Integrating Categorematic Unreducible Polyadic Quantifiers in LRS

Frank Richter[*]

Introduction

This paper has two goals: On the technical side, it presents a general integration of polyadic quantification in Lexical Resource Semantics (LRS) with categorematic (polymorphic) quantifiers in a functional type theory; on the analytic side, it puts these quantifiers to use in an explicit syntax-semantics interface that connects a simple HPSG-style syntax for sentences with the adjectives *different* and *same* to a semantics that interprets their semantic contribution as pieces of an unreducible polyadic quantifier.

Coverage

It has been observed that some readings that arise with the adjectives *different* and *same* exhibit properties that cannot be captured easily by assuming that the nominal phrase resulting from the combination of *different/same* + noun receives the expected noun phrase interpretation as generalized quantifier (as customary for NPs like *most donkeys* or *all aces*). The readings of (1a) and (2a) that we are interested in are (1biii) and (2bii), respectively (more on the other readings in the full paper).

(1) a. Two agencies in my country spy on different citizens.
  b. (i) Two agencies in my country spy on different citizens from the ones we know.
   (ii) Two agencies in my country spy on various/many citizens.
   (iii) The citizens that one of the agencies spies on are different from the citizens that the other agency spies on.

(2) a. Two agencies targeted the same citizens.
  b. (i) Two agencies targeted the citizens we are targeting.
   (ii) Whichever citizens one of the two agencies targeted were also targeted by the second agency.

The second quantifier in each sentence beside *different* or *same* may of course be any numeral quantifier, a universal or existential quantifier, or a generalized quantifier such as *many* or *most*, without affecting the fundamental peculiarity of the construction, i.e. the fact that the interpretation of the *different/same*+noun NP is intrinsically dependent on the interpretation of the second NP in the sentence, as will become clear in the next section.[1] In the remainder of this abstract, we will focus on *different*.

Semantics of Quantifiers with *different*

In order to spell out a semantics for *different*, we introduce a few conventions that will help to keep our notation compact. Given a set E and a binary relation R, $R \subseteq E^2$, for each $x \in E$, we write Rx for the set of objects x bears R to: $Rx = \{y|(x, y) \in R\}$. For the set of citizens agency a spies on, we may thus write $\mathsf{spy}\, a = \{c|(a, c) \in \mathsf{spy}\}$.

With the classification of quantifiers by Lindström (1966), *most* can be regarded a quantifier of type $\langle 1, 1\rangle$, taking two unary relations as arguments, while *most agencies* is of type $\langle 1\rangle$, taking a unary relation as argument. The binary quantifier (*most agencies, every citizen*) is of type $\langle 2\rangle$, because its argument is a binary relation. Given a domain E, subsets A_1 to A_n of E, and a quantifier Q of type $\langle 1^n, n\rangle$, we write $Q^{A_1,\ldots,A_n}$ for $Q(A_1, \ldots, A_n)$. With A a set of agencies and the quantifier 2, we write 2^A for the type $\langle 1\rangle$ quantifier *two agencies*, and $(2, \forall)^{A,C}$ for the type $\langle 2\rangle$ quantifier (*two agencies, every citizen*).

Definition (n-ary quantifiers as n-ary relation reducers)
Assume a universe E, and for each integer m, n (with $n \geq 1$) a relation $R \subseteq E^{m+n}$ and an $\langle n\rangle$-ary quantifier Q. $Q(R) := \{(x_1, \ldots, x_m) \in E^m | Q(\{(y_1, \ldots, y_n) \in E^n | (x_1, \ldots, x_m, y_1, \ldots, y_n) \in R\}) = 1\}$.

[*]Goethe Universität Frankfurt, Germany. Email: f.richter@em.uni-frankfurt.de

[1]Barker (2007) mentions related interesting adjectives which could be covered by a similar semantic theory, such as *distinct, separate, similar, identical, unrelated, mutually incompatible* or *opposite.*

To see how n-ary relation reduction by a quantifier works, assume a binary relation spy and a world in which two agencies, nsa and cia, spy on every citizen c_k. Let Q be the binary quantifier (*two agencies, every citizen*). We obtain $Q(\mathsf{spy}) = \{() \in E^0 | Q(\{(y_1, y_2) \in E^2 | (y_1, y_2) \in \mathsf{spy}\}) = 1\}$. Since for all citizens c_k, $(\mathsf{nsa}, c_k) \in \mathsf{spy}$ and $(\mathsf{cia}, c_k) \in \mathsf{spy}$, $Q(\mathsf{spy}) = \{()\} = 1$. With the notion of quantifiers as n-ary relation reducers at hand, we are ready for a semantics for *different*:
Definition (Semantics of a quantifier containing DIFFERENT (Δ))
For Q a polyadic quantifier of type $\langle 1^2, 2\rangle$ containing Δ, $A, B \subseteq E$, $R \subseteq E^2$, and Q_2 a quantifier of type $\langle 1, 1\rangle$, the interpretation of Q is as follows:
$Q^{A,B}(R) = 1$ iff there is an A', $A' \subseteq A$ such that (1) $Q_2^A(A') = 1$, and (2) for all $x, y \in A'$: $x \neq y \Rightarrow B \cap Rx \neq B \cap Ry$.

Applying this definition to *Two agencies spy on different citizens*, we get the following condition: $(2, \Delta)^{\mathsf{A},\mathsf{C}}(\mathsf{spy}) = 1$ iff there is a subset A' of the set of agencies A such that (1) $2^{\mathsf{A}}(\mathsf{A}') = 1$, and (2) for all $a_1, a_2 \in \mathsf{A}'$: $a_1 \neq a_2 \Rightarrow \mathsf{C} \cap \mathsf{spy}\, a_1 \neq \mathsf{C} \cap \mathsf{spy}\, a_2$. In a world in which C is the set of citizens of the USA and GB, and agency GHCQ spies on all US citizens, agency NSA spies on all GB citizens, we obtain $(2, \Delta)^{\mathsf{A},\mathsf{C}}(\mathsf{spy}) = 1$. Note that, according to our semantics of *different*, the two sets of people being spied on by the two agencies do not have to be disjoint as in the given example.
Some polyadic quantifiers (such as combinations of existential and universal quantifiers) can be reduced to a systematic combination of individual monadic constituent quantifiers, while others cannot. This is made precise in the following definition:
Definition: (Reducibility, Dekker (2003))
A type $\langle 2\rangle$ quantifier Q is $\langle 2\rangle$-reducible iff there are two type $\langle 1\rangle$ quantifiers Q_1 and Q_2 with $Q = Q_1 \circ Q_2$.

In the context of our discussion of *different* (and *same*), a theorem by Keenan (1992) can be exploited to show that the polyadic quantifiers in (1a) and (2b) are not reducible (readings (1biii) and (2bii)):
Theorem: (Reducibility Equivalence, Keenan 1992)
For every domain E and Q_1 and Q_2 reducible quantifiers of type $\langle 2\rangle$:
$Q_1 = Q_2$ iff for all $A, B \subseteq E$: $\quad Q_1(A \times B) = Q_2(A \times B)$

A simple proof based on Keenan's theorem shows that there is no way to have an independent semantics of the quantifiers *two agencies* and *different citizens* and still obtain the semantics (1biii) for sentence (1a) that we are aiming for. Either the semantic analysis must be changed, or the syntactic structure that feeds semantics must be modified so the two seemingly independent nominal phrases form an appropriate syntactic unit at the relevant level of syntax (LF movement). In this paper, however, we want to maintain Keenan's semantics, and we do not envision syntactic structure beyond how HPSG treats NPs and PPs *in situ*: $[_S[_{NP}$ Two agencies$]$ $[_{VP}$ $[_V$ spy$]$ $[_{PP}$ on $[_{NP}$different citizens$]]]]$. In order to achieve this, we need a technique to build representations of unreducible polyadic quantifiers systematically. LRS provides exactly the right tools to do so.

Integration in LRS
Traditionally, LRS employs a syncategorematic syntax for (generalized) quantifiers (Richter and Kallmeyer (2009), Iordăchioaia and Richter (2015)). To achieve maximal generality of a theory of polyadic quantification in LRS, it is advantageous to switch to a categorematic representation instead. In AVM notation, our new LRS representations of *two* and *two agencies* look as follows:

(3) a.
$$\begin{bmatrix} \text{PHON} & \langle \textit{two} \rangle \\ \text{SS LOC CONT} & \begin{bmatrix} \text{INDEX DR} & x \\ \text{MAIN} & \mathsf{two}' \end{bmatrix} \\ \text{LRS} & \begin{bmatrix} \text{EXC} & me \\ \text{INC} & \boxed{1}\ \mathsf{two}'(\lambda x.\alpha, \lambda x.\beta) \\ \text{PARTS} & \left\langle \boxed{1}, \boxed{1a}\ x, \boxed{1b}\ \mathsf{two}', \boxed{1c}(\lambda x.\alpha), \boxed{1d}\ (\lambda x.\beta), \boxed{1e}\ \mathsf{two}'(\lambda x.\alpha) \right\rangle \end{bmatrix} \end{bmatrix} \ \&\ x \triangleleft \alpha \ \&\ x \triangleleft \beta$$

b. $$\begin{bmatrix} \text{PHON} & \left\langle \textit{two, agencies} \right\rangle \\ \text{SS LOC CONT INDEX DR} & x \\ \text{LRS} & \begin{bmatrix} \text{EXC} & \boxed{1}\ \mathsf{two}'(\lambda x.\alpha, \lambda x.\beta) \\ \text{INC} & \boxed{2}\ \mathsf{agency}'(\boxed{1a} x) \\ \text{PARTS} & \left\langle \boxed{1}, \boxed{1a}\ x, \boxed{1b}\ \mathsf{two}', \boxed{1c}\ (\lambda x.\alpha), \boxed{1d}\ (\lambda x.\beta), \boxed{1e}\ \mathsf{two}'(\lambda x.\alpha), \boxed{2}, \boxed{2a} \mathsf{agency}' \right\rangle \end{bmatrix} \end{bmatrix}$$
$\& \ \boxed{2} \lhd \alpha \ \& \ x \lhd \alpha \ \& \ x \lhd \beta$

In the monadic case depicted above, two' is a constant of type $\langle\langle e,t\rangle, \langle\langle e,t\rangle, t\rangle\rangle$, and receives its usual interpretation. For reasons of space and perspicuity, we will from now on employ the linear LRS notation defined in CLLRS by Penn and Richter (2005). The LRS representation of (3a) becomes $\hat{}\,[\{\mathsf{two}'(\lambda x.\alpha[x], \lambda x.\beta[x])\}]$, and (3b) $\hat{}\,\mathsf{two}'(\lambda x.[\{\mathsf{agency}'(x)\}], \lambda x.\beta[x])$ (with the caret indicating EXCONT, curly braces indicating INCONT, and square brackets indicating a subterm relationship).
We generalize the specification of monadic generalized quantifiers like two' to underspecified polymorphic polyadic quantifiers of any type $\langle\langle e,t\rangle, \ldots \langle\langle e,t\rangle, \langle\langle e, \ldots \langle e,t\rangle \ldots\rangle, t\rangle\rangle \ldots\rangle$ (Lindström type $\langle 1^n, n\rangle$). Furthermore, we assume that their standard interpretation will be in the form of iterated or resumptive monadic quantifiers, and that their syntactic form keeps track of which restrictor belongs to which monadic quantifier. For English *two*, we obtain $\hat{}\,[\{(\ldots, \mathsf{two}'_n, \ldots)(\ldots, (\lambda x.\alpha[x])_n, \ldots, \ldots \lambda x.\beta[x])\}]$, where the subscript n indicates the position of two' in the representation of an $n+m$-ary polyadic quantifier and of its corresponding restrictor in the sequence of arguments. The representation of *two agencies* straightforwardly becomes $\hat{}\,(\ldots, \mathsf{two}'_n, \ldots)(\ldots, (\lambda x.[\{\mathsf{agency}'(x)\}])_n, \ldots, \ldots \lambda x.\beta[x])$ without any modification to the combinatory apparatus of LRS.
The special properties of polyadic *different* are twofold: It may not be monadic, and the interpretation of polyadic quantifiers containing Δ proceeds as before. Rephrasing the earlier definition for our type-theoretic language yields:
Definition (Restated semantics of a quantifier containing DIFFERENT (Δ))
For $Q = (Q_2, \Delta)$ a polyadic quantifier of type $\langle 1^2, 2\rangle$ containing Δ, x, y variables of type e, α, β expressions of type t, Q_2 a monadic generalized quantifier, and rel a relation of type $\langle e, \langle e,t\rangle\rangle$, the interpretation of Q is as follows:
$\mathsf{V}_{M,g}((Q_2, \Delta)(\lambda x.\alpha, \lambda y.\beta, \mathsf{rel})) = 1$ iff there is an A', $A' \subseteq \mathsf{V}_{M,g}(\lambda x.\alpha)$ such that (1) $\mathsf{V}_{M,g}(Q_2(\lambda x.\alpha))(A') = 1$, and (2) for all $\mathsf{e}_1, \mathsf{e}_2 \in A'$: $\mathsf{e}_1 \neq \mathsf{e}_2 \Rightarrow \mathsf{V}_{M,g}(\lambda y.\beta) \cap (\mathsf{V}_{M,g}(\mathsf{rel}))\mathsf{e}_1 \neq \mathsf{V}_{M,g}(\lambda y.\beta) \cap (\mathsf{V}_{M,g}(\mathsf{rel}))\mathsf{e}_2$.

It remains to be shown how the combinatorics of LRS determines the appropriate semantic representation for (1a) in our type theoretic language, $(\mathsf{two}', \Delta)(\lambda x.\mathsf{agency}'(x), \lambda y.\mathsf{citizen}'(y), \lambda x \lambda y.\mathsf{spy}'(x,y))$.
First of all, the representation of *different* is lexically provided as $\hat{}\,[\{(\gamma, \Delta)(\sigma, \lambda y.\beta[y], \ldots \lambda y.[y])\}]$.
When it combines with a nominal head such as *citizens*, the standard clause of the SEMANTICS PRINCIPLE for determiner-noun combinations applies and ensures (together with ordering restrictions for restrictors in polyadic quantifiers) that the INCONT of *citizens* ends up in β in the NP *different citizens*: $\hat{}\,(\gamma, \Delta)(\sigma, \lambda y.[\{\mathsf{citizen}'(y)\}], \ldots \lambda y.[y])$. If we further assume that case marking prepositions like *on* do not make any semantic contribution of their own, the semantic restrictions on the PP *on different citizens* are the same as on the embedded NP. The NP *two agencies* could in principle be constructed as a monadic quantifier. However, unless it constructs as a binary quantifier compatible with the restrictions that we have seen in *on different citizens*, the type system of the logical language ultimately prevents a coherent semantic integration of the contributions of the NP and the PP in the sentence. The only specification that will be compatible is $\hat{}\,(\mathsf{two}', \zeta)(\lambda x.[\{\mathsf{agency}'(x)\}], \psi, \lambda x.[x])$.
For the VP *spy on different citizens*, the quantifier of *on different citizens* combines with the contribution of *spy*, $\hat{}\ [\{\mathsf{spy}(\boxed{x}, \boxed{y})\}]$ in the usual manner, yielding $\hat{}\,(\gamma, \Delta)(\sigma, \lambda y.[\mathsf{citizen}'(y)], [\lambda y.\{\mathsf{spy}'(\boxed{x}, y)\}])$.
In the last step, we see that the INCONT of the VP must be in the scope of the binary polyadic quantifier *two agencies*, and a coherent well-typed expression can only result from assuming that the polyadic quantifier originating from *on different citizens* is identical with the binary quantifier originating from *two agencies*.

Conclusion

The present analysis of polyadic quantifiers subsumes the analysis of polyadic negative quantifiers introduced for a theory of negative concord in Iordăchioaia and Richter (2015), which is specialized to the case of a negative quantifier being the only source of (resumptive) polyadic quantification. Most importantly, the present study of *different* showed that LRS provides the means to integrate **unreducible** polyadic quantifiers in a systematic syntax-semantics interface in HPSG, which, to the best of our knowledge, has not been done in any other syntactic framework before. The full paper will contain all details of the extended combinatory system in LRS that could only be sketched above.

References

Chris Barker. Parasitic scope. *Linguistics and Philosophy*, 30(4):407–444, 2007.

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

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

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

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

Gerald Penn and Frank Richter. The other syntax: Approaching natural language semantics through logical form composition. In Henning Christiansen, Peter Rossen Skadhauge, and Jørgen Villadsen, editors, *Constraint Solving and Language Processing. First International Workshop, CSLP 2004, Roskilde, Denmark, September 1-3, 2004, Revised Selected and Invited Papers*, volume 3438 of *Lecture Notes in Computer Science*, pages 48–73. Springer, 2005.

Frank Richter and Laura Kallmeyer. Feature logic-based semantic composition: A comparison between LRS and LTAG. In Anders Søgaard and Petter Haugereid, editors, *Typed Feature Structure Grammars*, pages 32–92. Peter Lang, Frankfurt a.M., 2009.