Abstract

Early work on quantification in natural languages showed that sentences like *Every ape picked different berries*, on the reading that the sets of berries picked by any two apes are not the same, can be logically represented with a single polyadic quantifier for the two nominal phrases. However, since that quantifier cannot be decomposed into two quantifiers for the two nominal phrases, a compositional semantic analysis of this reading is not possible under standard assumptions about syntax and semantics. This paper shows how a constraint-based semantics with Lexical Resource Semantics can define a systematic syntax-semantics interface which captures the reading in question with a polyadic quantifier.

1 Introduction

One of an impressive series of fundamental contributions by Richard Montague to semantic theory was the consistent semantic treatment of nominal phrases as quantifiers, sets of sets of objects.[1] Although this proposal has been challenged and might not be the most adequate solution even for some of the cases it was originally designed for, an initial analysis of nominal phrases as (generalized) quantifiers (Barwise & Cooper, 1981) or a close variant thereof is still a fruitful and sound methodological strategy. It is this perception that forms the nucleus of the following proposal of treating a well-known but particularly challenging reading of nominal phrases with the adjective *different* as in *different berries* in terms of polyadic generalized quantifiers. *Polyadic* quantification is the necessary slight deviation from 'ordinary' monadic quantification in the present proposal that allows me to stay very close to the original spirit. The deviation is necessary because the relevant reading has been shown not to be amenable to the classic Montagovian treatment. Polyadic quantification also seems like the most conservative conceivable semantic modification available to provide an analysis of the data.

In fact, the proposed interpretation of *sentence internal* readings with *different* is not new at all. With certain modifications, benefiting from insights of subsequent literature on *different*, it is lifted straight from (Keenan, 1992). What is entirely new and the main topic of this paper is its full integration in an explicit syntax-semantics interface of a phrase structure grammar without an extra layer of elaborate LF syntax. Doing this has been impossible before in other frameworks of semantic composition for reasons to be explicated below.

I will be concerned with one particular reading of (1), namely (2-a).

[†]I thank three plus two anonymous reviewers and David Lahm for insightful comments, and audiences at Heinrich-Heine-Universität Düsseldorf, Goethe Universität Frankfurt a.M. and HeadLex16 in Warsaw for discussions of ideas presented in this paper. Janina Radó helped with proofreading.

[1]Intensionality will be ignored. Two-sorted Type Theory (Ty2, Gallin (1975)) with a type for worlds will be assumed for compatibility with other work in Lexical Resource Semantics, and to make clear that including possible worlds is desirable.

(1) $[_{\text{S}}\ [_{\text{NP}}$Every ape$]\ [_{\text{VP}}$picked $[_{\text{NP}}$different berries$]]]$

(2) a. The berries that any one of the apes picked were different from the berries each other ape picked.
b. Every ape picked berries that were different from the ones mentioned before.
c. Every ape picked various/many berries.

Following the broad characterization by Brasoveanu (2011) (and ignoring further possible distinctions), I call (2-a) the *sentence internal reading* and (2-b) an *external reading*; (2-c) is a third reading in which the *different* phrase shows no apparent contextual dependency similar to the other two readings.

In light of results proving that reading (2-a) cannot be obtained from two independent nominal phrases as in the syntactic analysis in (1), I develop a constraint-based syntax-semantics interface in Lexical Resource Semantics (LRS) that produces the reading in question. Instead of viewing the two nominal phrases as two monadic generalized quantifiers, they are semantically treated as a single categorematic polyadic quantifier which is unreducible to individual monadic quantifiers. At the same time the HPSG syntax with two unrelated nominal phrases is left intact, and lexical underspecification makes lexical redundancy unnecessary.

The crucial insight is that a constraint-based semantics can give a systematic and purely semantic account of the internal reading, unlike other sufficiently precise theories of a syntax-semantics interface. For reasons of space, the present discussion will mostly be confined to an explication of the semantic constraint system and the semantic lexical specification of the adjective *different* and interacting determiners that constitute the syntax-semantics interface. Broader empirical considerations and an extension of the proposal beyond a small set of core data have to await a later occasion.

2 *Different* with Polyadic Quantification

Figure 1 summarizes essential terminology and notation for the following exposition.

The nominal phrase *two apes* is seen as a monadic quantifier, Lindström type $\langle 1 \rangle$, that takes a set, or unary relation (of type $\langle et \rangle$), as an argument and returns a truth value, t, depending on the state of the world. The binary quantifier *two apes, all berries* (Lindström type $\langle 2 \rangle$) takes a binary relation as argument and returns a truth value – true if *two apes, all berries* holds of the relation, and false otherwise. In principle, this schema continues for polyadic quantifiers from any number of nominal phrases, with the Lindström type reflecting the number of NPs and the corresponding functional type.

Instead of combining NPs into larger units, we can also combine determiners to obtain new quantifiers. While *two* is a monadic generalized quantifier, taking two sets and returning a truth value, *two, all* forms a polyadic generalized quantifier

Figure 1: Some quantifiers with their Lindström types and functional types

Quantifier	Lindström type	Functional type
(two apes)	$\langle 1\rangle$	$\langle\langle et\rangle\, t\rangle$
(two apes, all berries)	$\langle 2\rangle$	$\langle\langle e\,\langle et\rangle\rangle\, t\rangle$
(two apes, every girl, many berries)	$\langle 3\rangle$	$\langle\langle e\,\langle e\,\langle et\rangle\rangle\rangle\, t\rangle$
($\mathrm{NP}_1, \ldots, \mathrm{NP}_n$)	$\langle n\rangle$	$\ldots$
(two)	$\langle 1, 1\rangle$	$\langle\langle et\rangle\,\langle\langle et\rangle\, t\rangle\rangle$
(two, all)	$\langle 1^2, 2\rangle$ $= \langle 1, 1, 2\rangle$	$\langle\langle et\rangle\,\langle\langle et\rangle\,\langle\langle e\,\langle et\rangle\rangle\, t\rangle\rangle\rangle$
(two, every, many)	$\langle 1^3, 3\rangle$ $= \langle 1, 1, 1, 3\rangle$	$\langle\langle et\rangle\,\langle\langle et\rangle\,\langle\langle et\rangle\,\langle\langle e\,\langle e\,\langle et\rangle\rangle\rangle\, t\rangle\rangle\rangle\rangle$

that takes as arguments two sets (of type $\langle et\rangle$) and a binary relation, type $\langle e\,\langle et\rangle\rangle$, and returns a truth value. The entire construct is of Lindström type $= \langle 1, 1, 2\rangle$, or short $\langle 1^2, 2\rangle$, and the complete functional type can be read off the table. Again this construction can be continued, this time for any number of determiners.[2]

The semantic object language of the HPSG grammar covering (1) will be a language of Ty2 with categorematic polyadic and polymorphic quantifiers with the functional types shown in Figure 1. Determiners like *two* or *every* will receive a semantic specification that permits their realization in the semantic representation of a proposition either as a monadic quantifier as shown for *two* in Figure 1 (this is the 'normal' case), or as a component of a polyadic quantifier as shown for *two, all* or *two, every many* – this will be a special case that presupposes the existence of a triggering element in the syntactic neighborhood. In our discussion, *different* will be the only relevant triggering element, but there can of course be others, such as negative quantifiers in negative concord constructions. Semantically, *different* will be translated by a constant much like the determiners *every* or *few*, but it will come with special contextual restrictions.

2.1 Intended Semantics

(Keenan, 1992, p. 202–203) observed that sentences of the structural type (3), for our purposes essentially a paraphrase of (1) when only considering reading (2-a), can be given a plausible semantics that cannot be reduced to some combination of two Fregean (or type $\langle 1\rangle$) quantifiers as independent translations of the two NPs.

(3) Different apes picked different berries.

This means that Keenan's (1992) quantifiers *different apes, different berries* as well

[2]For an historical account of the relevant concepts, see Westerståhl (1989). Lindström's paper is Lindström (1966). Keenan & Westerståhl (1997) has more on polyadic quantification in natural languages.

as *every ape, different berries* put such a specific, fine-grained condition on the binary picking relation that no combination of monadic quantifiers can express it. In linguistic terms, the condition cannot be obtained by having one of the two quantifiers take scope over the other, which would be expressed technically by combining the two putative component quantifiers by the operation of *iteration*.[3]

Before I can define the semantics of quantifiers containing *different*, I need a notational convention for certain sets of elements in a relation. E is the set with the elements in the discourse, E^2 ($= E \times E$) is the binary Cartesian product relation over E.

Notational Convention: Given a set E and a binary relation R, $R \subseteq E^2$, for each $x \in E$, I write Rx for the set of objects x bears R to: $Rx = \{y | (x, y) \in R\}$.

For example assume a binary relation pick that signifies who picks what, and assume further that pick contains pairs of apes and berries. The set of berries ape a picks can now simply be notated as $\mathsf{pick}\, a = \{b | (a, b) \in \mathsf{pick}\}$.

Keenan & Westerståhl (1997) develop a semantics in terms of polyadic quantification for many non-monadic examples in (Keenan, 1992), including quantifiers with *different*. I adapt their formulation[4] to the present discussion, and symbolize *different* with Δ.

Definition 1: Semantics of a quantifier containing Δ
For $\mathcal{Q}$ a polyadic quantifier of type $\langle 1^2, 2\rangle$ containing Δ, $A, B \subseteq E$, $R \subseteq E^2$, and H a quantifier of type $\langle 1, 1\rangle$, the interpretation of $\mathcal{Q}$ is as follows:
$\mathcal{Q}(A, B, R) = 1$ iff there is an A', $A' \subseteq A$ such that

$H(A, A') = 1$, and
for all $x, y \in A'$: $\quad (x \neq y) \Rightarrow (B \cap Rx \neq B \cap Ry)$.

Applied to (1), $\mathcal{Q}$ symbolizes *every, different*, with A the set of apes, B the set of berries, R the picking relation, and H the universal quantifier (*every*). The sentence is true iff there is a subset A' of the set of apes such that A is also a subset of A' ($H(A, A')$), i.e. A' actually equals the set of apes A; and for any pair x, y of distinct apes in A', the set of berries that one of them, say x, picks does not equal the set of berries that the other one, y, picks.

Various elaborations of this semantics are conceivable, such as demanding that A contain more than one element, or that the intersection of Rx and Ry be empty (no overlap in the set of berries picked by distinct apes). I will not pursue the topic of alternative formulations but want to stress in closing that H is meant to also comprise quantifiers other than *every*, including numerals such as *two*, or the

[3]Two tests for checking type $\langle 2\rangle$ quantifiers for reducibility are presented in (Keenan, 1992) with a generalization to quantifiers of type $\langle n\rangle$ in (Dekker, 2003). (Iordăchioaia, 2010, p. 37–38) uses one of Keenan's (1992) tests to show very carefully why *Two boys in my class date different girls* is unreducible. This proof can easily be adapted to our semantics of my sentence (1).

[4]Based on Iordăchioaia's (2010, p. 27) variant of (Keenan & Westerståhl, 1997, p. 877).

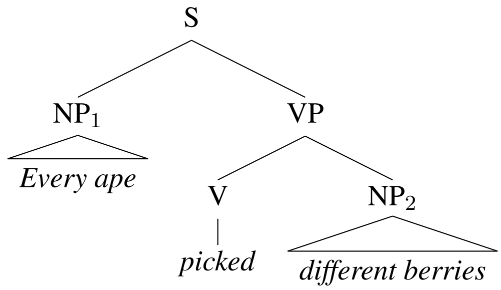
higher-order quantifiers *many* and *few*.[5]

Keenan & Westerståhl (1997) speculate that the polyadic quantifiers of natural languages are not arbitrary, they are built in certain regular ways from monadic quantifiers, and their semantics for quantifiers containing *different* is one of those regular patterns. If this is correct, a systematic syntax-semantics interface should be able to account for them, even if their makeup does not follow the typical pattern of iterated quantifiers that are usually in focus.

2.2 A Challenge for a Syntax-Semantics Interface

The non-reducibility results on *quantifiers containing* Δ mean that we are far from being done with a linguistic analysis of (1) if we adopt the semantic analysis above. In particular, it is impossible to interpret the two NPs *every ape* and *different berries* independently as generalized quantifiers and obtain that semantics. Even worse for syntactic theories like LFG and HPSG, a standard compositional semantic analysis using that semantics is impossible with the syntactic structure provided by the usual structural analysis of the sentence in these two frameworks, shown in the following tree:

(4)



In order to construct the intended polyadic quantifier in syntax, at the very least some additional syntactic movement would be necessary so that components of NP_1 and NP_2 form suitable syntactic units in the resulting representation. Neither HPSG nor LFG envisage this type of LF representation, as there is no genuinely syntactic evidence for its existence. Early versions of HPSG employed semantic representations expressed in feature logic that included a Cooper storage mechanism for storing and retrieving quantifiers in syntactic trees. Due to their limitations, quantifier storage mechanisms were superseded by theories that constrain the composition of logical contributions of syntactic constituents with *dominance constraints* or *subterm constraints* on semantic representations. I will show in the next section that one of these semantic composition theories, LRS, provides the necessary tools to keep the syntactic analysis above exactly as it is and still derive the semantic analysis of Keenan & Westerståhl (1997), and with it an analysis

[5](Keenan & Westerståhl, 1997, p. 877, fn. 18) contains a noteworthy remark that the monotonicity properties in the second argument of the quantifier H necessitates a refinement of the semantics they provide: as it stands, it does not yield the desired result when replacing *every* by *no*. They suggest an improved semantics which I avoid here for ease of exposition.

that is only a minimal variation of the idea that nominal phrases are interpreted as quantifiers.

3 An LRS Analysis

Three steps will be taken to prepare the syntax-semantics interface of LRS for an analysis of (1): First, the quantifiers in the logical object language receive a categorematic representation with sets (and relations) as arguments. Instead of representing *Every woman walks* as $\forall x(woman'_{\langle et\rangle}(x) \rightarrow walk'_{\langle et\rangle}(x))$, I use $\forall_{\langle\langle et\rangle\langle\langle et\rangle t\rangle\rangle}(\lambda x.woman'_{\langle et\rangle}(x), \lambda x.walk'_{\langle et\rangle}(x))$. $\forall_{\langle\langle et\rangle\langle\langle et\rangle t\rangle\rangle}$ is still a monadic quantifier, which leads to the second step: Polymorphic polyadic quantifiers in Ty2 are a generalization of these monadic quantifiers and will be presented in lexical entries that also illustrate how type polymorphism can be captured as an effect of lexical underspecification. The lexical specification of *different* is a special case that still fits the general pattern. In the third and final step, the LRS principles that govern the space of admissible semantic compositions must be generalized from monadic quantification to polyadic quantification. The essential architecture of LRS will not be reviewed here for reasons of space. For a compact and yet comprehensive introduction to all aspects of LRS relevant in the discussion below, the reader may want to consult (Iordăchioaia & Richter, 2015, pp. 626–632).

3.1 Revising the Representation of Quantifiers

Quantificational determiners in LRS introduce an appropriate constant for the corresponding logical quantifier ($\forall$ for *every*, most' for *most*), a variable that is bound in two lambda abstracts and used as a hook to nouns and verbs at the syntax-semantics interface, and a few restrictions on how these expressions of logical syntax enter into a larger quantificational expression, typically into the semantic representation of a sentence.

At the syntax-semantics interface, determiners indirectly identify syntactic valencies of a noun and a verb whose logical representations are functors of the variable the determiner introduces, and the determiners ensure (again indirectly) that the correct argument of the nominal and verbal predicates contain that variable.[6] Mostly as a matter of expository convenience when discussing essentially first-order examples, the formulae in the restrictor and scope of quantifiers have usually been represented as expressions of type t (Richter & Kallmeyer, 2009; Iordăchioaia & Richter, 2015). For better notational compatibility with generalized quantifier theory, it is preferable to switch the arguments to type $\langle et\rangle$, obtaining the usual monadic quantifiers of type $\langle\langle et\rangle\,\langle\langle et\rangle\,t\rangle\rangle$ (Lindström type $\langle 1,1\rangle$). For the semantic aspects of the lexical specification of a quantifier such as two', we obtain (5):

[6]For present purposes, I am only considering verb-argument structures.

(5)
$$\left[\begin{array}{ll}\text{PHON} & \langle \textit{two} \rangle \\ \text{SS LOC CONT} & \left[\begin{array}{ll}\text{INDEX DR} & \boxed{4a}\ x \\ \text{MAIN} & \boxed{4b}\ \mathsf{two}' \end{array}\right] \\ \text{LRS} & \left[\begin{array}{ll}\text{EXC} & \textit{me} \\ \text{INC} & \boxed{4}\ \mathsf{two}'(\lambda x.\alpha, \lambda x.\beta) \\ \text{PARTS} & \langle \boxed{4}, \boxed{4a}\ x, \boxed{4b}\ \mathsf{two}', \boxed{4c}(\lambda x.\alpha), \boxed{4d}\ (\lambda x.\beta), \boxed{4e}\ \mathsf{two}'(\lambda x.\alpha) \rangle \end{array}\right] \end{array}\right]$$
$$\&\ x \triangleleft \alpha\ \&\ x \triangleleft \beta$$

The determiner *two* introduces a fresh variable (x) as INDEX DR value. The MAIN value, i.e. the central lexical semantic contribution of the word, is the quantifier's constant, in this case two'. The IN(TERNAL-)C(CONTENT) contains the core expression that everything else in the syntactic projection of the determiner would outscope, and as such it is the most intuitive semantic representation of the word: two' applied to its two arguments, a lambda abstract $\lambda x.\alpha$, with as yet unknown α, and a second lambda abstract $\lambda x.\beta$, with as yet unknown β. The two subterm restrictions in the last line of (5) demand that x occur in α and in β. In simple constructions, α will ultimately contain the logical representation of the nominal projection that the determiner combines with, and β will contain the semantics of the verbal projection that the completed NP combines with (modulo operators with wider scope than the NP, which will not be subterms of β). The PARTS list enumerates all individual contributions to logical syntax by the determiner, including various function applications, e.g. $\mathsf{two}'(\lambda x.\alpha)$ (two' applied to the lambda abstract that is the quantifier's restrictor).

Nothing has to be changed in the usual syntactic analysis and LRS system of a grammar to derive nominal phrases such as *two apes*:

(6)
$$\left[\begin{array}{ll}\text{PHON} & \langle \textit{two, apes} \rangle \\ \text{SS LOC CONT} & \left[\begin{array}{ll}\text{INDEX DR} & \boxed{4a}\ x \\ \text{MAIN} & \boxed{5a}\ \mathsf{ape}' \end{array}\right] \\ \text{LRS} & \left[\begin{array}{ll}\text{EXC} & \boxed{4}\ \mathsf{two}'(\lambda x.\alpha, \lambda x.\beta) \\ \text{INC} & \boxed{5}\ \mathsf{ape}'(x) \\ \text{PARTS} & \left\langle \begin{array}{l}\boxed{4}, \boxed{4a}\ x, \boxed{4b}\ \mathsf{two}', \boxed{4c}\ (\lambda x.\alpha), \boxed{4d}\ (\lambda x.\beta), \\ \boxed{4e}\ \mathsf{two}'(\lambda x.\alpha), \boxed{5}, \boxed{5a}\mathsf{ape}' \end{array}\right\rangle \end{array}\right] \end{array}\right]$$
$$\&\ \boxed{5} \triangleleft \alpha\ \&\ x \triangleleft \alpha\ \&\ x \triangleleft \beta$$

The noun *apes* is the syntactic head of the noun phrase in (6) and selects the SYNSEM value of its determiner (5) by an appropriate valence attribute. The phrase *two apes* shares its INDEX DR and MAIN values with its head daughter, the noun *apes* (not depicted here individually). Similarly, the EX(TERNAL-)C(ONTENT) and the INC values are identified along the syntactic head path, while the PARTS list of the phrase contains all and only the elements of the PARTS lists of its two daughters.

The clause of the SEMANTICS PRINCIPLE that restricts the combination of determiners with nominal projections plays a crucial role.[7] It dictates that the INC

[7] A variant of that clause for polyadic quantification is spelled out below in (16).

value of the non-head daughter *two* ($\boxed{4}$) and the head daughter's EXC value be identical, hence the EXC of the phrase is also $\boxed{4}$, because EXC values are identical along syntactic head paths. Finally the INC of the head daughter, $\boxed{5}$, is a (possibly improper) subterm of the restrictor of the quantifier, expressed in (6) by $\boxed{5} \triangleleft \alpha$. Note that this makes sure that x occurs in α, as is lexically required by *two* in (5).

3.2 Lexical Specification of Polyadic Quantifiers

Keenan & Westerståhl's (1997) semantics for a 'polyadic quantifier containing *different*' can be reformulated straightforwardly for languages of Ty2 extended by a syntactic construct for this class of complex quantifiers. The new quantifiers are notated as $\mathcal{Q}$ below:

Definition 2: For H a quantifier of type $\langle 1, 1\rangle$ and $\mathcal{Q} = (H, \Delta)$ a polyadic quantifier of type $\langle 1^2, 2\rangle$ containing Δ, x, y variables of type e, α, β expressions of type $\langle et\rangle$, and ρ an expression of type $\langle e\,\langle et\rangle\rangle$, the interpretation of $\mathcal{Q}$ is as follows:
$[\![(H, \Delta)(\alpha, \beta, \rho)]\!]^{M,g} = 1$ iff there is an A', $A' \subseteq [\![\alpha]\!]^{M,g}$, such that

$[\![H(\alpha)]\!]^{M,g}(A') = 1$, and
$\forall \mathsf{e}_1, \mathsf{e}_2 \in A'$: $\mathsf{e}_1 \neq \mathsf{e}_2 \Rightarrow [\![\beta]\!]^{M,g} \cap [\![\rho]\!]^{M,g}(\mathsf{e}_1) \neq [\![\beta]\!]^{M,g} \cap [\![\rho]\!]^{M,g}(\mathsf{e}_2)$.

This definition presupposes a syntax where Δ is an appropriate logical constant for *different*, and for any type $\langle 1, 1\rangle$ quantifier H, (H, Δ) is a well-formed syntactic expression of the functional type $\langle\langle et\rangle\,\langle\langle et\rangle\,\langle\langle e\,\langle et\rangle\rangle\,t\rangle\rangle\rangle$ (see Figure 1).

Given such an extended syntax and semantics of Ty2, the necessary logical lexical specifications for the adjective *different* are simple:

$$
(7)\quad \begin{bmatrix}
\text{PHON} & \langle \textit{different} \rangle \\
\text{SS LOC} & \begin{bmatrix} \text{CAT HEAD MOD LOC CONT} & \begin{bmatrix} \text{INDEX DR } \boxed{1a}\, y \\ \text{MAIN } \zeta \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{INDEX DR } \boxed{1a}\, y \\ \text{MAIN } \boxed{1b}\, \Delta \end{bmatrix} \end{bmatrix} \\
\text{LRS} & \begin{bmatrix} \text{EXC} & \textit{me} \\ \text{INC} & \boxed{1}\,(\gamma,\Delta)(\sigma_1, \lambda y.\beta, \ldots \lambda y.\rho) \\ \text{PARTS} & \left\langle \begin{array}{l} \boxed{1}, \boxed{1a}\, y, \boxed{1b}\,\Delta, \boxed{1c}\,(\gamma,\Delta), \boxed{1d}(\lambda y.\beta), \boxed{1e}\,(\lambda y.\rho), \\ \boxed{1f}\,(\gamma,\Delta)(\sigma_1), \boxed{1g}\,(\gamma,\Delta)(\sigma_1,\lambda y.\beta) \end{array} \right\rangle \end{bmatrix}
\end{bmatrix}
$$

& $y \triangleleft \beta$ & $y \triangleleft \rho$ & $\zeta \triangleleft \beta$

The semantic specification is particularly perspicuous in the INC: *different* contributes a quantifier with Δ whose other component, γ, is lexically undetermined. It applies to two restrictor arguments, the first one lexically unknown (σ_1), whereas the second is a lambda abstract whose variable y is the DR value of *different* and shared with the DR value of the element it modifies (2nd line in (7)). The MAIN value, ζ, of a nominal projection that *different* modifies will end up in β of this restrictor, as required by the subterm statement $\zeta \triangleleft \beta$ in the last line of (7). The last argument of (γ, Δ), a binary relation, is largely undetermined except for the lambda abstraction over y, which is the second lambda abstraction in the expression, as indicated by the leading dots.

3.3 Principles of Semantic Composition

The final steps in capturing reading (2-a) of sentence (1) based on the syntactic analysis (4) concern the principles of semantic composition. In LRS all principles of the semantic combinatorics that are dependent on specific syntactic constellations are expressed as clauses of the SEMANTICS PRINCIPLE. We will have to reconsider adjective-noun combinations (for combining *different* with the nominal projection it modifies), and we have to be careful about determiner-noun combinations and NP-VP combinations, because these have to be generalized from monadic quantifiers, for which the relevant clauses of the SEMANTICS PRINCIPLE were originally defined, to the polyadic case. All of these will in fact be minor adaptations of existing clauses.

The first phrase to be examined is the noun phrase *different berries*, in which the adjective *different* forms a head-adjunct phrase with the count noun *berries*, which is the syntactic head of the phrase.

(8)
$$\begin{bmatrix} \text{PHON} & \left\langle \textit{different, berries} \right\rangle \\ \text{SS LOC CONT} & \begin{bmatrix} \text{INDEX DR} & \boxed{1a}\ y \\ \text{MAIN} & \boxed{2a}\ \mathsf{berry}' \end{bmatrix} \\ \text{LRS} & \begin{bmatrix} \text{EXC} & \boxed{1}\ (\gamma,\Delta)(\sigma_1, \lambda y.\beta, \ldots \lambda y.\rho) \\ \text{INC} & \boxed{2}\ \mathsf{berry}'(y) \\ \text{PARTS} & \left\langle \begin{array}{l} \boxed{1}, \boxed{1a}\ y, \boxed{1b}\ \Delta, \boxed{1c}\ (\gamma,\Delta), \\ \boxed{1d}\ (\lambda y.\beta), \boxed{1e}\ (\lambda y.\rho), \boxed{1f}\ (\gamma,\Delta)(\sigma_1), \\ \boxed{1g}\ (\gamma,\Delta)(\sigma_1, \lambda y.\beta), \boxed{2}, \boxed{2a}\ \mathsf{berry}' \end{array} \right\rangle \end{bmatrix} \end{bmatrix} \ \&\ \boxed{2a} \triangleleft \beta \ \&\ y \triangleleft \rho$$

According to the SEMANTICS PRINCIPLE, in head-adjunct phrases the EXC of the non-head (*different*) is a subterm of the EXC of the head (*berries*). Now observe that the NP *different berries* as it occurs in sentence (1) is the maximal projection of the noun *berries* (NP_2 in (4)), and NP_2 is a non-head daughter of a verbal projection (VP). Moreover, *different* is the maximal projection of the adjectival non-head *different* in the NP *different berries*. By the INCONT PRINCIPLE and clause (a) of the EXCONT PRINCIPLE[8] it follows that the INC value of *different* equals its own EXC value, i.e. both the INC and the EXC in (7) are $\boxed{1}$ in our sentence. But with $\boxed{1}$ being the largest logical expression on the PARTS list of *different berries* in the sense that all other elements on the PARTS list are subterms of $\boxed{1}$, $\boxed{1}$ must actually equal the EXC value of the non-head *different berries*, as depicted in (8); it cannot be a proper subterm of that EXC. The INDEX DR value and the MAIN value of the

[8](Richter & Kallmeyer, 2009, p. 47) present the two principles as follows:
INCONT PRINCIPLE:
In each *lrs*, the INCONT value is an element of the PARTS list and a component of the EXCONT value.
The EXCONT PRINCIPLE, Clause (a):
In every phrase, the EXCONT value of the non-head daughter is an element of the non-head daughter's PARTS list.

phrase are inherited from its head daughter, *berries*, by the CONTENT PRINCIPLE, as is the INC, $\boxed{2}$ (due to the LRS PROJECTION PRINCIPLE). Also note that the subterm restriction $\zeta \triangleleft \beta$ from (7) becomes more specific in NP_2, because we now know that the relevant MAIN value of the modified head is berry', which is why it is now required that $\boxed{2a}$, which is berry', be a component of the relevant second restrictor of the complex quantifier (γ, Δ).

The noun phrase *different berries* (8) forms a head-complement phrase with the verb *picked* shown in (9). Semantically, this head-complement phrase is an instance of combining a generalized quantifier as interpretation of the NP (recognizable through the form of its EXC) with the semantics of a verbal projection, represented here by the INC of (9).

(9)

$$\begin{bmatrix} \text{PHON} & \langle \textit{picked} \rangle \\ \text{SS LOC} & \begin{bmatrix} \text{CAT VAL} & \begin{bmatrix} \text{SUBJ} & \langle \text{NP}_{\boxed{4a}} \rangle \\ \text{COMPS} & \langle \text{NP}_{\boxed{1a}} \rangle \end{bmatrix} \\ \text{CONT MAIN} & \boxed{3b}\ \mathsf{pick}' \end{bmatrix} \\ \text{LRS} & \begin{bmatrix} \text{EXC} & \boxed{0} \\ \text{INC} & \boxed{3}\ \mathsf{pick}'(\boxed{4a}, \boxed{1a}) \\ \text{PARTS} & \langle \boxed{3}, \boxed{3a}\ \mathsf{pick}'(\boxed{1a}\, y), \boxed{3b}\ \mathsf{pick}' \rangle \end{bmatrix} \end{bmatrix}$$

(10) is the clause of the SEMANTICS PRINCIPLE that imposes combinatory restrictions on the VP *picked different berries* shown in (11).

(10) SEMANTICS PRINCIPLE, Clause 2,
combinations of quantified NPs with a verbal projection
(adapted from (Richter & Kallmeyer, 2009, p. 65))
In each *headed-phrase*, for some $n \geq 2$, if the non-head is a quantified NP with an EXC value of the form *generalized-quantifier*$(\sigma_1, \ldots, \sigma_n, \rho)$, the INC value of the head is a subterm of ρ.[9]

(11)

$$\begin{bmatrix} \text{PHON} & \langle \textit{picked, different, berries} \rangle \\ \text{SS LOC CONT MAIN} & \boxed{3b}\ \mathsf{pick}' \\ \text{LRS} & \begin{bmatrix} \text{EXC} & \boxed{0} \\ \text{INC} & \boxed{3}\ \mathsf{pick}'(\boxed{4a}, \boxed{1a}\, y) \\ \text{PARTS} & \left\langle \begin{array}{l} \boxed{3}, \boxed{3a}\ \mathsf{pick}'(\boxed{1a}\, y), \boxed{3b}\ \mathsf{pick}', \\ \boxed{1}\ (\gamma, \Delta)(\sigma_1, \lambda y.\beta, \ldots \lambda y.\rho), \boxed{1a}\ y, \boxed{1b}\ \Delta, \\ \boxed{1c}\ (\gamma, \Delta), \boxed{1d}\ (\lambda y.\beta), \boxed{1e}\ (\lambda y.\rho), \boxed{1f}\ (\gamma, \Delta)(\sigma_1), \\ \boxed{1g}\ (\gamma, \Delta)(\sigma_1, \lambda y.\beta), \boxed{2}\ \mathsf{berry}'(y), \boxed{2a}\ \mathsf{berry}' \end{array} \right\rangle \end{bmatrix} \end{bmatrix}$$

$$\& \ \boxed{2a} \triangleleft \beta \ \& \ \boxed{3} \triangleleft \rho$$

The VP in (11) inherits its MAIN value and its EXC and INC values from its syntactic verbal head, (9), as dictated by the CONTENT PRINCIPLE and the LRS PROJECTION PRINCIPLE. As always, the PARTS list contains all elements of the PARTS

[9] In the feature-logical encoding in a sort hierarchy, *generalized-quantifier* is a supersort of all quantifier symbols.

lists of the two syntactic daughters. Moreover, complying with (10), the INC of the verb *picked*, $\boxed{3}$, must be in the scope ρ of the quantifier in the EXC of *different berries*, $\boxed{1}$, in the phrase *picked different berries*. In (11) this is shown as the last conjunct in the conjunction of subterm conditions in the last line ($\boxed{3} \triangleleft \rho$).

The difference between (10) and the corresponding standard clause of the SEMANTICS PRINCIPLE in LRS is in the flexible number of restrictors σ_n of the generalized quantifier, and in the different functional types of the restrictors and the scope, which is not visible above because the types are not mentioned in the descriptions. A similar variant of Clause 2 of the SEMANTICS PRINCIPLE is already foreshadowed in the version of this clause in (Iordăchioaia & Richter, 2015, p. 631) in an analysis of Romanian negative concord constructions. The representations of the Romanian counterparts of n-words like *nobody* and *nothing* also have undetermined numbers of restrictors.

The last missing step in the analysis of sentence (1), adding the subject NP *every ape* to the verb phrase *picked different berries* shown in (11), involves all central techniques for the integration of complex unreducible polyadic quantifiers in the specification of the syntax-semantics interface. At its core the successful treatment of unreducible quantifiers hinges on one particular property of lexical semantic resources in LRS, namely on the fact that distinct lexical elements in the same utterance may contribute the same piece of semantic representation structure. This feature of LRS has been used extensively in accounts of negative concord in languages such as French, Polish, and Romanian.[10] To illustrate the point, consider (12) from a variety of American English with negative concord:

(12) This ain't no half-assed sub shop.

LRS captures the observation that (12) clearly has only a single negation reading by assuming that while *ain't* and *no* both lexically contribute a negation operator to the semantics of the sentence, the semantic constraint system of this variety of English forces those two negation operators to be the same negation operator. In essence, the single negation operator in the semantic representation of the sentence is connected to two distinct lexical units.

The parallelism of this type of treatment of negative concord to our construction with *different* NPs becomes more apparent in examples from Romanian negative concord such as (13-a) (Iordăchioaia & Richter, 2015, p. 610 (3c), pp. 638–639), because just like in our sentence with two NPs, *every ape* and *different berries*, the Romanian negative concord construction also involves two nominal phrases, *no student* and *no book* which, *prima facie*, seem to be independent negative quantifiers.[11]

[10] For Polish, see Richter & Sailer (2004); for a typological discussion of French, Polish and German Richter & Sailer (2006); and for Romanian Iordăchioaia & Richter (2015).

[11] In the current discussion I ignore the behavior and influence of the negative verbal prefix *nu*, because this leads to orthogonal morpho-syntactic and semantic considerations.

(13) a. Niciun student nu a citit nicio carte.
no student not has read no book
No student read any book.

b. *no((*x*, *y*), student′*(x)*, book′*(y)*, read′*(x, y)*)*

c. niciun student: *no((*x*, *ν_2*), student′*(x)*, *β*, *ρ*)*

d. nicio carte: *no((*ν_1*, *y*), *α*, book′*(y)*, *ρ'*)*

e. nu a citit: *no((*ν_1'*, *ν_2'*) , *α'*, *β'*, read′*(ν_1', ν_2')*)*

Despite the apparent independence of the two NPs, Iordăchioaia & Richter (2015) analyzes the negative concord reading of the sentence with a single polyadic negative quantifier as shown in (13-b).[12] (13-c)-(13-d) sketch in a highly informal notation what the semantic contributions in LRS of the two NP quantifiers and the negated verb are to the final semantic representation (13-b): *niciun student* contributes a variable x and a restrictor *student′*(x) whose linear position in the expression corresponds to the position of x in the initial sequence of variables: x is the first of two variables, and *student′*(x) is the first of two restrictor arguments. This correspondence can again be observed in (13-d), *nicio carte*, which contributes variable y (in the second position of the sequence of variables) and the linearly second restrictor, *book′*(y). The verb form in (13-e) contributes the material in the scope of the negative quantifier. In order to obtain the representation (13-b), the three sets of semantic contributions have to be matched up, simultaneously obeying all other restrictions on possible representations of the sentence from all LRS composition principles. The only way to do this is to conclude that, despite first appearances, there is in fact only one polyadic negative quantifier in this sentence, and this single polyadic negative quantifier is contributed by all three syntactic units, with the various syntactic slots of the polyadic quantifier filled in by each of the three expressions in turn. Variable x from (13-c) fills the unspecified slots ν_1 and ν_1', y fills the slots ν_2 and ν_2', *student′*(x) equals α and α', and so on. The semantic constraint system of the syntax-semantics interface is specified in such a way that (13-b) is the only solution for the negative concord reading, modulo variable names.[13]

The present syntax-semantics interface for polyadic quantifiers with Δ essentially proceeds along the same lines. There are two differences: (1) the syntactic representation format of polyadic quantifiers is changed, and (2) the NPs that contribute to the same quantificational expression do not contribute the same quantifier constant, instead they contribute different pieces to the same complex quantifier. Again simplifying an LRS constraint set by projecting it onto pseudo-logical expressions in a somewhat *ad hoc* notation that is meant to highlight the underlying ideas, (14) gives an informal sketch of the semantic contributions of the two NPs and the verb to the semantic representation (14-b) of sentence (14-a). In contrast

[12]As already mentioned earlier, Iordăchioaia & Richter (2015) assumes a different syntactic form for polyadic quantifiers than the present study.

[13]In some contexts, the sentence also has a double negation reading with two negative quantifiers. This is also captured by the grammar, but not shown here.

to the Romanian negative concord construction, the verb is not quantificational.

(14) a. Every ape picked different berries.
b. $(\forall, \Delta)(\lambda x.\mathsf{ape}'(x), \lambda y.\mathsf{berry}'(y), \lambda x \lambda y.\mathsf{pick}'(x, y))$
c. every ape: $(\forall, \gamma)(\lambda x.\mathsf{ape}'(x), \sigma_2, \lambda x \lambda \nu_2.\rho)$
d. different berries: $(\gamma', \Delta)(\sigma_1, \lambda y.\mathsf{berry}'(y), \lambda \nu_1 \lambda y.\rho')$
e. picked: $\mathsf{pick}'(\nu_1', \nu_2')$

As we saw already in the lexical specification of *different* and its combination with the noun *berries*, *different* produces an 'incomplete' complex quantifier whose first component must be contributed by some other quantifier. In (14-d) this open slot is marked γ'. To obtain a well-formed semantic expression, the open slot must be filled by some appropriate other lexical resource in the utterance. The principles of the semantic combinatorics of course determine which syntactic units are eligible for providing semantic resources for that slot. In sentence (14-a) there is only one candidate, the determiner *every*, which is contained in (14-c) as the specifier of the NP *every ape*. Lexically *every* must be appropriately underspecified in such a way that one of its realizations contains the universal quantifier, $\forall$, in the first slot of a complex quantifier whose second slot is open (γ in (14-c)). Since $\forall$ is the first component of the complex quantifier, its restrictor must also be in the first restrictor slot, while the second restrictor slot, σ_2, must ultimately be contributed by whatever fills the second syntactic quantifier slot. Note that there is also a lambda abstraction in corresponding linear order: Since $\forall$ is the first component quantifier and its restrictor fills the first slot, the relevant lambda abstraction λx also comes first in (14-c) (and second in (14-d), where it is λy). The other lambda abstraction slot does not contain a concrete variable but an open slot for a variable, ν_2 in (14-c) and ν_1 in (14-d). These two variable slots are filled by the variables from the other contributor to the quantifier, and ν_2 ultimately becomes y whereas ν_1 becomes x in the representation for the sentence, (14-b). Finally, the role of *picked*, (14-e), is to contribute the binary relation in the scope of the lambda abstractions, marked ρ and ρ', respectively.

It should already be clear from the Romanian negative concord example (13) how the semantic constraint set sketched in (14) can be resolved: If we assume that *every ape* and *different berries* contribute the same complex quantifier, (14-c) and (14-d) fill each other's open slots, and (14-e) turns the scope of the resulting completed complex quantifier into a well-formed expression, as shown in (14-b).

In the remainder of this section, I introduce the lexical specifications and discuss the clauses of the SEMANTICS PRINCIPLE that achieve what (14) illustrates. First of all, a lexical entry for *every* is needed whose semantic resources are specified in such a way that the NP quantifier (14-c) emerges as one possible result of combining *every* with *ape*. Doing so with complete formal rigor requires a rather technical relational specification in order to ensure the linear correspondence of the contributed component to the complex quantifier, the restrictor's position, and the order of the lambda abstraction in the scope (first quantifier component, first restrictor, first lambda abstraction). Instead of spelling out a relational constraint over tuples of term structures, I employ a more informal notation that indicates with sufficient clarity what a fully explicit specification would require.

(15)
$$\begin{bmatrix} \text{PHON} \ \langle \textit{every} \rangle \\ \text{SS LOC CONT} \begin{bmatrix} \text{INDEX DR} \ \boxed{4a}\, x \\ \text{MAIN} \ \boxed{4b}\, \forall \end{bmatrix} \\ \text{LRS} \begin{bmatrix} \text{INC} & \boxed{4}\,(\ldots,\forall_n,\ldots)(\ldots,(\lambda x.\alpha)_n,\ldots,\ldots(\lambda x)_n\ldots.\beta) \\ \text{PARTS} & \left\langle \begin{matrix} \boxed{4a}\, x, \boxed{4b}\,\forall, \boxed{4c}\,(\ldots,\forall,\ldots), \boxed{4d}(\lambda x.\alpha), \boxed{4e}\,(\lambda x\ldots.\beta),\ldots, \\ \boxed{4f}\,(\ldots,\forall,\ldots)(\ldots,\lambda x.\alpha),\ldots,\boxed{4} \end{matrix} \right\rangle \end{bmatrix} \end{bmatrix}$$
$$\& \ x \triangleleft \alpha \ \& \ x \triangleleft \beta$$

The notation with subscripts n in the INC is to be read as designating the position of the respective expressions: A quantifier in position n of the complex quantifier goes with a restrictor in position n of a sequence of restrictors of equal length, and with the nth lambda abstraction in a sequence of lambda abstractions. In addition, the lambda abstraction in the quantifier's scope binds the same variable as the lambda abstraction in the designated restrictor.

When *every* becomes the specifier in a nominal phrase with the nominal head *ape*, there is an underspecified complex quantifier for any finite number of restrictors, with the monadic constant $\forall$ contributed by *every* in one of the available positions of the complex quantifier, and the relation ape' of *ape* in the corresponding restrictor slot. But when *every ape* occurs in sentence (1), it is clear that only one choice of representation has a chance to be resolved to a well-formed logical expression: a complex quantifier with $\forall$ in first position, and ape' a subterm of the first restrictor, because this form will match the open slots in the quantificational expression of *different berries*. For any other choice, the two quantifiers from the two NPs cannot be identical, because they would either be inconsistent or there would be syntactic slots in the resulting complex quantifier that cannot be filled by syntactic contributions of lexical elements in the sentence. If the two contributed quantifiers are not identical, one of them must outscope the other. But then they would again have slots that do not contain expressions that are contributed by other lexical elements in the sentence. Such representations are ruled out by one of the standard LRS principles of semantic composition (EXCONT PRINCIPLE, Clause (b): All components of the logical representation of an utterance are contributed by some lexical element in the utterance).

The combination of quantificational determiners with nominal projections is semantically restricted by Clause 1 of the SEMANTICS PRINCIPLE. Here is a modified version for polyadic quantifiers:

(16) SEMANTICS PRINCIPLE, Clause 1,
combinations of quantificational determiners with a nominal projection
(adapted from (Richter & Kallmeyer, 2009, p. 65))
In each *headed-phrase*, for some $n \geq 1$,
if the non-head is a quantificational determiner with INC value of the form

$(\ldots, H_n, \ldots)(\ldots, (\lambda x.\alpha)_n, \ldots, \ldots \lambda x_n \ldots.\rho)$ in which H is a quantifier constant on the PARTS list of the non-head,
then the INC of the head is a subterm of α, and the INC value of the non-head daughter is identical with the EXC value of the head daughter.

Given Clause 1 of the SEMANTICS PRINCIPLE and focusing on the only version of *every ape* that can be consistently combined with *different berries*, we obtain (17) for the semantics of NP_1:

(17)
$$\begin{bmatrix} \text{PHON} & \langle \textit{every, ape} \rangle \\ \text{SS LOC CONT} & \begin{bmatrix} \text{INDEX DR} & \boxed{4a}\, x \\ \text{MAIN} & \boxed{5a}\, \mathsf{ape'} \end{bmatrix} \\ \text{LRS} & \begin{bmatrix} \text{EXC} & \boxed{4}\, (\forall, \psi)(\lambda x.\alpha, \sigma_2, \lambda x.\kappa) \\ \text{INC} & \boxed{5}\, \mathsf{ape'}(x) \\ \text{PARTS} & \left\langle \begin{matrix} \boxed{4}, \boxed{4a}\, x, \boxed{4b}\, \forall, \boxed{4c}\, (\forall, \psi), \boxed{4d}\, (\lambda x.\alpha), \boxed{4e}\, (\lambda x.\kappa), \\ \boxed{4f}\, (\forall, \psi)(\lambda x.\alpha), \boxed{4g}\, (\forall, \psi)(\lambda x.\alpha, \sigma_2), \boxed{5}, \boxed{5a}\, \mathsf{ape'} \end{matrix} \right\rangle \end{bmatrix} \end{bmatrix}$$
$\& \ \boxed{5} \triangleleft \alpha \ \& \ x \triangleleft \kappa$

The INDEX DR and MAIN values are inherited from the syntactic head of the NP (CONTENT PRINCIPLE), and the INC and EXC values are identical to the respective values of the syntactic head, *ape*. But the EXC of *ape* is also identical to the INC of *every* (according to Clause 1 of the SEMANTICS PRINCIPLE), which guarantees it is the quantifier, $\boxed{4}$. Finally, the INC $\boxed{5}$ of *ape* is a subterm of the lambda abstract in the first restrictor of the quantifier (again due to Clause 1 of the SEMANTICS PRINCIPLE).

For the last step, we have to consider Clause 2 of the SEMANTICS PRINCIPLE, (10), the NP *every ape*, (17), and the VP *picked different berries*, (11). According to Clause 2, the INC of *picked* (identical to all INC values along its syntactic head projection), must be a subterm of the scope of *every ape*, just as it must be in the scope of the quantifier of *different berries*, as we saw earlier. Initially this restriction leaves three alternatives, of which we can quickly rule out the first two: If *every ape* outscopes *different berries* or *vice versa*, there are slots in the two quantifiers (ψ in *every ape* and γ in *different berries*) that are not filled with syntactic contributions of other words, excluding these two possibilities as semantic representations of the utterance. This leads to assuming identity of the two quantificational expressions ($\boxed{1} = \boxed{4}$) from NP_1 and NP_2 as the only consistent solution of the semantic constraint set. This means that $\psi = \Delta$, $\gamma = \forall$, $\sigma_1 = \lambda x.\mathsf{ape'}(x)$, $\sigma_2 = \lambda y.\mathsf{berry'}(y)$, and that the scope argument of the polyadic quantifier takes the form $\lambda x \lambda y.\mathsf{pick'}(x, y)$. The result is shown in (18), omitting the long PARTS list which enumerates all and only the subterms of $\boxed{1}$.

(18)
$$\begin{bmatrix} \text{PHON} & \langle \textit{every, ape, picked, different, berries} \rangle \\ \text{SS LOC CONT MAIN} & \boxed{3b}\, \mathsf{pick'} \\ \text{LRS} & \begin{bmatrix} \text{EXC} & \boxed{1}\, (\forall, \Delta)(\lambda x.\mathsf{ape'}(x), \lambda y.\mathsf{berry'}(y), \lambda x \lambda y.\mathsf{pick'}(x, y)) \\ \text{INC} & \boxed{3}\, \mathsf{pick'}(\boxed{4a} x, \boxed{1a} y) \end{bmatrix} \end{bmatrix}$$

The EXC of an utterance contains its logical representation. According to **Definition 2** and the EXC value of (18), *Every ape picked different berries* is true iff any two apes picked unequal sets of berries.

4 Extensions

The main point of this paper is to show that a constraint-based semantics is capable of providing a systematic syntax-semantics interface for the semantics going back to (Keenan, 1992) of sentences with *different* such as (1) and (3). While Iordăchioaia & Richter (2015) show in their analysis of Romanian negative concord constructions that polyadic quantification can be used to give a parsimonious semantic account of difficult data that previously required additional syntactic assumptions about covert LF movement, the present account of *different* goes one step further. The negative polyadic quantifiers in the analysis of Romanian negative concord are reducible polyadic quantifiers in the sense that they can be decomposed into iterations of monadic quantifiers. Thus it is possible to have an analysis of that class of constructions in terms of traditional compositional semantics that assumes the same readings as the account in terms of polyadic quantifiers, even if only at the price of complicating syntax significantly. By contrast, the polyadic quantifiers in the present account of internal readings with *different* are (in-)famous for being unreducible to iterations of monadic quantifiers. It is for that very reason that previous accounts of 'compositional' semantics could not employ the elegant and direct semantics of **Definition 1**.

Seeing the feasibility of a syntax-semantics interface with unreducible polyadic quantification naturally leads to the question how the present approach fits into the general landscape of constructions with *different* that have been discussed in the literature, especially in (Beck, 2000; Brasoveanu, 2011; Bumford & Barker, 2013), and how its coverage can be extended. In this section, I offer a few initial thoughts on aspects of constructions with *different* in light of the current approach.

The main example (1) of the preceding discussion is similar to *Q-bound different* in Beck (2000), except that, similar to Keenan & Westerståhl's (1997) examples, I do not follow Beck's (2000) distinction between singular and plural *different*. One of the reasons is that it seems attractive to subsume under the general characterization of 'quantifiers containing Δ' other quantifiers not mentioned in (Beck, 2000), or (Brasoveanu, 2011):

(19) a. Two apes picked different berries.
b. Five apes picked different berries.
c. Few apes picked different berries.
d. Many apes picked different berries.
e. Most apes picked different berries.
f. No apes picked different berries.[14]

[14] See footnote 5 above for necessary refinements of the present semantics.

This extension is certainly in the spirit of the original idea of complex quantifiers, but it also requires a more careful future study of properties of monadic quantifiers that can enter into this construction.

For any theory of sentence internal *different*, it is crucial to determine in which syntactic and semantic environment *different* must find its licensing second quantifier component. The lexical specification in (7) determines two aspects of the distribution of *different*: It needs to find a second quantifier of Lindström type $\langle 1, 1\rangle$ for the empty first slot of the complex quantifier that it introduces. Moreover, this other quantifier must be structured in such a way that it matches the structure of *different*'s complex quantifier. With the latter property, it also fills the open first restrictor slot. This means that being able to contribute an underspecified complex quantifier structure of this form is a necessary property of determiners (or nominal lexical elements such as *everyone*) to which *different* can attach, and the ability to form such a quantifier determines what can be a felicitous licenser of sentence internal *different*.

In addition to form, scope also plays a role as a licensing condition. Without any further refinements of the theory, candidate licensing quantifiers for *different* must be in *different*'s scope domain. Since the possibility of being in the same scope domain of course depends to a great extent on the scope properties of the NP with *different* itself, this is not a very strong restriction. For example, it is possible that scope islands are weak for *different* NPs, in which case they can find their licenser outside of sentences that are scope islands to other quantifiers. For a first impression on the issues at stake, consider the following examples from (Bumford & Barker, 2013, p. 360):

(20) a. Every boy gave every girl a different poem.
b. Every boy gave every girl he liked a different poem.
c. Every boy said [every girl read a different poem].

Assuming that *a different poem* is analyzed similarly in all relevant aspects to our earlier plural NPs with *different*, there are two quantifiers, *every boy* and *every girl*, in (20-a) that can license the internal reading, because both are in the same scope domain with *a different poem*. Of course, the *different* NP requires via the mechanisms discussed above that one of the two universal quantifiers comes with a semantic representation that can be identified with *different*'s representation.

(20-b) is an example that Bumford & Barker (2013) consider problematic for Brasoveanu (2011), since in the latter approach only the most local distributivity operator (associated with the licensing universal quantifier) can function as the licenser of the *different* NP.[15] The pronoun in the relative clause attached to *every girl* in (20-b) requires that its binding operator outscope *every girl*. As a consequence, *every boy* cannot take immediate scope over *a different poem*, and, if only

[15] See Lahm (2016) for further discussion of this issue, and for problems with the alternative solution proposed by Bumford & Barker (2013). Correctly identifying the licensing NPs and their domain is an open problem.

the closer of the two quantifiers can function as a licenser then only the intervening *every girl* is able to do so. However, both internal readings are available for (20-b): The sentence may mean that every boy chose a different poem for any girl he likes, i.e. no two girls he likes receive the same poem from him. But a girl who is liked by two boys might receive the same poem from the two. Alternatively every boy may give the same poem to every girl he likes, but no two boys give the same poem to the girls they like.

This constellation of readings is compatible with the present approach since the complex quantifier comprising the universal quantifier and *different* arising from *every boy* and *a different poem* can outscope *every girl he liked*, and it is also possible that a monadic quantifier *every boy* outscopes a complex polyadic quantifier arising from *different poem* and *every girl he liked*.

Yet another aspect of licensing domains is illustrated with (20-c), where one of the potential licensers of *a different poem* is in a matrix clause. Without specific additional restrictions, the combinatorics of LRS allows quantifiers in embedded clauses to take scope outside of their clause. It follows that *every boy* and *a different poem* can be in the same scope domain (the matrix clause), and that they can form a complex quantifier in the matrix clause. On the other hand, *every girl* and *a different poem* may form a complex quantifier in the embedded clause. The present theory is so far silent about possible *de re*/*de dicto* readings of *a different poem* that come from its structural position in the embedded clause and the possibility of interpreting it in the matrix clause. Lahm (2016) argues that there is a *de re*/*de dicto* ambiguity of *different* phrases with licensers in this constellation and shows that his theory of the behavior of sentence internal *different* in terms of a restriction on Skolem functions can account for it.[16]

As a final set of examples, consider the following pattern:

(21) a. Every ape picked different berries.
b. Every ape picked two different berries.
c. Every ape picked a different berry.

(21-a)–(21-c) all have sentence internal readings. The last two examples show that at least some specifiers can be added to *different* phrases while retaining the sentence internal reading, and that this phenomenon goes beyond the indefinite singular construction often discussed in the literature. A first conceivable representation of (21-b) that extends the polyadic theory of *different* is shown in (22):

(22) $(\forall, (\mathsf{two}', \Delta))(\lambda x.\mathsf{ape}'(x), \lambda y.berry'(y), \lambda x\lambda y.\mathsf{pick}'(x, y))$

The idea here is to follow the lead of Keenan & Westerståhl (1997) and assume

[16]Another example by Lahm (p.c.) highlights the related issue of intervening modalities between the *different* phrase and its putative licenser as a challenge to the polyadic quantifier analysis:

(i) John and Mary want to live in different cities.

that polyadic quantifiers in natural languages are constructed following certain systematic but rather limited patterns. (22) tentatively extends the polyadic quantifier for *different* by adding the quantifier constant of the specifier to Δ. **Definition 2** can then be modified in a relatively straightforward way through imposing an additional restriction on the restrictor set $[\![\beta]\!]$ that is due to the extra quantifier, parallel to the condition H imposes on $[\![\alpha]\!]$.

The considerations in this section suggest that the polyadic perspective on *different* emphasizes the importance of taking a closer look at its distribution. Recent methods of corpus research could offer interesting new insights that would help to see the advantages and disadvantages of current theories.

5 Conclusion

I demonstrated that the constraint-based semantics of Lexical Resource Semantics can give an explicit semantic account of sentence internal readings with *different* with categorematic unreducible polyadic quantifiers. The present paper mainly focused on the lexical semantic specifications and on the combinatoric principles expressed in the clauses of the SEMANTICS PRINCIPLE that lead to this result. The result is theoretically significant because such a semantics cannot be obtained with the kind of minimalistic syntactic structure assumed in so-called surface-oriented frameworks like LFG and HPSG in combination with the flavor of compositional semantics that is widely adopted in linguistics.

In Section 4 I began to explore new perspectives on the data which are opened up by the polyadic analysis of sentence internal *different*, indicating that the new analysis exhibits a few promising properties. At this early stage there are of course also many open questions: Plural semantics was ignored as well as consequences of intensional contexts, there was no satisfying treatment of singular indefinites, and the readings (2-b) and (2-c) of the ambiguous main example (1) were not addressed. They should soon come into view when turning to the question of assigning an appropriate semantics to the mysterious constant Δ of Lindström type $\langle 1, 1\rangle$, the core of my logical representation of *different*.

References

Barwise, Jon & Robin Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophie* 4. 159–219.

Beck, Sigrid. 2000. The semantics of *different*: Comparison operator and relational adjective. *Linguistics and Philosophy* 23(2). 101–139.

Brasoveanu, Adrian. 2011. Sentence-internal *different* as quantifier-internal anaphora. *Linguistics and Philosophy* 34(2). 93–168.

Bumford, Dylan & Chris Barker. 2013. Association with distributivity and the problem of multiple antecedents for singular *different*. *Linguistics and Philosophy* 36(5). 355–369.

Dekker, Paul. 2003. 'Meanwhile, within the Frege boundary'. *Linguistics and Philosophy* 26. 547–556.

Gallin, Daniel. 1975. *Intensional and higher-order modal logic*. Amsterdam: North-Holland.

Iordăchioaia, Gianina. 2010. *Negative concord with negative quantifiers: a polyadic quantifier approach to Romanian negative concord*: University of Tübingen dissertation. http://nbn-resolving.de/urn:nbn:de:bsz:21-opus-48224.

Iordăchioaia, Gianina & Frank Richter. 2015. Negative concord with polyadic quantifiers. The case of Romanian. *Natural Language and Linguistic Theory* 33(2). 607–658.

Keenan, Edward L. 1992. Beyond the Frege boundary. *Linguistics and Philosophy* 15. 199–221.

Keenan, Edward L. & Dag Westerståhl. 1997. Generalized quantifiers in linguistics and logic. In Alice ter Meulen & Johan van Benthem (eds.), *Handbook of logic and language*, 837–893. Elsevier Science B.V.

Lahm, David. 2016. "Different" as a restriction on Skolem functions. In *Proceedings of SALT 26*, To appear.

Lindström, Per. 1966. First order predicate logic with generalized quantifiers. *Theoria* 32. 186–195.

Richter, Frank & Laura Kallmeyer. 2009. Feature logic-based semantic composition: A comparison between LRS and LTAG. In Anders Søgaard & Petter Haugereid (eds.), *Typed feature structure grammars*, 32–92. Frankfurt a.M.: Peter Lang.

Richter, Frank & Manfred Sailer. 2004. Basic concepts of Lexical Resource Semantics. In *ESSLLI 2003 – Course Material I*, vol. 5 Collegium Logicum, Kurt Gödel Society Wien.

Richter, Frank & Manfred Sailer. 2006. Modeling typological markedness in semantics: The case of negative concord. In Stefan Müller (ed.), *Proceedings of the 13th international conference on Head-driven Phrase Structure Grammar*, 305–325. CSLI Publications.

Westerståhl, Dag. 1989. Quantifiers in formal and natural languages. In Dov Gabbay & Franz Guenthner (eds.), *Handbook of philosophical logic*, vol. IV, chap. IV.1, 1–131. D. Reidel Publishing Company.