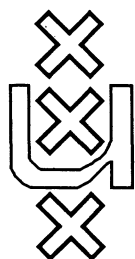


Institute for Language, Logic and Information

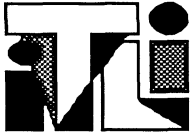
POLYADIC QUANTIFIERS

Johan van Benthem

ITLI Prepublication Series 87-04



University of Amsterdam



Institute for Language, Logic and Information
Instituut voor Taal, Logica en Informatie

POLYADIC QUANTIFIERS

Johan van Benthem
Department of Mathematics and Computer Science
University of Amsterdam

Received August 1987

To appear in Linguistics and Philosophy

Correspondence to:

Faculteit der Wiskunde en Informatica
(Department of Mathematics and Computer Science)
Roetersstraat 15
1018WB Amsterdam

Faculteit der Wijsbegeerte
or (Department of Philosophy)
Grimburgwal 10
1012GA Amsterdam

I Monadic and polyadic quantifiers

Standard generalized quantifiers are of the unary form

$$Qx \bullet \phi(x),$$

with a set-theoretic interpretation of the type ' $[\phi] \in Q$ '.
Polyadic quantifiers generalize this to higher arities:

$$Qx_1 \dots x_n \bullet \phi(x_1, \dots, x_n).$$

For instance, the following binary form defines the set of all *transitive* binary relations:

$$Qxy \bullet \phi(x,y) := \forall x \forall y (\phi(x,y) \rightarrow \forall z (\phi(y,z) \rightarrow \phi(x,z))).$$

The linguistic uses of the unary notion (introduced in Mostowski 1957) have been amply demonstrated in the well-known trilogy Barwise and Cooper 1981, Higginbotham and May 1981, Keenan and Stavi 1986. Recently, however, linguists have also turned toward the more general version (due to Lindström 1966), witness Keenan 1987b and May 1987. This note addresses two issues concerning this new development: its empirical motivation, and its theoretical properties.

II Empirical evidence

Leaving aside such technical examples as the above one, is there any evidence for higher polyadic quantification in natural language? This is one central empirical question to be answered.

One somewhat trivial example is induced by the basic case itself:

- *Iterated unary quantifiers*

Iteration creates complexes such as

$$Q_1 x \bullet Q_2 y \bullet \phi(x,y)$$

(as in "*every boy loves a girl*").

But of course, treating this as a binary complex per se ignores the crucial Fregean insight, which put an end to traditional ad-hoc theories of 'multiple quantification' (see Dummett 1973, chapter 1): such complexity can already be accounted for by a compositional use of the meanings of unary quantifiers.

These iterated cases gain interest, however, with certain additions; such as in Keenan's example

"every boy loves a *different* girl".

Here, the meaning is no longer a simple decomposable $\forall\exists$, as the dependency expressed should now be one-to-one. Keenan takes the latter to be a genuine binary generalized quantifier.

Still, one might prefer to treat "different" here as a higher-order operator on an ordinary unary iteration - reflecting our intuitive ideas about the compositional structure of this sentence, as being a 'connected' (or 'frozen') iteration.

Further examples of this phenomenon arise through

- *Iteration with anaphoric links*

Unary iterations can be 'tied together' by anaphoric links. Again, one might prefer to analyze such cases as (higher-order results of transformations applied to) instances of the basic unary pattern. This will work, e.g., with

"every boy loves a girl-friend of his";

using the unary predicates 'boy' and 'love a girl-friend of oneself'. But it will not work, apparently, with the Bach-Peters type sentences considered by May:

"a boy who loved her left the girl who despised him".

As May argues, we seem to need quantification over *couples* of individuals here to get the correct reading.

A related perspective is found in Fenstad et al. 1985:

- *Parametrized unary quantifiers*

The donkey sentence

"every farmer who owns a donkey, beats it"

can be analyzed as a parametrized unary case 'every A B' with a parameter x:

(every farmer who owns a donkey x)y•y beats (x).

What should this 'parametrization' mean? One idea is to say that every actual value supplied for x turns this into an ordinary unary case, i.e.,

$\forall x:\forall y((\text{farmer}(y) \ \& \ \text{owns}(y,x) \ \& \ \text{donkey}(x))\rightarrow\text{beats}(y,x))$.

But, this fails with a sentence like

"most farmers who own a donkey, beat it":

which does not mean 'for all donkeys, for most farmers...'. The better strategy seems to consist in using couples again, and hence polyadic quantifiers:

'every xy•....', 'most xy•...'

Digression. Of course, problems remain on the latter reading too - as the 'most'-sentences now need not imply (the unary reading of)

"most farmers who own a donkey, beat a donkey":

which does seem to be a logical consequence of the former sentence, whatever its construal. But, this is not our main concern here. \square

As a final example, consider another case studied by May:

- *Resumptive quantifiers*

The sentence "no one liked no one" has a reading of the form

No $xy \bullet \phi(x,y)$

expressing that no couple (x,y) belonged to $[\phi]$. Now, since neither of the two iterated unary readings for the 'no'-quantifiers has this meaning, the binary approach seems necessary here.

Note, however, that there are unary reductions here in a broader sense. Thus,

No $xy \bullet \phi(x,y) \Leftrightarrow$ No $x \bullet \exists y \bullet \phi(x,y)$

One $xy \bullet \phi(x,y) \Leftrightarrow$ One $x \bullet \exists y \bullet \phi(x,y)$ & One $y \bullet \exists x \bullet \phi(x,y)$.

We shall return to this phenomenon below.

Summing up, the claim seems justified that

- (1) there is a good case to be made for the necessity of higher (non-unary) types of generalized quantifier in natural language;
- (2) but, many of these cases are still similar to the standard ones, in that they amount to treating *tuples* of individuals like individuals themselves.

To get yet higher cases, one should look at genuine *branching* quantification (if that exists), or perhaps at Keenan's type of example, which does not reduce to ordinary quantification over tuples (see below).

III Logical properties

The preceding discussion at least motivates taking a look at the logical properties of polyadic quantifiers. For convenience, and practical importance, we restrict attention to the *binary case*.

III.1 General constraints

Already on a universe with n individuals, the class of potential binary generalized quantifiers is quite large. Categorially, the type of Q in the schema $Qxy \bullet \phi(x,y)$ is

$((e,(e,t)),t)$:

and the size of the corresponding denotational domain is $2^{2^{(n^2)}}$. But, there are some plausible *constraints*: as was already the *raison d'être* for the theory of the unary case (see the survey Westerståhl 1986).

• *Logicality*

The general concept of Logicality applies here too (cf. van Benthem 1986, chapter 3): as *invariance* of Q under permutations of binary relations induced by *permutations of the individuals*. For such permutations π , one requires

$$R \in Q \text{ iff } \pi[R] \in Q, \\ \text{for all binary relations } R.$$

(Thus, one retains the 'arrow pattern' of the relation, while disregarding the specific individuals occurring at their ends.) To see the effect of this requirement, one must determine the relation

$$R \approx S,$$

defined as ' $S = \pi[R]$ for some individual permutation π '.

For unary relations R, S , this just amounts to equicardinality. For binary R, S , the behaviour of \approx is more complex:

Example. With $n=2$, \approx has 10 equivalence classes.

Digression. There is a logical characterization of \approx :

Proposition. The following are equivalent on a finite universe M :

- i) $R \approx S$
- ii) $M, R = \sigma(X)$ iff $M, S = \sigma(X)$, for all first-order formulas σ in one binary predicate letter X and identity
- iii) ii) only for universal positive first-order σ .

Proof. By elementary model theory. ■

But this result still does not give one single numerical invariant matching \approx . □

To continue, a logical quantifier Q can now be fully specified as a set of \approx -equivalence classes accepted by it. Examples of such logical binary quantifiers are

- 1) all iterations of logical unary quantifiers,
- 2) all resumptive quantifiers reducing to logical unary quantifiers over couples, but also
- 3) such cases as the earlier-mentioned collection of all transitive binary relations.

Behind all these cases lies a general result (see van Benthem 1986, chapter 7.5):

Proposition . Any predicate over binary relations which can be defined by means of some formula of Type Theory (being a full lambda language with equality) using logical parameters only, is itself logical.

A converse holds too. Every logical polyadic quantifier on some fixed finite universe is definable in the above Type Theory. (See van Benthem 1987 for a general connection between logical invariance and type-theoretic definability.)

- *Special Invariance Properties*

There are also stronger forms of permutation invariance. One of them can be used to delineate the earlier important subclass of quantifiers over tuples (but otherwise 'first-order'). Here, one requires *invariance for permutations of couples of individuals*. Thus, essentially, one can only express conditions on the cardinality of the set $[\phi]$. As every permutation of individuals induces a unique permutation of couples (though not conversely!), this indeed strengthens the earlier kind of logicity.

For a non-example, consider the earlier-mentioned Keenan quantifier 'every A R a different B'. It holds in the left-most situation depicted below, but not in its companion (arising from a permutation of couples):



Other, intermediate types of permutation on couples may be used as well to describe important special classes of polyadic quantifiers. Examples can be found in Higginbotham and May 1981, and De Mey 1987 (on *reciprocals*). Here is an illustration from the former paper. Permutations of couples may be introduced by individual permutations, as in the definition of Logicity:

$$\pi(a,b) = (\pi(a), \pi(b)).$$

But also, independent permutations might be allowed for the two argument positions:

$$\pi(a,b) = (\pi_1(a), \pi_2(b)).$$

Invariance under such *duplex* permutations defines a new class of quantifiers, in between the logical ones and the resumptive cases.

Example . $\lambda R. \exists x R x x$ is logical, but not duplex-invariant.

$\lambda R. \exists x \forall y R x y$ is duplex-invariant, but not resumptive.

A more complex example of this kind would be:

$\lambda R. \forall x \forall y \exists z (Rxz \wedge Ryz)$: which is not a 'unary iteration' in the sense of the following Section. \square

The example also suggests a more formal way of registering the effects of special invariance properties; in terms of their behaviour on standard *first-order* statements about the relation R. As was noted above, *all* such statements are logical. But beyond that, restrictions appeared. For instance, is there a perspicuous syntactic characterization of those first-order formulas which define duplex-invariant polyadic quantifiers?

- *Conservativity*

Eventually, the above discussion will have to be generalized to *restricted* settings: for reasons analogous to those governing unary quantifiers (where the basic pattern is $(QA)B$ or $Q(A,B)$), as well as some new ones, witness the examples in Section II.

For instance, resumptives such as

"no A likes no B"

call for a representation somewhat like this:

	A	B	
No		•	L(x,y)
	x		y

And, the earlier-mentioned donkey sentences are even explicitly of the form

All $x^S y \bullet R(x,y)$.

So, in general, the restriction itself can be a *relation* on the tuple of relevant variables (cf. Higginbotham and May 1981).

Remark. Keenan 1987a shows that the restriction in the first type of example (being technically of type (1,1,2)) cannot be naturally reduced to that in the second (which is of type (2,2)). The obvious move: replacing A,B by the binary relation AxB , has certain pitfalls. \square

There is room here for a generalization of such 'unary' topics as *conservativity* (see Keenan and Stavi 1986), and the interplay of restricting and predicative argument positions generally. Van Eyck 1987 presents a first attempt. For instance, in the last-mentioned case of restricted binary forms $Q(S,R)$, Conservativity becomes

$Q(S,R)$ iff $Q(S,R \cap S)$;

and Logicality likewise

$Q(S,R)$ iff $Q(\pi[S],\pi[R])$
for all individual permutations π .

And similar definitions are possible for the case with two unary restrictions; where, e.g., Conservativity assumes the form

$Q(A,B,R)$ iff $Q(A,B,R \cap (A \times B))$.

Nevertheless, for reasons of technical convenience, we shall stay with the simpler unrestricted forms in the following Section.

III.2 Unary Definability

Given the interest in the border-line between unary iterations and essentially polyadic quantifiers (compare various reductions discussed in May 1987), the following question arises:

Is there also some kind of invariance characterizing the subclass of binary quantifiers definable by unary compounds?

Indeed, there is.

We start with a

Definition . A quantifier $Q_{xy} \bullet \phi(x,y)$ is a *unary complex* if it can be defined as a Boolean combination of forms

$$Q_1x \bullet Q_2y \bullet \phi(x,y),$$

with Q_1, Q_2 logical unary quantifiers.

First, we isolate an invariance property of such complexes.

Definition . Set $R \sim S$ if, for all individuals x ,

$$|R_x| = |S_x|.$$

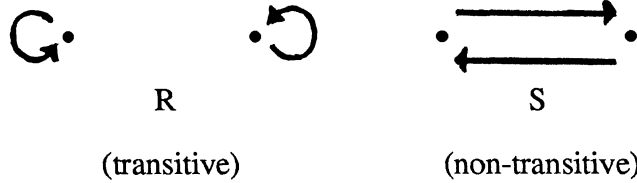
Here, R_x stands for $\{y \mid (x,y) \in R\}$.

A quantifier Q is *right-oriented* if it is closed under the relation \sim .

Proposition . Unary complexes are right-oriented.

Proof . For all individuals x , $Q_2y \bullet Rxy$ holds iff $R_x \in Q_2$, iff (by the definition of $R \sim S$, and permutation invariance for Q_2) $S_x \in Q_2$, i.e., $Q_2y \bullet Sxy$. But then, $Q_1x \bullet Q_2y \bullet Rxy$ if and only if $Q_1x \bullet Q_2y \bullet Sxy$. \square

As an application, note that the earlier Transitivity is not monadically definable - witness the following counter-example (where $R \sim S$):



Remark. This result can be extended to include converse forms of definition $Q_1x \bullet Q_2y \bullet R_{yx}$ - by using an additional requirement concerning predecessors: $|\underset{x}{R}| = |\underset{x}{S}|$. (E.g., Transitivity will still remain undefinable, as the above R, S also satisfy this additional requirement.) The relevance of this extension is shown by the earlier unary definition given for the binary quantifier 'One $xy \bullet \phi(x, y)$ '.

Is the above condition also *sufficient* for unary definability? One illustration is provided by the earlier resumptives. These are all right-oriented. (The reason is this. If $|\underset{x}{R}| = |\underset{x}{S}|$ for all x , then $|R| = |S|$.) And in fact, they are all definable by unary complexes - at least locally in each finite universe.

Example. $|R| = 2$, in a universe with 3 individuals.
A defining form is this (with 'Pi' for 'exactly i'):

$$(P1x \bullet P2y \bullet Rxy \ \& \ P2x \bullet \neg \exists y \bullet Rxy) \ \vee \\ (P2x \bullet P1y \bullet Rxy \ \& \ P1x \bullet \neg \exists y \bullet Rxy).$$

This observation inspires the following general result.

Theorem. On a finite universe, a binary quantifier Q is definable by some unary complex if and only if
i) Q is logical (i.e., permutation-invariant), and
ii) Q is right-oriented.

Proof. 'Only if'. This follows from the preceding observations.

'If'. Suppose that Q satisfies i) and ii). Let there be n individuals. The following unary complex defines the quantifier Q :

$$\bigwedge_{R \in Q} \bigwedge_{i=1}^n P_{n_i x} \bullet P_{i y} \bullet Rxy ,$$

where the conjuncts enumerate all sizes $|R_x|$ occurring in R with their exact multiplicity.

To show that this works, it suffices to check that, if a relation S satisfies this formula, then it must belong to Q . Now, S will satisfy some disjunct, and hence it has the same ' R_x -distribution' as some $R \in Q$. Let π be any permutation of the individuals sending the n_i x having exactly i S -successors to those having

exactly i R -successors. Then we have

$$S \approx \pi[S] \sim R.$$

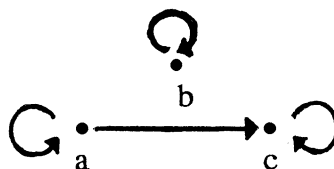
And so, by i) and ii), S must be in Q too. ■

The preceding definability result is only local, in some specific universe. But it can probably be extended to provide a characterization of unary definability uniformly in all finite universes.

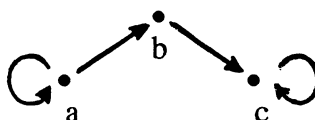
On the other hand, refutations by this method are strong: in that they even refute unary definability within one specific model. Another illustration of this phenomenon is the Keenan quantifier, read technically as

' $\forall x \exists y Rxy$ & R contains a 1-1 function with the same domain'.

The latter statement is true in the following situation (take the identity function):



But it fails in the next situation; although the three points there have the same numbers of successors and predecessors (i.e., \sim holds):



No one-to-one function can be selected, as there would be a clash in the values for b, c .

Incidentally, the Keenan quantifier is not first-order definable in general - and it is not even first-order definable on the finite universes (as may be proved by a Fraïssé-type game argument).

Remark . Keenan himself (personal communication) doubts the above higher-order reading for his 'different' sentence. But the pictorial argument also seems to go through for a whole range of other meanings for this quantifier combination.

Keenan 1987 also studies the question of unary reducibility. His notions and results seem somewhat different, however, from those presented here; involving various technicalities.

Addendum. In the meantime, Sjaak de Mey has suggested that the analysis given here is reminiscent of a certain type of permutation invariance found in Higginbotham and May 1981. Call a permutation of couples *dependent duplex* if it can be written in the following form, allowing movement of the second argument in dependence on the first:

$$\pi(a,b) = (p(a), q_a(b)),$$

where p is a permutation of the individuals,
and all q_a are injections defined on R_a .

The exact correspondence is as follows.

Proposition. A binary quantifier is definable by some unary complex if and only if it is invariant for dependent duplex permutations.

Proof. It suffices to show that dependent duplex invariance is equivalent to logicity plus right-orientation. From left to right, logicity is the special case where all q_a equal p . Also, right-orientation follows by letting p be the identity map. From right to left, note that

$$R \approx \{(a, p^{-1} \circ q_a(b)) \mid (a,b) \in R\} (=R^*)$$

and

$$\pi(R^*) = \{(p(a), q_a(b)) \mid (a,b) \in R\}.$$

Then apply right-orientation and logicity. ■

III.3 Ordering and scope

Although unary iterations are not essentially polyadic, they do raise some interesting questions of their own, beyond the standard unary framework. For instance, several authors have studied scope and order of operators in this setting. Many expressions show a certain freedom of behaviour here, which has intrigued linguists. A few examples will illustrate this emerging trend.

- *Proper names*

Zwarts 1986 considers generalized quantifiers which *lack scope with respect to Boolean connectives*. Notably, *proper names* show a collapse of sentence negation and predicate negation:

$$\text{"Mary (doesn't complain)" } \Leftrightarrow \text{"Not (Mary complains)".}$$

This property is called 'self-duality' in Löbner 1987: $Q = \neg Q \neg$. It seems already so strong that it might completely determine the proper names. But, this is not quite true.

Example . Consider a universe $\{1,2,3\}$ with a quantifier $Q = \{\{1\}, \{2\}, \{1,2\}, \{1,2,3\}\}$. Then Q is self-dual without even being a filter - and hence it cannot be the denotation of any proper name. \square

But then, proper names also satisfy distribution over conjunction or disjunction:

"Mary (complains or worries)" \Leftrightarrow "(Mary complains) or (Mary worries)".

Using a standard characterization of principal ultrafilters, Zwarts concludes that

Proposition . The proper names are precisely those generalized quantifiers lacking scope with respect to Boolean connectives.

- *Scopeless quantifiers*

Another notion of scopelessness arises with iterated unary quantifiers in Zimmermann 1987, who considers interchangeability of a quantifier Q with all generalized quantifiers Q' , in the schema (with either pure or restricted occurrences of Q, Q'):

$$Qx \bullet Q'y \bullet Rxy \text{ iff } Q'y \bullet Qx \bullet Rxy.$$

Again, proper names are the prime example here - and Zimmermann proves a converse too:

Proposition . The scopeless quantifiers are precisely the proper names.

To illustrate the kind of reasoning involved, we give a simplified version of his proof. We derive scopelessness with respect to Boolean operations - which reduces the proposition to the preceding result.

Negation . The following semi-syntactic calculation suffices:

$$\begin{aligned} \neg X \in Q &\text{ iff } Qy \bullet \neg Xy \text{ iff } Qy \bullet \exists z (z=z \ \& \ \neg Xy) \text{ iff} \\ Qy \exists z \neg (z \neq z \vee Xy) &\text{ iff (!) } \exists z \neg (Qy \bullet (z \neq z \vee Xy)) \text{ iff } \exists z \neg (X \in Q) \\ (\text{as } \lambda y \bullet (z \neq z \vee Xy) &= \lambda y \bullet Xy) \text{ iff } X \notin Q. \end{aligned}$$

Disjunction . Let $\{X_i \mid i \in I\}$ be a family of subsets of the universe. Using the Axiom of Choice, select a subfamily $\{X_j \mid j \in J\}$, together with a set Y of representatives $y_j (j \in J)$ such that i) the union of the X_i equals that of the X_j , and ii) each y_j belongs to a unique X_j (for $j \in J$). Then, define a binary relation R among individuals as follows:

$$Ryx \text{ if } y=y_j \text{ for some } j \in J \text{ such that } x \in X_j.$$

Note that $x \in \cup \{X_i \mid i \in I\}$ iff $x \in \cup \{X_j \mid j \in J\}$ iff $\exists y \in Y \bullet Ryx$.

Now calculate as follows:

$$\bigcup \{X_i \mid i \in I\} \in Q \text{ iff } Qx \bullet \exists y \in Y \bullet Ryx \text{ iff (!) } \exists y \in Y \bullet Qx \bullet Ryx \text{ iff} \\ \exists y_j \in Y \bullet X_j \in Q \text{ (as } \lambda x \bullet Ry_j x \text{ defines } X_j) \text{ iff } \exists j \in J \bullet X_j \in Q \text{ iff } \exists i \in I \bullet X_i \in Q.$$

As to the latter equivalence, one half is obvious, since $J \subseteq I$. Conversely, starting from any given $i \in I$, the selected family $\{X_j \mid j \in J\}$ can be chosen so as to contain X_i . ■

Remark . In a sense, proper names are not genuine generalized quantifiers, having been raised from type e to type $((e,t), t)$ (cf. van Benthem 1986, chapter 7). Thus, their freedom of movement in the latter category may be really a sign of 'low status'. It would be of interest to know if something similar holds in general.

Type raising has other uses in this setting too. For instance, in Keenan 1987a, the question is studied which polyadic quantifiers can be viewed as natural lifted versions of monadic counterparts. Notably, Noun Phrases in direct object position have a type $((e,(e,t)),(e,t))$, which is derived from that of their occurrences in subject position, being $((e,t),t)$. A short survey of possible *lambda-normal-forms* for this transition yields only two possible outcomes:

$$\lambda x_{((e,(e,t)),(e,t))} \bullet \lambda y_e \bullet c_{((e,t),t)}(x(y)) \quad , \\ \lambda x_{((e,(e,t)),(e,t))} \bullet \lambda y_e \bullet c_{((e,t),t)}(\lambda u_e \bullet x(u)(y)) \quad .$$

These two formulas represent the usual analysis (with small scope for the direct object Noun Phrase), as well as a passive variant thereof. Of course, there may also be further 'emergent' readings, not obtainable via this mechanism of generalization.

• *Self-commuting quantifiers*

Finally, a special case of scopelessness in the above sense is displayed by the *self-commuting* quantifiers of van Benthem 1984:

$$Qx \bullet Qy \bullet Rxy \Leftrightarrow Qy \bullet Qx \bullet Rxy.$$

Prime examples are the existential and universal quantifiers. E.g., "everyone loves everyone" is equivalent to "everyone is loved by everyone". (So, for these quantifiers, Passivization is a meaning-preserving transformation. Another linguistic aspect of these quantifiers seems to be that they do not allow genuine *branching* with respect to themselves.) Non-examples are already such first-order quantifiers as *exactly one* , *at least two*. The matter is studied further in Westerståhl 1986a, who proves this

Proposition . The only (upward) monotone self-commuting quantifiers are *all* , *some*, *true* and *false*.

Proofs of such results have a more combinatorial flavour than those in the standard unary theory. This is already shown in the following

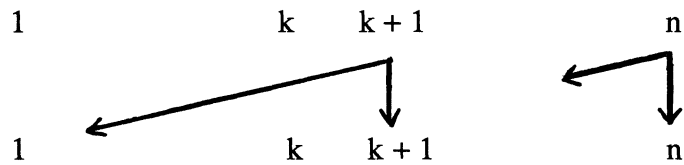
Example. On a universe with 2 elements, *exactly one* is still a self-commuting quantifier. □

By a somewhat laborious calculation, Westerståhl's result can be improved to the following

Proposition. All, some, true and false are the only self-commuting continuous quantifiers.

Instead of a proof, here is an

Example. The downward monotone ('persistent') quantifier $Q = \textit{at most } k$ is not self-commuting. To see this, consider the following picture of a universe with n individuals ($n > k$):



Here, $1, \dots, k$ have no R -successors; while, for $i > k$, $(i, i), \dots, (i, i-k) \in R$ (i.e., $k+1$ R -successors each). It is easy to check that i) $Qx \bullet Qy \bullet Rxy$, while ii) not $Qy \bullet Qx \bullet Rxy$. (Ad ii : $1, \dots, k$ have at most k R -predecessors - e.g., k is preceded by $k+1, \dots, k+k$ -, but so does n , which has only one predecessor). ■

Finally, self-commuting quantifier pairs form a special case of what may be called *converting* binary quantifiers, satisfying the condition

$$Q(R) \text{ iff } Q(\overset{\smile}{R}) \quad (\text{with } \overset{\smile}{R} \text{ the converse relation of } R)$$

This notion has again interesting connections with earlier ones from Section II. For instance, since $\overset{\smile}{R}$ satisfies the same numerical conditions on its set of pairs as R , we have:

If a binary quantifier is invariant for permutations of pairs, then it is converting.

The converse does not hold in general; witness the case of

$$Qxy \bullet Rxy := \forall xy (Rxy \rightarrow Ryx).$$

But, for self-commuting iterations $Qx \bullet Qy$, the two notions may actually be equivalent.

Here is where one can ask for a full-scale extension of the standard theory of unary generalized quantifiers to iterated, and eventually to all polyadic cases. Here is also where we stop.

III.4 A Generalized Perspective

Quantifiers form only one special type of expression. Nevertheless, their study often brings to light semantic phenomena of wider significance across natural language. One way of formulating these is in a Categorical Grammar, with an associated Type Theory (see van Benthem 1986, chapters 3 and 7).

For instance, does the *iterative versus complex* distinction drawn in the above also make sense with other categorial types of expression? The transitive verb pattern studied earlier involve the following types:

(u,v) NP1	(s,x) TV	(y,z) NP2
--------------	-------------	--------------

Here, the functors NP1, NP2 should be able to combine with the TV argument in any order, with the same type of outcome. Therefore, they must have identical types. Moreover, assuming that the final step will be an ordinary application, and the first step a composition ('parametrized application'), there can be only general pattern which fits:

((s,y),y)	(s,(s,y))	((s,y),y).
-----------	-----------	------------

Unfortunately, no other contexts of this kind seem to occur in natural language.

But then, we may also consider more general contexts, where operators interact which are quantificational in some way. For instance, we consider the two adverbial modifiers in the complex verb

"often (e,t),(e,t))	walk (e,t)	a mile" (e,t),(e,t))
------------------------	---------------	-------------------------

There are two readings here which arise by iteration: often (walking a mile) versus (often walking) a certain mile. Is there also a truly complex one: say, like the *cumulative* reading of "many hands lifted eleven players", where all those hands *together* lifted the winning team of the Soccer League? The answer appears to be negative. And similar negative, or at least inconclusive outcomes arise in combinations such as

"write (e,(e,t))	five letters ((e,t),t)	to-day" ((e,t),(e,t))
---------------------	---------------------------	--------------------------

Here, the cumulative reading actually con-incides with one of the iterated ones. And, replacing "to-day" by a more quantificational expression will actually force us to make any intended cumulation morphologically visible:

"write	five letters	in three hours".
--------	--------------	------------------

Thus, the question as to linguistic generalizations of our initial situation remains open. An alternative remains, of course, to study the more general role of *compounding particles*, such as "in". Indeed, this would already be relevant for quantifiers themselves - since there too, non-iterative readings often involve such particles:

"three boys *together* ate all plums".

There are also more general *mathematical* questions raised by the earlier account of quantifiers. Quite generally,

Which items in some type of expression are already definable using only items from lower types?

As it stands, this question is still rather vague. But, it can be made more precise using suitable notions of 'definable' and 'lower'. (See van Benthem 1985, chapter XIX, for one particular general version.) Notably, it makes sense to think of definability as usual, by means of *applications* and *lambdas*.

Example: Reducible Noun Phrases.

Which items in type $((e,t),t)$ are definable using only items from the lower types (e,t) , e and t ? Consider any definition for such an item, possibly with parameters. Without loss of generality, the definition can be brought into a *lambda-normal-form*, leaving no more lambda-conversions to be performed. Moreover, types of variables occurring in the normal form must all be subtypes of $((e,t),t)$. Then, the following facts may be deduced:

it starts with $\lambda_{x_{(e,t)}}$, followed by an application with types (e,t) and e , or some constant of type t .

Thus, the only candidates are:

$\lambda_{x_{(e,t)}} \bullet x_{(e,t)}(a_e)$	(the 'lifted individual' a_e)
$\lambda_{x_{(e,t)}} \bullet b_{(e,t)}(a_e)$	(the empty and
$\lambda_{x_{(e,t)}} \bullet c_t$	the universal cases)

And, even adding *identity* to the defining language will add no new reducible items here. \square

In this general perspective, we can also return to the earlier issue of reducible polyadic quantifiers. By a simple calculation, the latter form a dwindling minority in the type $((e,(e,t)),t)$:

$$2^{2^n} \times 2^{2^n} \quad (= 2^{2^{n+1}}) \quad \text{versus} \quad 2^{2^{n^2}}$$

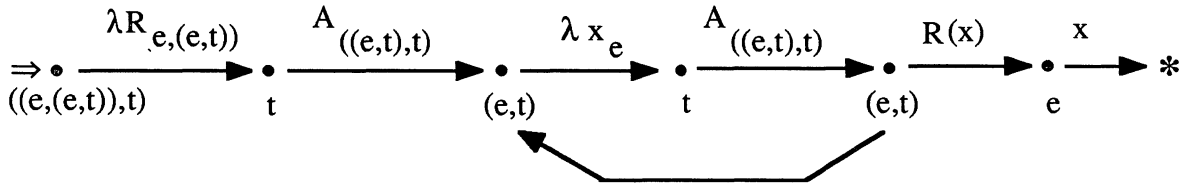
But, what if we allow the two parameters in type $((e,t),t)$ (i.e., the unary quantifiers involved in the reduction) to combine, not just via application, but with full lambda abstraction, as above? Then, in principle, there are *infinitely* many possibilities for schemes of definition. Still, there is a collapse to a fixed finite number of combination modes:

Proposition. Let a, b be two items in the type $((e,t),t)$ of some model. The items in the type $((e,(e,t)),t)$ which are lambda/application definable from these reduce to forms $\lambda R \bullet$ followed by a matrix in the following list:

- i $(\neg) A (\lambda_{x_e} \bullet (\neg) R (x)(x))$
- ii $(\neg) A (\lambda_{x_e} \bullet (\neg) A ((\neg) R (x)))$
- iii $(\neg) A (\lambda_{x_e} \bullet (\neg) A (\lambda_{y_e} \bullet (\neg) R (y)(x)))$.

Here, 'A' indicates either a or b , and ' (\neg) ' denotes an optional negation.

Proof. The argument again consists in an analysis of possible lambda normal forms. These may be described in the following *finite state machine*:



For further reference, we consider one particular example:

$$b (\lambda k \bullet a (\lambda y \bullet b (\lambda s \bullet a (\lambda x \bullet b (\lambda u \bullet a (\lambda z \bullet R (x)(y))...)))$$

There may be infinitely many of these forms; and so, we must establish a reduction to one of the cases listed above. The following sequence of steps illustrates the algorithm which effects this:

1. $a (\lambda z \bullet R (x) (y))$ is equivalent to

$$(R(x)(y) \wedge a (\lambda z \bullet \text{TRUE})) \vee (\neg R(x)(y) \wedge a (\lambda z \bullet \text{FALSE}))$$

Since both $a (\lambda z \bullet \text{TRUE})$ and $a (\lambda z \bullet \text{FALSE})$ have *fixed* truth values in our model, the above disjunction reduces to one of the following, *uniformly in R*:

$$R(x)(y), \quad \neg R(x)(y), \quad \text{TRUE} \quad \text{or} \quad \text{FALSE}.$$

Let us say $R(x)(y)$.

2. $b (\lambda u \bullet R(x)(y))$ again reduces to one of the forms

$$R(x)(y), \quad \neg R(x)(y), \quad \text{TRUE} \quad \text{or} \quad \text{FALSE};$$

by a similar argument.

3. Then the $\lambda x \bullet$ can attach to the first two terms, yielding $\lambda x \bullet R(x)(y)$ ($=\check{R}(y)$) or $\lambda x \bullet \neg R(x)(y)$ ($=\neg\check{R}(y)$), or again two constant cases.

4. Inserted into a , this gives one of

$$a(\check{R}(y)), \quad a(\neg\check{R}(y)), \quad \text{TRUE} \quad \text{or} \quad \text{FALSE}.$$

Now, apply similar reductions as above, distinguishing cases for $a(\check{R}(y))$, etcetera, to arrive at one of the forms mentioned in the theorem. ■

Thus, even with full lambda-definability, few polyadic quantifiers will be reducible.

There are obvious generalizations of this kind of analysis to other types. Even though these will not be undertaken here, the present Section may have shown the importance of a more general type-theoretical view of polyadic quantification, and the basic semantic issues raised by it.

Acknowledgement:

I would like to thank Sjaak de Mey for some stimulating discussions on these topics, in connection with his forthcoming dissertation on dyadic quantification and related linguistic phenomena.

IV References

1. J. Barwise and R. Cooper, 1981, 'Generalized Quantifiers and Natural Language', *Linguistics and Philosophy* 4, 159-219.
2. J. van Benthem, 1984, 'Questions about Quantifiers', *Journal of Symbolic Logic* 49, 443-466.
3. J. van Benthem, 1985, *Modal Logic and Classical Logic*, Bibliopolis/The Humanities Press, Napoli/Atlantic Heights.
4. J. van Benthem, 1986, *Essays in Logical Semantics*, Reidel, Dordrecht.
5. J. van Benthem, 1987, 'Categorial Grammar and the Theory of Types, ASL/LSA Colloquium, Stanford University. (To appear in *Linguistics and Philosophy*.)
6. P. Gärdenfors, ed, 1987, *Generalized Quantifiers: Linguistic and Logical Approaches*, Reidel, Dordrecht.
7. M. Dummett, 1973, *Frege, the Philosophy of Language*, Duckworth, London.
8. J. van Eyck, 1987, 'Quantifiers', in A. von Stechow and D. Wunderlich, eds, 1987.
9. J-E. Fenstad, P-K Halvorsen, T. Langholm and J. van Benthem, 1985, *Equations, Schemata and Situations*, report 85-29, Center for the Study of Language and Information, Stanford. (To appear with Reidel, Dordrecht, as *Situations, Language and Logic*.)
10. D. Gabbay and F. Guenther, eds, 1987, *Handbook of Philosophical Logic*, vol. IV, Reidel, Dordrecht, to appear.
11. J. Groenendijk, D. de Jongh and M. Stokhof, eds, 1987a, *Foundations of Pragmatics and Lexical Semantics*, Foris, Dordrecht.
12. J. Groenendijk, D. de Jongh and M. Stokhof, eds, 1987b, *Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers*, Foris, Dordrecht.
13. J. Higginbotham and R. May, 1981, 'Questions, Quantifiers and Crossing', *The Linguistic Review* 1, 41-79.
14. E. Keenan, 1987a, 'Non-Standard Noun Phrases and Semantically based Universals', paper presented at the 6th Amsterdam Colloquium, University of Amsterdam, April 1987. (To appear in Proceedings of that Colloquium.)
15. E. Keenan, 1987b, 'Unreducible n-ary Quantifiers in Natural Language', in P. Gärdenfors, ed, 1987, 109-150.
16. E. Keenan and Y. Stavi, 1986, 'A Semantic Characterization of Natural Language Determiners', *Linguistics and Philosophy* 9, 253-326. (First version from 1981.)

17. P. Lindström, 1966, 'First-Order Predicate Logic with Generalized Quantifiers', *Theoria* 32, 186-195.
18. S. Löbner, 1987, 'Quantification as a Major Module of Natural Language Semantics', in J. Groenendijk et al., eds, 1987b, 53-85.
19. R. May, 1987, 'Interpreting Logical Form', *Linguistics and Philosophy*, to appear.
20. J. de Mey, 1987, 'Dyadic Quantifiers', Instituut voor Algemene Taalwetenschap, Rijksuniversiteit, Groningen.
21. A. Mostowski, 1957, 'On a Generalization of Quantifiers', *Fundamenta Mathematicae* 44, 12-36.
22. A. von Stechow and D. Wunderlich, eds, 1987, *Handbook of Semantics*, De Gruyter, Berlin, to appear.
23. D. Westerståhl, 1986, *Quantifiers in Formal and Natural Languages*, report 86-55, Center for the Study of Language and Information, Stanford. (To appear in D. Gabbay and F. Guentner, eds, 1987.)
24. D. Westerståhl, 1986a, 'On the Order between Quantifiers', in *Logic and Abstraction. Essays dedicated to Per Lindström on his fiftieth birthday*, Acta Philosophica Gothoburgiensia I, University of Göteborg, 273-285.
25. E. Zimmermann, 1987, 'Transparent Adverbs and Scopeless Quantifiers', in J. Groenendijk et al., eds, 1987a, 81-99.
26. F. Zwarts, 1986, *Categoriale Grammatica en Algebraische Semantiek*, dissertation, Nederlands Instituut, Rijksuniversiteit, Groningen. (English version to appear with Foris or Reidel, Dordrecht.)