Normal Forms for Characteristic Functions on n-ary Relations

JAN VAN EIJCK, *Centre for Mathematics and Computer Science (CWI),
Amsterdam and Uil-OTS, Utrecht, The Netherlands.
E-mail: Jan.van.Eijck@cwi.nl*

Abstract

Functions of type $\langle n \rangle$ are characteristic functions on n-ary relations. Keenan established their importance for natural language semantics, by showing that natural language has many examples of irreducible type $\langle n \rangle$ functions, i.e. functions of type $\langle n \rangle$ that cannot be represented as compositions of unary functions. Keenan proposed some tests for reducibility, and Dekker improved on these by proposing an invariance condition that characterizes the functions with a reducible counterpart with the same behaviour on product relations. The present paper generalizes the notion of reducibility (a quantifier is reducible if it can be represented as a composition of quantifiers of lesser, but not necessarily unary, types), proposes a direct criterion for reducibility, and establishes a diamond theorem and a normal form theorem for reduction. These results are then used to show that every positive $\langle n \rangle$ function has a unique representation as a composition of positive irreducible functions, and to give an algorithm for finding this representation. With these formal tools it can be established that natural language has examples of n-ary quantificational expressions that cannot be reduced to any composition of quantifiers of lesser degree.

Keywords: Semantics of natural language, quantifier decomposition, Fregean versus non-Fregean quantifiers, polyadic quantification, compositionality.

1 Introduction

Instead of analysing the sentence *Every lawyer cheated a firm* as a relation between the CN property of being a lawyer and the VP property of cheating firms (namely the relation of inclusion), it is also possible to look at the complex expression *Every lawyer ___ a firm*, and interpret that as a function that takes a relation (a denotation of a transitive verb, such as *cheated*, *defended*) and produces a truth value. Similarly, *Every firm received a letter from some lawyer* can be analysed as stating that the set of firms is included in the set of letters received from some lawyer, but it is also possible to look at the complex expression *Every firm ___ a letter from some lawyer*, and even at *Every firm ___ a letter ___ some lawyer*. The interpretation of *Every firm ___ a letter from some lawyer* is again a function from binary relations to truth values, the interpretation of *Every firm ___ a letter ___ some lawyer* is a function from ternary relations to truth values.

This gives two ways to analyse *Every lawyer ___ a firm*: as a composition of the interpretation of *Every lawyer* with that of *a firm*, or, alternatively, as a function that classifies binary relations. In this case, the first analysis seems preferable, but in many cases only the alternative analysis is available. Consider *Every lawyer cheated a different firm*. This means that the relation of cheating, when restricted to the set of pairs (l, f) with l a lawyer and f a firm, is an injective function. There is no way to express this as a relation between an CN property (*being a lawyer*) and a VP property. Intuitively, *cheating a different firm* does not express a property.

If three noun phrases are present, as in *Every executive awarded himself a huge bonus*, the question arises how this should be analysed:

- As a complex quantifier *every executive ___ himself a huge bonus* that combines with a ternary

relation?

- As as composition of the interpretation of *every executive* with a complex quantifier for *himself a huge bonus*?

- As a compositition of a complex quantifier for *every executive* __*himself* and a quantifier for *a huge bonus*?

- As a composition of three quantifiers for *every executive*, *himself* and *a huge bonus*?

In cases with four noun phrases there are even more possibilities. This paper will give a full characterization of what is the simplest compositional analysis in every conceivable case.

2   Functions, types, lifting, decomposition

Following Keenan [5] we call a function from properties (unary relations) to truth values a type $\langle 1 \rangle$ function, a function from binary relations to truth values a type $\langle 2 \rangle$ function, and, in general, a function from n-ary relations to truth values a type $\langle n \rangle$ function. Note that a type $\langle n \rangle$ function is in fact a characteristic function on n-ary relations.

Let E be the domain of discourse. Let e be the type of an object in E, and let t be the type of truth values.

On the type t, we use $\top$ for truth and $\bot$ for falsehood, and we allow the usual Boolean functions for conjunction, disjunction and negation. We write $\wedge pq$ as $p \wedge q$, and similarly for disjunctions.

We will work with a higher order logic that allows higher order abstraction and application. Expressions and types look like this:

$$
\begin{aligned}
E &::= & x \mid (E_1 E_2) \mid \lambda x \cdot E \mid \lambda(x_1, \ldots, x_n) \cdot E \mid (E_1, \ldots, E_n) \\
T &::= & e \mid t \mid T_1 \rightarrow T_2 \mid T_1 \times \cdots \times T_n.
\end{aligned}
$$

These formation rules are constrained by a welltypedness criterion. We will use $E :: T$ to express that expression E has type T. The welltypedness rules are:

$$
\frac{E_1 :: T_1 \rightarrow T_2 \quad E_2 :: T_1}{(E_1 E_2) :: T_2}
$$

$$
\frac{x :: T_1 \quad E :: T_2}{\lambda x \cdot E :: T_1 \rightarrow T_2}
$$

$$
\frac{x_1 :: T_1 \quad \cdots \quad x_n :: T_n \quad E :: T_{n+1}}{\lambda(x_1, \ldots, x_n) \cdot E :: (T_1 \times \cdots \times T_n) \rightarrow T_{n+1}}
$$

$$
\frac{E_1 :: T_1 \quad \cdots \quad E_n :: T_n}{(E_1, \ldots, E_n) :: T_1 \times \cdots \times T_n}
$$

Note the following:

- $\langle 1 \rangle$ abbreviates the type $(e \rightarrow t) \rightarrow t$,
- $\langle 2 \rangle$ abbreviates the type $((e \times e) \rightarrow t) \rightarrow t$,
- $\langle n \rangle$ abbreviates the type $(\underbrace{(e \times \cdots \times e)}_{n \text{ times}} \rightarrow t) \rightarrow t$.

Using e^n for $\underbrace{(e \times \cdots \times e)}_{n \text{ times}}$, we can say that $\langle n \rangle$ abbreviates the type $(e^n \to t) \to t$. We will use $\mathbf{1}$ for functions of types $(e^n \to t) \to t$ (with $n \geq 1$) that yield *true* for any argument, i.e. $\mathbf{1}$ is the quantifier $\lambda R. \top$. Similarly, $\mathbf{0}$ is the quantifier $\lambda R. \bot$ (the quantifier that yields *false* for any argument).

If $R :: e^2 \to t$ then $R(a, x)$ has type t, and $\lambda x \cdot R(a, x)$ has type $e \to t$.

Sometimes set notation is more convenient than lambda notation. For example, $P \times Q$ is more readable than the equivalent lambda expression $\lambda(x, y) \cdot (Px \wedge Qy)$. For this reason we will occasionally switch back and forth between lambda notation and set notation: $\lambda x \cdot R(a, x)$ corresponds to the set $\{x \mid R(a, x)\}$. Also, characteristic functions will sometimes be applied to sets rather than the corresponding lambda expressions. So if f has type $(e \to t) \to t$, we will sometimes write $f(\{x \mid R(a, x)\}) = \top$ instead of $f(\lambda x \cdot R(a, x)) = \top$ to express that f classifies the set as true.

We will occasionally omit application parentheses, using the convention that application associates to the left. Thus, $E_1 \ E_2 \ E_3$ abbreviates $((E_1 \ E_2) \ E_3)$.

Abstraction over tuples can be used for currying and uncurrying of functions, as follows. If $R :: e^2 \to t$ and $x :: e, y :: e$, then:

$$\lambda R \lambda x \lambda y \cdot R(x, y) :: (e^2 \to t) \to (e \to e \to t).$$

This is the *currying* operation. If $R :: e \to e \to t$ and $x :: e, y :: e$, then:

$$\lambda R \lambda(x, y) \cdot Rxy :: (e \to e \to t) \to (e^2 \to t).$$

This is the *uncurrying* operation.

A type $\langle 1 \rangle$ function f on E can be lifted to a function $(\boldsymbol{L}^{(n+1),n} f)$ from $(n + 1)$-ary relations to n-ary relations by means of the following lift operator:

$$\boldsymbol{L}^{(n+1),n} = \lambda f \lambda R \lambda(x_1, \ldots, x_n) \cdot f(\lambda x \cdot R(x_1, \ldots, x_n, x)).$$

Note that if $f :: \langle n \rangle$ and $R :: e^{n+1} \to t$, $x_1 :: e, \ldots, x_n :: e, x :: e$ (i.e. R is an $(n + 1)$-ary relation and x_i, x are individual variables) then $(\boldsymbol{L}^{(n+1),n} f)$ is of the required type, i.e. $(\boldsymbol{L}^{(n+1),n} f) :: (e^{n+1} \to t) \to (e^n \to t)$.

Similarly, a type $\langle n \rangle$ function F can be lifted to a function $(\boldsymbol{L}^{(m+n),m} F)$ from $(m + n)$-ary relations to m-ary relations, by means of:

$$\boldsymbol{L}^{(m+n),m} =$$
$$\lambda F \lambda R \lambda(x_1, \ldots, x_m) \cdot F(\lambda(x_{m+1}, \ldots, x_{m+n}) \cdot R(x_1, \ldots, x_m, x_{m+1}, \ldots, x_{m+n})).$$

If $F :: \langle n \rangle$ and $R :: e^{m+n} \to t$, then $(\boldsymbol{L}^{(m+n),m} F) :: (e^{m+n} \to t) \to (e^m \to t)$, i.e. $(\boldsymbol{L}^{(m+n),m} F)$ maps $(m + n)$-ary relations to m-ary relations.

Lifted type $\langle 1 \rangle$ functions can then be composed by means of the following operation (assume $R :: e^{n+2} \to t$):

$$(\boldsymbol{L}^{(n+1),n} f) \circ (\boldsymbol{L}^{(n+2),(n+1)} g) = \lambda R \cdot ((\boldsymbol{L}^{(n+1),n} f)((\boldsymbol{L}^{(n+2),(n+1)} g)R)).$$

Note that $(\boldsymbol{L}^{(n+1),n} f) \circ (\boldsymbol{L}^{(n+2),(n+1)} g)$ maps $(n + 2)$-ary relations to n-ary relations, i.e. it is of type $(e^{n+2} \to t) \to (e^n \to t)$.

For the particular case of binary relations, we get, on the assumption that $R :: e^2 \to t$:

$$f \circ (\boldsymbol{L}^{2,1} g) = \lambda R \cdot f(\boldsymbol{L}^{2,1} \ g \ R).$$

If $F :: \langle 2 \rangle$ equals $f \circ (L^{2,1} g)$ for some $f :: \langle 1 \rangle, g :: \langle 1 \rangle$, we say that F can be *decomposed* into f and g, or that F *reduces* to a composition of f and g.

Clearly, many type $\langle 2 \rangle$ functions can be decomposed in this way into pairs of type $\langle 1 \rangle$ functions. For instance, the type $\langle 2 \rangle$ function F that interprets the complex expression *Every lawyer __ a firm* can be decomposed into a type $\langle 1 \rangle$ function g that interprets *a firm* and a type $\langle 1 \rangle$ function f that interprets *every lawyer*, for $f \circ (L^{2,1} g)$ equals F.

More generally, if $F :: \langle n \rangle$ equals

$$f_1 \circ (L^{2,1} f_2) \circ \cdots \circ (L^{n,(n-1)} f_n)$$

for some $f_1 :: \langle 1 \rangle, \ldots, f_n :: \langle 1 \rangle$, we say that F can be *decomposed* into (or *reduced* to) $f_1, \ldots, f_n$.

Thus, the function F of type $\langle 3 \rangle$ that interprets the quantification in

> *Every firm __ a letter __ some lawyer*

(on its natural scope reading) is a composition $f_1 \circ (L^{2,1} f_2) \circ (L^{3,2} f_3)$, where f_1 is the interpretation of *every firm*, f_2 is the interpretation of *a letter*, and f_3 is the interpretation of *some lawyer*.

In the rest of this paper, we will leave the lifting operations implicit. We will use $f_1 \circ f_2$ as shorthand for $f_1 \circ (L^{2,1} f_2)$, use $f_1 \circ f_2 \circ f_3$ as shorthand for $f_1 \circ (L^{2,1} f_2) \circ (L^{3,2} f_3)$, and so on.

More generally, if $F :: \langle m \rangle$ and $G :: \langle n \rangle$, then $F \circ G$ is shorthand for the function of type $\langle m + n \rangle$ that results from the following lift:

$$F \circ (L^{(m+n),m} G).$$

Spelled out in full, this is the following function (assume $R :: e^{m+n} \to t$):

$$\lambda R \cdot F(\lambda(x_1, \ldots, x_m) \cdot G(\lambda(x_{m+1}, \ldots, x_{m+n}) \cdot R(x_1, \ldots, x_m, x_{m+1}, \ldots, x_{m+n}))).$$

Call a function F of type $\langle n \rangle$ *positive* if $F(\emptyset) = \bot$, and *negative* otherwise. The interpretations of *some firm* and *every lawyer* are positive, those of *no lawyer* and *not every firm* are negative.

When studying compositions of functions $F \circ G$, we will always assume, without loss of generality, that G is positive: if not, one can simply replace G by $\neg G$ and F by $F \neg$. More precisely, if $F :: \langle n \rangle, G :: \langle m \rangle$, then:

$$\neg G = \lambda R \cdot \neg(GR)$$
$$F \neg = \lambda S \cdot F \, \lambda(x_1, \ldots, x_n) \cdot \neg(S(x_1, \ldots, x_n))).$$

Clearly, $F \circ G :: \langle m + n \rangle$ is the same function as $F \neg \circ \neg G :: \langle m + n \rangle$.

3   Failures of decomposition

In [5] it is demonstrated that there are cases where quantifiers of type $\langle 2 \rangle$ or higher are *not* decomposable. Keenan shows that the following sentence exhibits an example of non-decomposable type $\langle 2 \rangle$ quantification:

(3.1) Different students answered different questions.

For sentence (3.1) to make sense, we have to assume that there are at least two students. The sentence is true if there is a one-to-one correspondence between students and sets of questions they answered. Thus, *Different students __ different questions* is interpreted as the type $\langle 2 \rangle$ function expressing that its argument relation R satisfies the property that all the aR, with a ranging over students, are

different (here aR is used as shorthand for $\{x \mid R(a, x)\}$). Keenan has an ingenious method to prove this fact. He states and proves a theorem to the effect that for any two type $\langle 2 \rangle$ functions F, G that are reducible it holds that these functions are equal iff they act the same on Cartesian products, i.e. if for all subsets P, Q of the domain of discourse E it holds that $F(P \times Q) = G(P \times Q)$ (see Section 4 below).

How can this be used to show that a type $\langle 2 \rangle$ function F is non-reducible? Here is how, for the example of (3.1).

Let F be the type $\langle 2 \rangle$ function that interprets *different students __ different questions*. Assume that F is reducible.

Let S be the set of students and Q the set of questions. Assume there are at least two students and at least two questions, for otherwise statement (3.1) becomes trivial. Let $A \times B$ be a product relation, i.e. a relation that links every object in A to every object in B. If there are two students in $S - A$, then they bear $A \times B$ to the same questions, namely, no questions. If there are two students in $S \cap A$, then the questions they bear $A \times B$ to are again the same, namely $B \cap Q$. Again, $F(A \times B) = \perp$.

Recall that $\mathbf{0}$ is the type $\langle 1 \rangle$ function that is false for any argument. Then, by the above, $F(R) = (\mathbf{0} \circ \mathbf{0})(R)$ for any product relation R.

By Keenan's theorem, it follows from this F is equal to $\mathbf{0} \circ \mathbf{0}$. Contradiction, for obviously, F is different from the composition $\mathbf{0} \circ \mathbf{0}$, for F is true of $\{(s_1, q_1), (s_2, q_2)\}$ (with $s_1, s_2 \in S$ and $q_1, q_2 \in Q$), and $\mathbf{0} \circ \mathbf{0}$ is not. Thus, the assumption that F is reducible must be false. F is not reducible.

Here are some further examples of quantifiers that Keenan shows to be not reducible.

(3.2) Three boys in my class dated the same girl.

(3.3) All girls fancied the same boy.

(3.4) John criticized Bill and no-one else criticized anyone else.

(3.5) The women at the wedding all wore different hats.

(3.6) Every student criticized everyone but himself.

(3.7) The students criticized each other.

(3.8) Two detectives interviewed a total of twenty witnesses.

(3.9) The boys gave the same presents to the same girlfriends on the same occasions.

(3.10) Every student gave different answers to different questions.

Here are some example quantifiers, with their types. Restricted universal quantifier, type $\langle 1 \rangle$.

$$\mathbf{forall}_A \; \lambda x \cdot \phi(x) \; :\Leftrightarrow \; \forall x (Ax \Rightarrow \phi(x)).$$

Transitivity quantifier, type $\langle 2 \rangle$.

$$\mathbf{Tr} \; \lambda(x, y) \cdot \phi(x, y) \; :\Leftrightarrow \; \forall x \forall y (\phi(x, y) \Rightarrow \forall z (\phi(y, z) \Rightarrow \phi(x, z))).$$

Injectivity quantifier, type $\langle 2 \rangle$.

$$\mathbf{Inj} \; \lambda(x, y) \cdot \phi(x, y) \; :\Leftrightarrow \; \forall x \forall y (x \neq y \Rightarrow \exists u \exists v (\phi(x, u) \wedge \phi(y, v) \wedge u \neq v)).$$

Set injectivity quantifier, type $\langle 2 \rangle$.

$$\mathbf{INJ} \; \lambda(x, y) \cdot \phi(x, y) \; :\Leftrightarrow \; \forall x \forall y (x \neq y \Rightarrow \lambda u \cdot \phi(x, u) \neq \lambda v. \phi(y, v)).$$

The set injectivity quantifier captures the meaning of *Different students gave different answers*.

4   Crossing the Frege boundary

This section gives Keenan's theorem that underpins his method for establishing the irreducibility facts mentioned above, plus Dekker's generalization and Dekker's indirect criterion for irreducibility [3]. Section 5 proposes a direct criterion for (ir)reducibility.

Keenan [5] starts out from the following Fact about the behaviour of type $\langle 1 \rangle$ functions on products:

FACT 4.1 (Keenan)
Let f be a positive function of type $\langle 1 \rangle$ and let $P, Q \subseteq E$. Then:

$$f(P \times Q) = \begin{cases} P & \text{if } f(Q) = \top \\ \emptyset & \text{otherwise.} \end{cases}$$

PROOF. Let $f :: \langle 1 \rangle$ be a positive function. Let $P, Q \subseteq E$.
First assume $P = \emptyset$. Then $P \times Q = \emptyset$, and

$$\begin{aligned} f(P \times Q) &= \{d \in \emptyset \mid f\{d' \in Q \mid (d, d') \in \emptyset\} = \top\} \\ &= \emptyset. \end{aligned}$$

From $P = \emptyset$ and $f(P \times Q) = \emptyset$, the Fact follows directly.
Now assume $Q = \emptyset$. Again $P \times Q = \emptyset$, and $f(P \times Q) = \emptyset$. From the positivity of f we get $f(Q) = f(\emptyset) = \bot$, and the Fact follows.
Finally, assume $P \neq \emptyset, Q \neq \emptyset$. Now $P \times Q \neq \emptyset$, and:

$$\begin{aligned} f(P \times Q) &= \{d \in E \mid f\{d' \in E \mid (d, d') \in P \times Q\} = \top\} \\ &= \{d \in E \mid d \in P, f\{d' \in E \mid d' \in Q\} = \top\} \text{ because f positive} \\ &= \{d \in P \mid f(Q) = \top\} \\ &= \begin{cases} P & \text{if } f(Q) = \top \\ \emptyset & \text{otherwise.} \end{cases} \end{aligned}$$

∎

From this we get immediately:

FACT 4.2 (Keenan)
Let f, g be positive functions of type $\langle 1 \rangle$, and let $P, Q \subseteq E$. Then:

$$(g \circ f)(P \times Q) = \top \text{ iff } g(P) = \top \wedge f(Q) = \top.$$

Recall that a $\langle 2 \rangle$ function F is reducible if there are type $\langle 1 \rangle$ functions f, g with $F = f \circ g$.

THEOREM 4.3 (Keenan)
If F and G are reducible type $\langle 2 \rangle$ functions, then $F = G$ iff for all $P, Q \subseteq E$ it holds that $F(P \times Q) = G(P \times Q)$.

PROOF. If $F = G$ then their behaviour on products is the same.
For the other direction, assume F, G have the same behaviour on products.
First suppose F, G positive. Then, because of reducibility there are positive f_1, f_2, g_1, g_2 with $F = f_1 \circ f_2$ and $G = g_1 \circ g_2$. Because F, G act the same on products, using Fact 4.2 we see that $f_1 = g_1$ and $f_2 = g_2$. Thus $F = f_1 \circ f_2 = g_1 \circ g_2 = G$.

Now assume F, G negative. Then, because of reducibility there are f_1, f_2, g_1, g_2, with f_1, g_1 negative, f_2, g_2 positive, $F = f_1 \circ f_2$ and $G = g_1 \circ g_2$.

Clearly, if $f_2 = g_2$, then by Fact 4.2, $f_1 = g_1$, and $F = f_1 \circ f_2 = g_1 \circ g_2 = G$.

If $\exists Q : \bot = f_2(Q) \neq g_2(Q) = \top$, then for any P,

$$(f_1 \circ f_2)(P \times Q) = f_1(\emptyset) = \top = (g_1 \circ g_2)(P \times Q).$$

It follows that $f_1 = g_1 = \mathbf{1} :: \langle 1 \rangle$, and $F = G = \mathbf{1} :: \langle 2 \rangle$. ∎

Dekker [3] generalizes Fact 4.1 to Fact 4.4, and Theorem 4.3 to the case of reducing a type $\langle n \rangle$ function to n functions of type $\langle 1 \rangle$ (n-reducibility).

FACT 4.4 (Dekker)
Let $F = f_1 \circ \cdots \circ f_n$, with all f_i positive. Then for all $Q_i \subseteq E$:

$$F(Q_1 \times \cdots \times Q_n) = \top \Leftrightarrow \forall i (1 \leq i \leq n \Rightarrow f_i(Q_i) = \top).$$

PROOF. Suppose $F(Q_1 \times \cdots \times Q_n) = \top$. Assume there is a i with $f_i(Q_i) = \bot$. Without loss of generality we may assume that for $i < j \leq n$, $f_j(Q_j) = \top$. Then, from $F = f_1 \circ \cdots \circ f_n$, with $n - i$ applications of (an obvious generalization of) Fact 4.1:

$$F(Q_1 \times \cdots \times Q_n) = (f_1 \circ \cdots \circ f_i)(Q_1 \times \cdots \times Q_i).$$

From this, with $f_i(Q_i) = \bot$, and again Fact 4.1,

$$F(Q_1 \times \cdots \times Q_n) = (f_1 \circ \cdots \circ f_{i-1})(\emptyset),$$

and from this, by positivity of the f, $F(Q_1 \times \cdots \times Q_n) = F(\emptyset) = \bot$, and contradiction with the given about F.

For the other direction, assume $\forall i \, 1 \leq i \leq n$: $f_i(Q_i) = \top$. Then with n applications of Fact 4.1, $F(Q_1 \times \cdots \times Q_n) = \top$. ∎

THEOREM 4.5 (Dekker)
If F and G are positive n-reducible type $\langle n \rangle$ functions, then $F = G$ iff $\forall Q_1, \ldots, Q_n \subseteq E$: $F(Q_1 \times \cdots \times Q_n) = G(Q_1 \times \cdots \times Q_n)$.

PROOF. If F, G are the same, then they behave the same on products.

Conversely, assume $F = f_1 \circ \cdots \circ f_n$ and $G = g_1 \circ \cdots \circ g_n$, with all the f_i, g_i positive, and suppose F and G act the same on products. Then: $F(Q_1 \times \cdots \times Q_n) = 1$ iff (Fact 4.4) for all $i : 1 \leq i \leq n$ it holds that $f_i(Q_i) = 1$. Similarly for G. Since F and G act the same on products, the f_i must be equal to the g_i, whence $F = G$. ∎

Dekker also succeeds in finding suitable candidate type $\langle 1 \rangle$ functions for this reduction, provided a function satisfies the following condition of invariance.

DEFINITION 4.6 (Invariance, Dekker)
A type $\langle n \rangle$ function F is invariant if for all non-empty $Q_1, \ldots, Q_n \subseteq E$ with $F(Q_1 \times \cdots \times Q_i \times \cdots \times Q_n) = \bot$ the following holds: either for all non-empty Q_i':

$$F(Q_1 \times \cdots \times Q_{i-1} \times Q_i' \times Q_{i+1} \times \cdots \times Q_n) = \bot,$$

or for all non-empty Q_j' $(j \neq i)$:

$$F(Q_1' \times \cdots \times Q_{i-1}' \times Q_i \times Q_{i+1}' \times \cdots \times Q_n') = \bot.$$

The importance of invariance is that it gives us a means of defining positive functions $f_i :: \langle 1 \rangle$ (for $1 \leq i \leq n$) from a positive invariant function F of type $\langle n \rangle$. The recipe is this. To determine whether $f_i(Q) = \bot$, check whether F takes the value $\bot$ for arbitrary choices of the other argument places in the product

$$Q_1 \times \cdots \times Q_{i-1} \times Q \times Q_{i+1} \times \cdots \times Q_n.$$

THEOREM 4.7 (Dekker)
A positive type $\langle n \rangle$ function F is invariant iff F has a product equivalent n-reducible correlate G.

PROOF. Only if: Assume that F is invariant and positive. Then $g_1, \ldots, g_n$ can be defined by means of:

$$
\begin{aligned}
g_i(\emptyset) \quad &:= \quad \bot \\
g_i(Q \neq \emptyset) = \bot \quad &:\Leftrightarrow \quad \forall \text{ non-empty } Q_1, \ldots, Q_{i-1}, Q_{i+1}, \ldots, Q_n \subseteq E, \\
&\qquad F(Q_1 \times \cdots \times Q_{i-1} \times Q \times Q_{i+1} \times \cdots \times Q_n) = \bot.
\end{aligned}
$$

By definition, all g_i are positive. By invariance and positivity of F, $F(Q_1 \times \cdots \times Q_n) = \top$ iff all Q_i are non-empty and $g_1(Q_1) = \top \wedge \cdots \wedge g_n(Q_n) = \top$ iff (Fact 4.4)

$$(g_1 \circ \cdots \circ g_n)(Q_1 \times \cdots \times Q_n) = \top.$$

Conversely, assume that $G = g_1 \circ \cdots \circ g_n$ is a function that acts like F on products. Assume G and all g_i positive. Suppose $F(Q_1 \times \cdots \times Q_n) = \bot$. Then by the fact that F and G act the same on products:

$$(g_1 \circ \cdots \circ g_n)(Q_1 \times \cdots \times Q_n) = \bot.$$

Suppose $g_i(Q_i) = \bot$. Then, by Fact 4.4,

$$(g_1 \circ \cdots \circ g_n)(Q'_1 \times \cdots \times Q'_{i-1} \times Q_i \times Q'_{i+1} \times \cdots \times Q'_n) = \bot.$$

Thus,

$$F(Q'_1 \times \cdots \times Q'_{i-1} \times Q_i \times Q'_{i+1} \times \cdots \times Q'_n) = \bot.$$

Suppose on the other hand that $g_i(Q_i) = \top$. Then by Fact 4.4, there is a $j \neq i$ with $g_j(Q_j) = \bot$. In this case, for any Q'_i,

$$(g_1 \circ \cdots \circ g_n)(Q_1 \times \cdots \times Q_{i-1} \times Q'_i \times Q_{i+1} \times \cdots \times Q_n) = \bot,$$

i.e.

$$F(Q_1 \times \cdots \times Q_{i-1} \times Q'_i \times Q_{i+1} \times \cdots \times Q_n) = \bot.$$

It follows that F is invariant.   ∎

5   A direct criterion for reducibility

In this section we will show that if a positive function $F :: \langle n \rangle$ is n-reducible, then it is possible to give explicit definitions of positive functions f_i ($1 \leq i \leq n$) with $F = f_1 \circ \cdots \circ f_n$. For this, we first define what we mean by the *reduct* of a positive function F, and next show that F is n-reducible iff F its equal to its reduct.

DEFINITION 5.1 (Reduct)
The reduct $F^\bullet$ of a positive type $\langle n \rangle$ function F is defined as

$$F^\bullet = f_1 \circ \cdots \circ f_n,$$

with f_i given by:

$$
\begin{aligned}
f_i(\emptyset) \quad &:= \quad \bot \\
f_i(Q \neq \emptyset) = \top \quad &:\Leftrightarrow \quad \exists Q_1, \ldots, Q_{i-1}, Q_{i+1}, \ldots, Q_n \subseteq E, \\
&\qquad F(Q_1 \times \cdots \times Q_{i-1} \times Q \times Q_{i+1} \times \cdots \times Q_n) = \top.
\end{aligned}
$$

Clearly, each f_i is positive. This gives us a simple test for reducibility:

THEOREM 5.2
For all positive type $\langle n \rangle$ functions F: $F = F^\bullet$ iff F is reducible.

PROOF. Only if. Immediate, for $F^\bullet$ has the form $f_1 \circ \cdots \circ f_n$.

Conversely, suppose there are positive $g_1, \ldots, g_n$ with $F = g_1 \circ \cdots \circ g_n$. Let $F^\bullet = f_1 \circ \cdots \circ f_n$. We have to show that $g_1 \circ \cdots \circ g_n$ equals $f_1 \circ \cdots \circ f_n$. By Theorem 4.5 it is enough to show, for all $Q_1, \ldots, Q_n \subseteq E$:

$$(g_1 \circ \cdots \circ g_n)(Q_1 \times \cdots \times Q_n) = (f_1 \circ \cdots \circ f_n)(Q_1 \times \cdots \times Q_n).$$

We have:
$$F(Q_1 \times \cdots \times Q_n) = (g_1 \circ \cdots \circ g_n)(Q_1 \times \cdots \times Q_n) = \bot$$

iff (Fact 4.4)
$$\exists i (1 \leq i \leq n \wedge g_i(Q_i) = \bot)$$

iff (positivity of F)

$$
\begin{aligned}
&\forall Q_1', \ldots, Q_{i-1}', Q_{i+1}', \ldots, Q_n', \\
&\quad F(Q_1' \times \cdots \times Q_{i-1}' \times Q_i \times Q_{i+1}' \times \cdots \times Q_n) = F(\emptyset) = \bot
\end{aligned}
$$

iff (definition of f_i)
$$f_i(Q_i) = \bot$$

iff (Fact 4.4, definition of $F^\bullet$)
$$F^\bullet(Q_1 \times \cdots \times Q_n) = \bot.$$

∎

The test 'Is F equal to its reduct?' is easy to apply. An irreducibility argument based on it is different from the irreducibility reasoning proposed by Dekker, where the irreducibility of the symmetry function is deduced from the fact that the function is not invariant. In the case of invariant functions that are irreducible (such as the transitivity function), Dekker needs a different test. In our case, the test is the same for any function.

Take as an example the function F that characterizes the *symmetric* relations. Since this is a negative function (for the empty relation is symmetric), switch to $\neg F$ instead. To establish whether $\neg F$ is reducible, we must ask what $(\neg F)^\bullet$ looks like. $(\neg F)^\bullet = f \circ g$ with $f(P \neq \emptyset) = \top$ iff $\exists Q \subseteq E, Q \neq \emptyset$ with $(\neg F)(P \times Q) = \top$. Such Q surely exists, for $P \times Q$ is non-symmetric iff $P \neq Q$. So $f = \mathbf{1}$, the constant $\top$ function. By the same reasoning we see that $g = \mathbf{1}$. So $(\neg F)^\bullet = \mathbf{1} \circ \mathbf{1} \neq \neg F$, and therefore $\neg F$ is irreducible (and so is F).

Take the function G that characterizes transitive relations. Again, since this is a negative function (the empty relation is transitive), we switch to its negation $\neg G$. $(\neg G)^\bullet = f \circ g$, where $f(P \neq \emptyset) = \top$ iff $\exists Q \subseteq E, Q \neq \emptyset$ with $(\neg F)(P \times Q) = \top$. This is never possible, for any product relation is transitive, so $f = \mathbf{0}$. Similarly, $g = \mathbf{0}$, and $(\neg G)^\bullet = \mathbf{0} \circ \mathbf{0} \neq \neg G$, and therefore $\neg G$ is irreducible (and so is G).

6   Reduction on the far side

The fact that a function is not n-reducible does not mean that it is in its simplest possible form. The following definition allows us to discuss reduction on the far side of the Frege boundary.

DEFINITION 6.1 ($m - n$-reduction)
A function $\mathbf{F}$ of type $\langle m + n \rangle$ is (m, n)-reducible if there are functions G, H, of types $\langle m \rangle$ and $\langle n \rangle$ respectively, with $\mathbf{F} = G \circ H$.

This is a useful concept, because it allows stating and answering further questions about reducibility of functions. Take for instance the quantification pattern of (6.1).

(6.1) Every prosecutor charged the same suspects with the same crimes.

(6.2) No lawyer argued for the same treatment of suspects of the same offence.

These examples are certainly not 3-reducible, but they might well be $(1, 2)$-reducible, in which case they could be construed by composing a Fregean quantifier with a type $\langle 2 \rangle$ function.
    Or take the pattern in (6.3).

(6.3) The prosecutors assisted each other in asking for identical punishments for identical offences.

This pattern is certainly not 4-reducible, but it might well be $(2, 2)$-reducible. For a compositional treatment of quantification beyond the Frege boundary these issues are crucial.
    If $R \subseteq E^m$ and $S \subseteq E^n$, then $R \times S$ is the following $(m + n)$-ary relation:

$$\lambda(x_1, \ldots, x_m, x_{m+1}, \ldots, x_{m+n}) \cdot R(x_1, \ldots, x_m) \wedge S(x_{m+1}, \ldots, x_{m+n}).$$

Here is the corresponding set-theoretic expression:

$$\{(d_1, \ldots, d_m, d_{m+1}, \ldots, d_{m+n}) \in E^{m+n} \mid (d_1, \ldots, d_m) \in R, (d_{m+1} \ldots, d_{m+n}) \in S\}.$$

Thus, $R \times S$ consists of all $(m + n)$-tuples over E that result from concatenating a tuple in R with one in S.
    Fact 4.1 can be generalized as follows:

FACT 6.2
Let $\mathbf{F}$ be a positive function of type $\langle n \rangle$, and let $R \subseteq E^m, S \subseteq E^n$. Then:

$$\mathbf{F}(R \times S) = \begin{cases} R & \text{if } \mathbf{F}(S) = \top \\ \emptyset & \text{otherwise.} \end{cases}$$

The following generalizations are also straightforward:

THEOREM 6.3
If $\mathbf{F}$ and $\mathbf{G}$ are (m, n)-reducible functions of type $\langle m + n \rangle$, then $\mathbf{F} = \mathbf{G}$ holds iff $\mathbf{F}$ and $\mathbf{G}$ act the same on products $R \times S$ with $R \subseteq E^m$ and $S \subseteq E^n$.

THEOREM 6.4

Let $\mathbf{F} = G \circ H$, with $G :: \langle m \rangle$ and $H :: \langle n \rangle$ both positive. Then: $\mathbf{F}(R \times S) = \top$ iff $G(R) = \top$ and $H(S) = \top$.

PROOF. Only if. Assume $\mathbf{F}(R \times S) = \top$. Suppose $H(S) = \bot$. Then with Fact 6.2, $\mathbf{F}(R \times S) = G(\emptyset)$. Contradiction with the positivity of G. Suppose $H(S) = \top$. Then with Fact 6.2, $\mathbf{F}(R \times S) = G(R)$, and done.

   The other direction follows immediately from Fact 6.2.   ■

   The following definition will be our tool for characterizing the (m, n)-reducible functions.

DEFINITION 6.5 ($m - n$-reduct)

The (m, n)-reduct of a type $\langle m + n \rangle$ function $\mathbf{F}$ is the composition $G \circ H$, with G of type $\langle m \rangle$ and H of type $\langle n \rangle$, with G defined by

$$G(R) = \top :\Leftrightarrow \exists S \subseteq E^n, \mathbf{F}(R \times S) = \top$$

and H by

$$H(S) = \top :\Leftrightarrow \exists R \subseteq E^m, \mathbf{F}(R \times S) = \top.$$

   The notion of an (m, n)-reduct provides us with a direct criterion for (m, n)-reducibility:

THEOREM 6.6

A positive type $\langle m + n \rangle$ function $\mathbf{F}$ is equal to its own (m, n)-reduct iff $\mathbf{F}$ is (m, n)-reducible.

PROOF. Only if. Immediate, for the (m, n)-reduct has the form $G \circ H$, with G of type $\langle m \rangle$ and H of type $\langle n \rangle$.

   Conversely, assume $\mathbf{F} = K \circ M$, with both K and M positive, K of type $\langle m \rangle$, M of type $\langle n \rangle$. Let $G \circ H$ be the (m, n)-reduct of $\mathbf{F}$. We show that $K = G$ and $M = H$. For this, it is enough to show that it holds for all $R \subseteq E^m$ and $S \subseteq E^n$ that

$$(K \circ M)(R \times S) = (G \circ H)(R \times S).$$

Let $R \subseteq E^m, S \subseteq E^n$. Then

$$
\begin{array}{lll}
\mathbf{F}(R \times S) = \top & \text{iff} & (K \circ M)(R \times S) = \top \\
& \text{iff} & K(R) = \top \text{ and } M(S) = \top \\
& \text{iff} & G(R) = \top \text{ (for there is an } S \text{ with } \mathbf{F}(R \times S) = \top) \\
& & \text{and} \\
& & H(S) = \top \text{ (for there is an } R \text{ with } \mathbf{F}(R \times S) = \top) \\
& \text{iff} & (G \circ H)(R \times S) = \top.
\end{array}
$$

■

   We will show now that any positive type $\langle n \rangle$ function can be reduced in a unique manner to a composition of irreducible functions. For this, we need to establish confluence of the reduction process. This is stated in the following theorem.

THEOREM 6.7 (Diamond property)

If $\mathbf{F} = F \circ G = K \circ M$ with $F :: \langle m \rangle$ and $G :: \langle n \rangle$, $m < m'$, $K :: \langle m' \rangle$, $M :: \langle m + n - m' \rangle$, all of $\mathbf{F}, F, G, K, M$ positive, then there is a positive function $H :: \langle m' - m \rangle$ such that $\mathbf{F} = F \circ H \circ M$.

$$\begin{array}{ccc} \mathbf{F} & \longrightarrow & F \circ G \\ \downarrow & & \downarrow \\ K \circ M & \longrightarrow & F \circ H \circ M \end{array}$$

PROOF. Let $H \circ M'$ be the $(m'-m, m+n-m')$-reduct of G. Let $F' \circ H'$ be the $(m, m'-m)$-reduct of K. We show that $F = F'$, $H = H'$, $M = M'$ by showing that $F \circ H \circ M' = F' \circ H' \circ M$. For this, it is enough to show that the two compounds have the same values for products $R \times S \times T$, with $R \subseteq E^m, S \subseteq E^{m'-m}, T \subseteq E^{m+n-m'}$.

$$\begin{array}{ll} & (F \circ H \circ M')(R \times S \times T) = \top \\ \text{iff } (H, M' \text{ is reduct of } G) & (F \circ G)(R \times S \times T) = \top \\ \text{iff } (\mathbf{F} = F \circ G) & \mathbf{F}(R \times S \times T) = \top \\ \text{iff } (\mathbf{F} = K \circ M) & (K \circ M)(R \times S \times T) = \top \\ \text{iff } (F', H' \text{ is reduct of } K) & (F' \circ H' \circ M)(R \times S \times T) = \top. \end{array}$$

$\blacksquare$

THEOREM 6.8 (Normal form)
Every positive $\mathbf{F} :: \langle n \rangle$ is uniquely representable as

$$\mathbf{F} = F_1 \circ \cdots \circ F_k,$$

with F_i positive and irreducible for all $i : 1 \leq i \leq k$. Moreover, on finite domains E there exists an algorithm for finding this normal form $\mathrm{NF}(\mathbf{F})$.

PROOF. $\mathbf{F}$ is irreducible if for no k with $1 \leq k < n$, $\mathbf{F}$ equals its $(k, n-k)$ reduct. If $\mathbf{F}$ is irreducible, $\mathrm{NF}(\mathbf{F}) = \mathbf{F}$. Otherwise, find the smallest k for which $\mathbf{F}$ equals its $(k, n-k)$-reduct $F_1 \circ \mathbf{F}'$, and put $\mathrm{NF}(\mathbf{F}) = F_1 \circ \mathrm{NF}(\mathbf{F}')$. Then F_1 is irreducible by virtue of its definition. By the diamond theorem, the normal form is unique.

To find the normal form of $\mathbf{F}$ assuming that E is finite, note that for all k with $1 \leq k < n$, the equality test between $\mathbf{F}$ and its $(k, n-k)$ reduct is decidable. $\blacksquare$

Note that the n-reducible positive functions of type $\langle n \rangle$ are precisely the positive functions F for which $\mathrm{NF}(F) = F^{\bullet}$.

7   Application to natural language semantics

Let us look at some examples to see how all of this can be applied to natural language semantics.

(7.1) Some hermit forbade himself every pleasure.

Theorem 6.8 can be used to see that the type $\langle 3 \rangle$ quantifier in example (7.1) is $(2, 1)$ reducible, as follows. The sentence is true on domain E iff there exist $R \subseteq E^2$ and $Q \subseteq E$ with R a reflexive relation with at least one hermit in its domain, and Q a set containing every pleasure, such that $R \times Q \subseteq \mathrm{FORBID}$. This means that the type $\langle 3 \rangle$ quantifier in (7.1) is equal to its $(2, 1)$ reduct, so it is $(2, 1)$ reducible.

(7.2) Some hermit forbade some hermit every pleasure.

The quantifier in (7.1) is not $(1, 2)$ reducible, for its $(1, 2)$-reduct is (7.2), and this is not equivalent to (7.1). So the normal form of the quantification in (7.1) is:

$$(\lambda R \cdot ((\mathrm{dom}(R) \cap \mathrm{HERMIT}) \neq \emptyset \wedge \forall x \, R(x, x)) \circ \lambda Q \cdot \mathrm{PLEASURE} \subseteq Q)(\mathrm{FORBID}).$$

Next, look at example (3.10), repeated here as (7.3) for convenience.

(7.3) Every student gave different answers to different questions.

This is reducible to **forall**$_S \circ$ **Inj**, with **forall**$_S$ :: $\langle 1 \rangle$ and **Inj**$_{A \times Q}$ :: $\langle 2 \rangle$. In other words, the quantifier is $(1, 2)$-reducible. By Keenan's result, the quantifier from this example is not fully reducible. It follows from the diamond theorem that it cannot be $(2, 1)$-reducible.

We can also show that the type $\langle 3 \rangle$ quantifier of example (6.1) is neither $(2, 1)$ nor $(1, 2)$-reducible. Here is the example repeated for convenience.

(7.4) Every prosecutor charged the same suspects with the same crimes.

This is not $(1, 2)$-reducible, for its $(1, 2)$-reduct is equivalent to $(\mathbf{q} :: \langle 1 \rangle) \circ (\mathbf{1} :: \langle 2 \rangle)$, since it holds for every $Q \subseteq E$ and $R \subseteq E^2$ that $Q \times R$ is in the quantifier relation, for $Q \times R$ expresses that every p in Q is related to every (s, c) pair in R, so it is indeed the case that every p charges every s with the same crimes, namely the crimes in sR. Neither is it $(2, 1)$-reducible, for its $(2, 1)$-reduct is equivalent to $(\mathbf{1} :: \langle 2 \rangle) \circ (\mathbf{1} :: \langle 1 \rangle)$, since it holds for every $R \subset E^2$ and $Q \subseteq E$ that $R \times Q$ is in the quantifier relation, for $R \times Q$ expresses that every (p, s) pair in R is related to every c in Q, so if (p_1, s) and (p_2, s) both in R then p_1 and p_2 charge s with the same crimes, namely *all* crimes in Q. This establishes the following fact about natural language:

FACT 7.1
Natural languages can express type $\langle 3 \rangle$ quantifiers that cannot be reduced to any composition of lesser types.

The iterated 'same' construction can be used to generalize this fact.

(7.5) Every politician told the same lies to the same audiences on the same occasions.

(7.6) Every politician told the same variations on the same lies to the same audiences on the same occasions.

Examples like these show:

FACT 7.2
For all reasonable n, natural languages present examples of type $\langle n \rangle$ quantificational expressions that cannot be reduced to any composition of quantifiers of lesser degree.

Thanks to Ed Keenan for urging me to be explicit about these facts about natural language.

8   Related work

Keenan's first examples of irreducible type $\langle n \rangle$ quantifiers are from [4]; the treatment of Section 4 is based on [5]. Van Benthem [2] gives a characterization of the reducible type $\langle n \rangle$ quantifiers that satisfy (an appropriate version of) permutation: these are exactly the Boolean compounds of unary quantifiers. Paper [5] has updates with further examples of irreducible quantifiers: [6, 7].

Ben-Shalom [1] remarks that Keenan's methods do not allow to establish the reducibility of (8.1).

(8.1) Two students criticised themselves.

A Keenan-style argument would try to find a composition of two unary quantifiers that behave the same on products, and conclude from the fact that this composition is not equivalent to the original quantifier that the original quantifier is not reducible. As Ben-Shalom remarks, restricted to products, (8.1) is equivalent to (8.2) rather than to (8.3).

(8.2) Two students criticized the same two students.

(8.3) Two students criticized two students.

Since (8.2) is not an example of a composition of two type $\langle 1 \rangle$ quantifiers, a Keenan style argument does not get off the ground. Our argumentation for showing that (8.1) is irreducible remains intact, however. The reduct of (8.1) is (8.3), and from the fact that (8.1) and (8.3) are different it follows that (8.1) is irreducible.

Ben-Shalom, by the way, defines $\langle k \rangle$-reducibility of a function $F :: \langle n \rangle$ as follows: $F :: \langle n \rangle$ is $\langle k \rangle$-reducible if there is a function $f :: \langle n - k \rangle$ and a positive $g :: \langle k \rangle$ with $F = f \circ g$. The 'right-hand-side bias' in this definition is connected to the tree-based representation of n-ary relations that is at the core of Ben-Shalom's proof technique for irreducibility. This makes the definition less natural than the one we adopted: it misses (e.g.) the distinction between $(2, 1)$-reducibility and $\langle 1 \rangle$-reducibility (in our sense).

(8.4) The students answered the same questions on two exams.

(8.5) There were two exams where the students answered the same questions.

According to Ben-Shalom's definition, the type $\langle 3 \rangle$ function in (8.4) is (BS) $\langle 1 \rangle$-reducible, for it is a composition $F \circ q_2$ of the functionality quantifier $F :: \langle 2 \rangle$ and the quantifier $q_2 :: \langle 1 \rangle$. The type $\langle 3 \rangle$ function in (8.5), however, is (BS) $\langle 2 \rangle$-reducible but not (BS) $\langle 1 \rangle$-reducible, for it can be decomposed as $q_2 \circ F$.

Using our characterization of $(n - k, k)$-reducibility of type $\langle n \rangle$ functions, we can establish a link with Ben-Shalom's graphical invariance theorem, as follows: if $\mathbf{F} :: \langle n \rangle$ is $(n - k, k)$-reducible, then the $G :: \langle k \rangle$ given by

$$G(S) = \top :\Leftrightarrow \exists R \subseteq E^{n-k}, \ \mathbf{F}(R \times S) = \top$$

satisfies the 'replace tree' and 'delete tree' properties.

Acknowledgements

References

[1] Dorit Ben-Shalom. A tree characterization of generalized quantifier reducibility. In M. Kanazawa and C.J. Pinón, editors, *Dynamics, Polarity and Quantification*, number 48 in CSLI Lecture Notes, pages 147–171. CSLI, 1994.

[2] J. van Benthem. Polyadic quantifiers. *Linguistics and Philosophy*, 12(4):437–464, 1989.

[3] P. Dekker. Meanwhile, within the Frege boundary. *Linguistics and Philosophy*, 26:547–556, 2003.

[4] E. Keenan. Unredicible n-ary quantification in natural language. In P. Gärdenfors, editor, *Generalized Quantifiers, Linguistic and Logical Approaches*, pages 109–150. Reidel, Dordrecht, 1987.

[5] E. Keenan. Beyond the Frege boundary. *Linguistics and Philosophy*, 15(2):199–221, 1992.

[6] E Keenan. Further beyond the Frege boundary. In J. van der Does and J. van Eijck, editors, *Quantifiers, Logic, and Language*, number 54 in CSLI Lecture Notes, pages 179–201. CSLI Publications, Stanford, 1996.

[7] E. Keenan. Beyond the Frege boundary; II. `http://www.folli.uva.nl/CD/1999/library/pdf/Fr-bdr2c.pdf`, 1999.