof search—which is a common perspective in TLG—such a complex proof strategy is very likely labor-intensive for the human online parser. It is then not too surprising that some natural languages have developed devices for explicitly flagging the intermediate statuses of the subproofs involved in such proofs, so as to efficiently narrow down the proof search space. Thus, on this view, extraction pathway marking can be naturally understood as a functionally motivated strategy, one that has fully developed into a grammatically encoded distinction in certain languages.

4 Conclusion

To conclude, extraction pathway marking is essentially a linguistic encoding of proof structure. We have arrived at this conclusion by making one tiny step into the domain of proof theory. This of course comes at the price of breaking the traditional dogma in CG research that proofs are not representational objects. In our view, this price is worth paying, for both technical and conceptual reasons. Technically, the step we have made is literally a tiny step, as it merely involves making reference to the notion of free variables in a typed lambda calculus, something that is already needed in formally interpreting lambda terms (for example, in the semantic domain). Conceptually, this gives us a cyclicity-free, feature-checking-free analysis of extraction pathway marking which moreover potentially illuminates the fundamental nature of the phenomenon: extraction pathway marking can now be seen as a functionally motivated grammatical encoding of (otherwise invisible long-distance) semantic linkage that is hard to identify.

A new, meaning-centered approach to the typology of extraction pathway marking then comes into sight, which can be contrasted with the structure-driven approach in traditional syntax (derivational and nonderivational alike). An almost immediate consequence of this approach is that extraction pathway marking makes reference to the *semantic* relationship between an unwithdrawn hypothesis (corresponding to a free variable) and a larger expression containing it. And there are a couple of 'obvious' choices for encoding such semantic sensitivity in specific morpho-syntactic devices, all attested in the literature:

- Irish (complementizer): Mark extraction pathways on the embedding markers directly. This is the most straightforward strategy, in which the argument of the function directly corresponds to an incomplete proof for a proposition-denoting constituent.

- Ulster English (*all* stranding), Dinka (plural marking): Impose a restriction pertaining to the semantic interpretation of the relevant free variable. Interestingly, this is syntactically realized as an adverb in Ulster English (the Dinka 'plural marker' may be analyzed similarly). VPs denote actions and properties, so, imposing a domain restriction on an embedded variable (whose semantic interpretation is indeterminate) via adverbial modification may facilitate the semantic processing of the VP denotation.

- Dinka (V2 word order): This is the most abstract and subtle type of encoding in which the 'prominent' status of the free variable (which eventually corresponds to the *wh*-filler) competes with some other item for the informationally distinguished status in the interpretation of an intermediate clause. Here again, the semantic interpretation of the variable within the subexpression in which it occurs plays a crucial role in licensing the relevant intermediate proof.

One of the central ideas driving comparative syntax in the generative tradition is that the combinatoric system underlying syntax has unique properties characterizing human language. Successive cyclicity has been one major (and quite attractive) candidate for such a property. While we do not claim to have addressed all the arguments for this notion, we believe that the present paper has offered an interesting alternative interpretation of some of the key facts. In our TLG analysis, extraction pathway marking reduces to nothing more than a fairly straightforward manifestation—in the overt syntax of natural language—of a well-understood property of logic. Our conclusion (and contention), then, is simple. TLG calls into question a widely entertained accepted wisdom by offering an alternative perspective: what looked like a *unique* property of the system of human language turns out to be a mere reflection of a *general* property of logic underlying that system.

References

Ajdukiewicz, Kazimierz. 1935. Die syntaktische Konnexität. In S. McCall, ed., *Polish Logic 1920-1939*, 207–231. Oxford: Oxford University Press. Translated from *Studia Philosophica*, Vol. 1: 1–27.

Arnold, Doug and Danièle Godard. 2021. Relative clauses in hpsg. In A. Abeillé, R. D. Borsley, J.-P. Koenig, and S. Müller, eds., *Head-Driven Phrase Structure Grammar: The Handbook*, 595–663. Language Science Press.

Asudeh, Ash and Richard Crouch. 2002. Derivational parallelism and ellipsis parallelism. In L. Mikkelsen and C. Potts, eds., *WCCFL 21 Proceedings*, 1–14. Somerville, MA: Cascadilla.

Bar-Hillel, Yehoshua. 1953. A quasi-arithmetic notation for syntactic descriptions. *Language* 29:47–58.

Borsley, Robert. 2010. An HPSG approach to Welsh unbounded dependencies. In S. Müller, ed., *Proceedings of the 17th International Conference on Head-Driven Phrase Structure Grammar*, 80–100. Stanford: CSLI.

Bouma, Gosse, Rob Malouf, and Ivan A. Sag. 2001. Satisfying constraints on extraction and adjunction. *Natural Language and Linguistic Theory* 19(1):1–65.

Buszkowski, Wojciech. 1987. The logic of types. In J. Srzednicki, ed., *Initiatives in Logic*, 180–206. Dordrecht: Springer Netherlands.

Carpenter, Bob. 1997. *Type-Logical Semantics*. Cambridge, MA: MIT Press.

Chaves, Rui P. and Michael T. Putnam. 2020. *Unbounded Dependency Constructions:Theoretical and Experimental Perspectives*. Oxford: Oxford University Press.

Chomsky, Noam. 1973. Conditions on transformations. In S. A. Anderson and P. Kiparsky, eds., *A Festschrift for Morris Halle*, 232–286. New York: Holt, Rinehart and Winston.

Chomsky, Noam. 1977. On *wh*-movement. In P. Culicover, T. Wasow, and A. Akmajian, eds., *Formal Syntax*, 71. New York: Academic Press.

Chomsky, Noan and Howard Lasnik. 1977. Filters and control. *Linguistic Inquiry* 8:425–504.

Chung, Sandra. 1982. Unbounded dependencies in chamorro grammar. *Linguistic Inquiry* 13:39–77.

Cooper, Robin. 1983. *Quantification and Syntactic Theory*, vol. 21 of *Synthese Language Library*. Dordrecht: Reidel.

Crouch, Dick and Josef van Genabith. 2000. Linear logic for linguists. Unpublished manuscript.

de Groote, Philippe. 2001. Towards abstract categorial grammars. In *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter*, 148–155. Association for Computational Linguistics.

Deane, Paul D. 1992. *Grammar in Mind and Brain*. Berlin: de Gruyter.

Dowty, David R., Robert Wall, and Stanley Peters. 1981. *Introduction to Montague Semantics*. Dordrecht: Reidel Publishing Co.

Fox, Danny. 2000. *Economy and Semantic Interpretation*. Cambridge, MA: MIT Press.

Hofmeister, Philip and Ivan A. Sag. 2010. Cognitive constraints and island effects. *Language* 86(2):366–415.

Howard, William A. 1969. The formulae-as-types notion of construction. In Jonathan P. Seldin and J. Roger Hindley (eds.) 1980, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*, 479–490 (New York: Academic Press).

Hukari, Thomas E. and Robert D. Levine. 1995. Adjunct extraction. *Journal of Linguistics* 31:195–226.

Kayne, Richard and Jean-Yves Pollock. 1978. Stylistic inversion, successive cyclicity and Move NP in French. *Linguistic Inquiry* 9:595–621.

Kluender, Robert. 1992. Deriving island constraints from principles of predication. In

H. Goodluck and M. Rochemont, eds., *Island Constraints: Theory, Acquisition, and Processing*, 223–258. Dordrecht: Kluwer.

Kluender, Robert. 1998. On the distinction between strong and weak islands: A processing perspective. In P. Culicover and L. McNally, eds., *The Limits of Syntax*, vol. 29 of *Syntax and Semantics*, 241–279. San Diego: Academic Press.

Kubota, Yusuke. 2014. The logic of complex predicates: A deductive synthesis of 'argument sharing' and 'verb raising'. *Natural Language and Linguistic Theory* 32(4):1145–1204.

Kubota, Yusuke. 2021. HPSG and categorial grammar. In A. Abeillé, R. D. Borsley, J.-P. Koenig, and S. Müller, eds., *Head-Driven Phrase Structure Grammar: The Handbook*, 1331–1394. Language Science Press. Available at https://zenodo.org/record/5599876/files/259-M

Kubota, Yusuke and Robert Levine. 2020. *Type-Logical Syntax*. Cambridge, MA: MIT Press.

McCloskey, James. 1979. *Transformational Syntax and Model-Theoretic Semantics*. Dordrecht: Reidel.

McCloskey, James. 2000. Quantifier float and *wh*-movement in an irish English. *Linguistic Inquiry* 31:57–84.

Mihaliček, Vedrana and Carl Pollard. 2012. Distinguishing phenogrammar from tectogrammar simplifies the analysis of interrogatives. In P. de Groote and M.-J. Nederhof, eds., *Formal Grammar 2010/2011*, 130–145. Heidelberg: Springer.

Moortgat, Michael. 2011. Categorial type logics. In J. van Benthem and A. ter Meulen, eds., *Handbook of Logic and Language*, 95–179. Amsterdam: Elsevier, 2nd edn.

Moortgat, Michael. 2014. Typelogical grammar. In E. N. Zalta, ed., *The Stanford Encyclopedia of Philosophy (Spring 2014 Ed.)*. Stanford University. Available at https://plato.stanford.edu/archives/spr2014/entries/typelogical-grammar/.

Morrill, Glyn. 1994. *Type Logical Grammar: Categorial Logic of Signs*. Dordrecht: Kluwer.

Morrill, Glyn. 2017. Grammar logicised: Relativisation. *Linguistics and Philosophy* 40(2):119–163.

Muskens, Reinhard. 2003. Language, lambdas, and logic. In G.-J. Kruijff and R. Oehrle, eds., *Resource Sensitivity in Binding and Anaphora*, 23–54. Dordrecht: Kluwer.

Newmeyer, Frederick J. 2016. Nonsyntactic explanations of island constraints. *Annual Review of Linguistics* 2:187–210.

Oehrle, Richard T. 1994. Term-labeled categorial type systems. *Linguistics and Philosophy* 17(6):633–678.

Pullum, Geoffrey K. 1992. CLS28: Papers from the 28th regional meeting of the Chicago Linguistic Society. In J. M. Denton, G. P. Chan, and C. Canakis, eds., *The origins of the cyclic principle*, vol. 2, 209–236. Chicago Linguistic Society.

Ross, John Robert. 1967. *Constraints on Variables in Syntax*. Ph.D. thesis, MIT. Reproduced by the Indiana University Linguistics Club.

Sag, Ivan A. 2010. Feature geometry and predictions of locality. In A. Kibort and

G. G. Corbett, eds., *Features: Perspectives on a Key Notion in Linguistics*, 236–271. Oxford: Oxford University Press.

Steedman, Mark. 2000. *The Syntactic Process*. Cambridge, MA: MIT Press.

van Urk, Coppe. 2020. Successive cyclicity and the syntax of long-distance dependencies. *Annual Review of Linguistics* 6:111–130.

van Urk, Coppe and Norvin Richards. 2015. Two components of long-distance extraction: Successive cyclicity in dinka. *Linguistic Inquiry* 46:113–155.

von Plato, Jan. 2018. The development of proof theory. In E. N. Zalta, ed., *The Stanford Encyclopedia of Philosophy (Winter 2018 Edition)*. https://plato.stanford.edu/archives/win2018/entries/proof-theory-development/.

Wansing, Heinrich. 1992. Formulas-as-types for a hierarchy of sublogics of intuitionistic propositional logic. In D. Pearce and H. Wansing, eds., *Nonclassical Logics and Information Processing*, 125–145. Berlin and Heidelberg: Springer.

Zaenen, Annie. 1983. On syntactic binding. *Linguistic Inquiry* 14(3):469–504.