APPENDIX

SAFE AND POLYNOMIAL

In this appendix the theorem will be presented which was announced in chapter 2, at the end of section 7. The theorem states that in an infinitely generated free algebra all safe operations are polynomially definable (free algebras are algebras which are isomorphic to a term algebra). Remind that $f : A_{s_1} \times \ldots \times A_{s_n} \rightarrow A_{s_{n+1}}$ is safe in $\Sigma$-algebra $\langle A, F \rangle$ if for every $\Sigma$-algebra $\langle D, G \rangle$ and every $h \in \text{Epi}(\langle A, F \rangle, \langle D, G \rangle)$ there is a unique $\hat{f} : D_{s_1} \times \ldots \times D_{s_n} \rightarrow D_{s_{n+1}}$ such that $h \in \text{Epi}(\langle A, F \cup \{\hat{f}\}, D, G \cup \{\hat{f}\} \rangle)$. The proof originates from F. Wiedijk (pers. comm.).

THEOREM. Let $A = \langle \{A_s \}_{s \in S}, (F_s)_{y \in c} \rangle$ be a free algebra, that has a generating set $(B_s)_{s \in S}$ where each $B_s$ is infinite. Let $f : A_{s_1} \times \ldots \times A_{s_n} \rightarrow A_{s_{n+1}}$ be a safe operator. Then $f$ is a polynomially definable over $A$.

HEURISTICS. First I will give some heuristic considerations, there after the theorem will be proved by proving two Lemmas.

Let us assume for the moment that the theorem holds and let us try to reconstruct from $f$ the polynomial $p$ that defines $f$. Let $\langle b_1, \ldots, b_n \rangle$ be a possible argument for $f$, where $b_1, \ldots, b_n$ are generators of $A$. There is a term $t \in T_{x,A}$ such that $t_A = f(b_1, \ldots, b_n)$. Since $A$ is free, this term is unique. Hence $t$ is obtained from the polynomial $p$ we are looking for, by means of substituting, for the respective variables in $p$ constants corresponding to $b_1, \ldots, b_n$. Term $t$ (probably) contains constants for $b_1, \ldots, b_n$, but it is not yet clear for any given occurrence of such a constant in $t$, whether it occurs in $p$ as parameter, or is due to substitution for a variable.

In order to decide in these matters, we consider the value of $f$ for generators $\langle c_1, \ldots, c_n \rangle$ which are different from $\langle b_1, \ldots, b_n \rangle$ and from the constants in $t$. Suppose that for term $u$ we have $u_A = f(c_1, \ldots, c_n)$. Then $u$ can also be obtained from $p$ by substituting constants. We already know that all constants in $p$ also occur in $t$. Since $c_1, \ldots, c_n$ do not occur in $t$, all their occurrences in $u$ are due to of their substitution for variables. So if we replace in $u$ all occurrences of (constants corresponding with) $c_1, \ldots, c_n$ by variables, we have found the polynomial $p$ we were looking for.

This idea is followed in the next lemma. We perform these steps and prove that the polynomial so obtained has the desired properties.
LEMMA 1. There is an infinite sequence \( (z_k)_{k=1,2,...} \) of disjoint \( n \)-tuples of generators of \( A \), and a polynomial \( p \) such that for each \( z_k \), \( f(z_k) = p(z_k) \).

PROOF. We define by induction a sequence \( (z_k)_{k=0,1,...} \). Let
\[ B_0, B_1, ..., B_n = B_0, B_1, ..., B_n \]
and take \( z_0 \in B_0 \times ... \times B_n \) arbitrarily. This \( n \)-tuple is used for the first attempt to reconstruct the polynomial \( p \) which corresponds with \( f \). Below I will define an infinite sequence of attempts to reconstruct \( p \), and there after it will be proved that from the second attempt on always the same polynomial will be found; this is the polynomial \( p \) we were looking for, as will be proven in lemma 2.

Assume that \( z_0, ..., z_k \) and \( p_0, ..., p_k \) are already defined. Then we obtain \( z_{k+1} \) and \( p_{k+1} \) as follows.

Let \( C_{k,s} \) be the set of generators of sort \( s \) which corresponds with constants in \( p_k \), and let \( \{ z_{k,s} \} \) be the set of components of \( z_k \) of sort \( s \). Define \( B_{k,s} = B_{k,s} / (C_{k,s} \cup \{ z_{k,s} \}) \), and let \( z_{k+1} \in B_{k+1} \times ... \times B_{k+1} \).

Since \( A \) is generated by \( (B_s)_{s \in S} \), \( f(z_{k+1}) \) can be represented by a term \( t^n \) with parameters from \( (B_s)_{s \in S} \) including \( z_{k+1} \). This term \( t \) can be expressed as a polynomial expression in \( z_{k+1} \), say \( p_{k+1}(z_{k+1}) \), where, moreover, no component of \( z_{k+1} \) occurs as parameter in \( p_{k+1} \). Since for all \( k \) and \( s \) the sets \( C_{k,s} \) and \( \{ z_{k,s} \} \) are finite, and \( B_{k,s} \) was infinite, it follows that \( B_{k+1,s} \) is infinite. Hence this construction can be repeated for all \( k \).

Next it will be proven that the polynomials \( p_1, p_2, ..., \) are identical, thus proving the theorem for the sequence \( z_0, z_1, ..., \) (note that \( p_0 \) and \( z_0 \) are not included). The basic idea of the proof is that we introduce for each \( k \) a homomorphism which maps \( z_k \) on \( z_0 \), and then apply the assumptions of the theorem.

Consider the mapping \( \hat{h} \) defined by
\[
\begin{cases}
\hat{h}(b) = b & \text{if } b \in (B_s / \{ z_{k,s} \}) \text{ for some } s \\
\hat{h}(z_k(i)) = z_0(i), & \text{where } z_k(i) \text{ is the } i\text{-th component of } z_k.
\end{cases}
\]

Since \( A \) is free, the mapping \( \hat{h} \) determines uniquely a homomorphism \( h : A_F \rightarrow \langle (B_s / \{ z_{k,s} \}) \rangle \). Moreover, \( h \) is an epimorphism since all generators of the 'range'–algebra occur in the range of \( h \). The polynomials \( p_k \) were chosen to contain no constants corresponding to components of \( z_k \), therefore \( h(p_k(z_k)) = p_k(h(z_k)) \) holds for all \( k \).

Since operator \( f \) is safe, there is a unique \( \hat{f} \) such that
\( h \in \text{Epi}(\langle A, F \cup \{f\} \rangle, \langle [B_s, \{z_{k_s}\}_{s \in S}], F \cup \{f \rangle \} ). \)

Now the following equalities hold:

I \quad h(f(z_k)) = \hat{f}(h(z_k)) = \hat{f}(z_0) = \hat{f}(h(z_0)) = h(h(z_0)) = h(p_0(z_0)) = p_0(h(z_0)) = p_0(z_0).

II \quad h(f(z_k)) = h(p_k(z_k)) = p_k(h(z_k)) = p_k(z_0).

From I and II follows \( p_0(z_0) = p_k(z_0). \) Analogously we can prove that \( p_1(z_1) = p_k(z_k). \)

Since \( A \) is free, there is a unique term \( t \) such that \( p_k(z_0) = t = p_0(z_0). \)

So if we replace the variables in \( p_k \) and \( p_0 \) by constants corresponding to the components of \( z_0 \), we obtain the same expression. From this, and the fact that no components of \( z_0 \) occur as constants in \( p_0 \), it follows that the constants in \( p_k \) consists of:

a1) all the constants in \( p_0 \)

a2) possibly some constants corresponding to components of \( z_0 \).

Analogously it follows that the constants in \( p_k \) consist of:

b1) all the constants in \( p_1 \)

b2) possibly some constants corresponding to components of \( z_k \).

We have chosen \( z_i \) in such a way that no constant in \( p_0 \) corresponds to a component of \( z_i \), and no component of \( z_0 \) equals a component of \( z_i \). So if \( p_k \) contained constants for components of \( z_i \), this would conflict with a1) and a2). Therefore we have to conclude that the constants in \( p_k \) are the same as the constants in \( p_1 \), and none of these, moreover, corresponds to components of \( z_i \). So for all \( k \geq 1 \) we have \( p_k \equiv p_1 \). Call this polynomial \( p \). Then \( f(z_k) = p(z_k) \) for all \( k \geq 1 \).

**Lemma 2.** Let \( p \) be the polynomial guaranteed by lemma 1. Then for all \( a_1 \in A_{\delta_1} \times \ldots \times A_{\delta_n} \), \( f(a) = p(a) \).

**Proof.** Let \( a = \langle a^{(i)}_1, \ldots, a^{(i)}_n \rangle \) and assume that \( a^{(i)} = t^{(i)}(b^{(i)}_1, \ldots, b^{(i)}_n) \), where \( t^{(i)} \) is a polynomial without constants, and the \( b^{(i)}_j \)'s are generators of \( A \). Assume moreover that \( f(a) = t \). Let \( z_k \) be the infinite sequence of disjoint \( n \)-tuples of generators given by lemma 1. Since there are only finitely many constants in \( t \) and finitely many \( b^{(i)}_j \)'s, there is an \( m \) such that the components of \( z_m \) are all different from the constants in \( t \) and the \( b^{(i)}_j \)'s.
Define

\[
\hat{h} \text{ by } \begin{cases} 
\hat{h}(b) = b & \text{if } b \in B_T / \{z_m \} \\
\hat{h}(z_m^{(i)}) = a^{(i)} & \text{where } z_m^{(i)} \text{ is the } i\text{-th component of } z_m.
\end{cases}
\]

This mapping \( \hat{h} \) defines an epimorphism \( h \in \text{Epi}(\langle A, F \rangle, \langle [B_s \setminus \{z_m \}]_{s \in S}, F \rangle) \).

Since \( f \) is safe, there is a unique operation \( \hat{f} \) such that

\[
h \in \text{Epi}(\langle A, F \cup \{f\} \rangle, \langle [B_s \setminus \{z_m \}]_{s \in S}, F \cup \{\hat{f}\} \rangle).
\]

Now the following equalities hold

\[
f(a^{(1)}, \ldots, a^{(n)}) = t_A \overset{1}{\Rightarrow} h(t_A) = hf(a^{(1)}, \ldots, a^{(n)}) =
\]

\[
h(f(t^{(1)}_b), \ldots, f(t^{(n)}_b)) = \hat{f}(h(t^{(1)}_b), \ldots, h(t^{(n)}_b)) =
\]

\[
h(h(z^{(1)}_m), \ldots, h(z^{(n)}_m)) = h(f(z^{(1)}_m), \ldots, z^{(n)}_m) =
\]

\[
h(p(z^{(1)}_m), \ldots, z^{(n)}_m)) = p(h(z^{(1)}_m), \ldots, z^{(2)}_m)) = p(a^{(1)}, \ldots, a^{(n)}).
\]

Equalities 1 and 2 hold since \( z_m \) has no components which occur in \( t \) or \( b \).

END LEMMAS.

From lemma 1 and lemma 2 the theorem follows

END THEOREM.
APPENDIX I

INDIVIDUAL CONCEPTS IN PTQ

In chapter 4 the syntax and semantics of the PTQ fragment were presented. The common nouns and intransitive verbs of the fragment were translated into constants which denote predicates on individual concepts. Reduction rules allowed us to replace them by predicates on individuals. What is then the benefit of using such concepts? The argument given in chapter 4 was based upon the artificial name Bigboss. In PTQ two less artificial examples are given as justification for the translation into predicates on individual concepts.

Consider the sentences (1) and (2).

(1) The temperature is ninety
(2) The temperature is rising.

A naive analysis of (1) and (2), using standard logic, might allow to conclude for (3).

(3) Ninety rises.

This would not be correct since intuitively sentence (3) does not follow from (1) and (2). So we have to provide some analysis not having this consequence. This example is known as the temperature paradox.

Montague's second example is a variation of the temperature paradox. Sentence (6) does not follow from sentences (4) and (5), whereas a naive analysis might implicate this.

(4) Every price is a number
(5) A price rises
(6) A number rises.

The solution of these problems is based upon the use of individual concepts. The idea of the solution is explained as follows. Imagine the situation that the price of oil is $ 40, and becomes $ 50. In this situation one might say:

(7) The price of oil changes.

By uttering (7) one does not intend to say that $ 40 changes, or that $ 50 changes. It is intended to express a property of the oil price, considered
as a function from moments of time to amounts of dollars. Therefore (7) could be translated into a formula expressing a property of an individual concept: the oil price concept. Formally spoken, prices are considered as functions from indices to numbers, and the same for temperatures. Numbers are considered as elements in $D_e$, so prices and temperatures are of type $<s,e>$ they are individual concepts.

The technical details of the solution of the temperature paradox can be illustrated by the treatment of sentence (1). The first step of its production is application of $S_5$ to be and ninety. This yields (8); the corresponding translation reduces to (9).

(8) be ninety
(9) $\lambda x [x = \text{ninety}].$

The next step is to combine (8) with term (10), which has (11) as translation.

(10) the temperature
(11) $\lambda x [\exists y [\text{temperature}(y) \leftrightarrow x = y] \land \forall p(x)].$

Since the meaning postulate for common nouns (MP2) does not hold for temperature, its translation (11) cannot be reduced to a formula with quantification over individuals. Combination of (8) with (11) according to $S_4$ yields sentence (1); the corresponding translation reduces to (12).

(12) $\exists x [\forall y [\text{temperature}(y) \leftrightarrow x = y] \land \forall x = \text{ninety}].$

The translation of sentences (2) and (3) are respectively (13) and (14).

(13) $\exists x [\forall y [\text{temperature}(y) \leftrightarrow x = y] \land \text{rise}(x)]$
(14) rise($\exists \text{ninety}$).

From (12) and (13) it does not follow that (14) is true.

Montague's treatment of the temperature paradox has been criticized for his analysis of the notion temperature. But there are examples of the same phenomenon which are not based upon temperatures (or prices). Several examples are given by LINK (1979) and LOEHRNER (1976). One of their examples is the German version of (15).

(15) The trainer changes.

On the reading that a certain club gets another trainer, it would not be correct to translate (15) by a formula which states that the property of
changing holds for a certain individual.

The temperature paradox (and related phenomena) explain why individual concepts are useful. But in most circumstances we want to reduce them to individuals. In the remainder of this appendix it will be investigated when such a reduction is allowed. First we will do so for translations of intransitive verbs, then for other verbs, and finally for translation of common nouns.

The only intransitive verbs in the PTQ fragment which do not express a property of an individual, but of an individual concept, are rise and change. Therefore we restrict our attention to those models of IL in which the constants corresponding with the other intransitive verbs are interpreted as expressing properties of individuals. This is expressed by the following meaning postulate.

1. Meaning Postulate 3

\[ \exists w x [\delta(x) \leftrightarrow [\forall y [\forall z \in \text{VAR} <s, e, t>] (\text{MeVAR} <s, e, t>)] \]

where \( w \in \text{VAR} <s, e, t> \) and \( \delta \) translates any member of \( B_1 \) other than rise or change.

1. END

This meaning postulate states that for all involved predicates on individual concepts there is for each index an equivalent predicate on individuals. This predicate is index dependent: the set of walkers now may differ from the set of walkers yesterday. MP3 expresses the existence of such an equivalent predicate by the existential quantification \( \exists M \). This \( M \) is of type \( <s, e, t> \) because variables get an index independent interpretation, and as argued before, the predicate on individuals corresponding with \( \delta \) has to be index dependent.

In chapter 4 section 2, the \( \delta^* \) notation was introduced as an abbreviation for \( \lambda u [\delta(u)] \), so as an abbreviation for those cases where it could be said that \( \delta \) was applied to an individual. The above meaning postulate says that for certain constants \( \delta \) the argument always is an individual, even if this is not apparent from the formula. Therefore it might be expected that MP3 allows us to introduce the \( \delta^* \) notation for those constants in all contexts. The following theorems allow us to replace a formula with an occurrence of \( \delta \) (where MP3 holds for \( \delta \)), by a formula with an occurrence of \( \delta^* \).
2. **THEOREM.** MP3 is equivalent with

\[ \vdash \square \delta(x) \leftrightarrow \delta_x x. \]

**PROOF part 1.** Suppose that MP3 holds, so \( \vdash \exists \mathcal{M} \square [\delta(x) \leftrightarrow [\mathcal{M}] \langle x \rangle]. \)

Then there is a \( g \) such that \( g \models \forall x \square [\delta(x) \leftrightarrow [\mathcal{M}] \langle x \rangle]. \)

Now for all \( g' \models g' \models \delta_x x = \lambda u [\delta'(\langle u \rangle)](x) = \delta'(\langle x \rangle) = [\mathcal{M}] \langle \lambda u \delta'(\langle u \rangle) \rangle = [\mathcal{M}] \langle x \rangle. \)

Consequently \( g' \models \delta_x x = \mathcal{M}. \)

So there is a \( g : g \models \forall x \square [\delta(x) \leftrightarrow \delta_x x]. \)

Since there are no free variables in this formula, we have

\[ \vdash \square [\delta(x) \leftrightarrow \delta_x x]. \]

**REMARK.** The following more direct approach is incorrect because the conditions for \( \lambda \)-conversion are not satisfied.

\[ g \models \delta_x x = \lambda u \delta(\langle u \rangle) x = \delta(\langle x \rangle) = [\mathcal{M}] \langle x \rangle. \]

**PROOF part 2.** Suppose \( \vdash \square [\delta(x) \leftrightarrow \delta_x x]. \)

Let \( g,i \) be arbitrary and define \( g' \models g \) by \( g'(\mathcal{M}) = [\lambda \delta(\langle u \rangle)] A, 1, g. \)

Then \( i, g' \models \delta(x) \leftrightarrow [\lambda \delta(\langle u \rangle)] \langle x \rangle \leftrightarrow [\lambda \delta(\langle x \rangle)] \leftrightarrow [\mathcal{M}] \langle x \rangle. \)

Since \( g,i \) were arbitrary, MP2 follows.

2. END

On the basis of this theorem, we have besides RR, another reduction rule introducing the *.

3. **Reduction rule 11**

Let be given an expression of the form \( \delta(x) \), where \( \delta \) is the translation of an intransitive verb other than \textit{rise} or \textit{change}.

Then replace \( \delta(x) \) by \( \delta_x x. \)

**CORRECTNESS PROOF**

Apply theorem proof.

3. END

Now we have two rules for the introduction of \( \delta_x : \text{RR}_3 \) and \( \text{RR}_{11} \). The one requires that the argument is of a certain form, the other that the function is of a certain nature. They have different conditions for
application, and none makes the other superfluous. In case both reduction rules are applicable, they yield the same result. It is not clear to me why MP3 is formulated as it is, and not directly in the form given in theorem 2.

For verbs of other categories there are related meaning postulates. For instance the transitive verb find should be interpreted as a relation between individuals. The meaning postulate for the transitive verbs were already given in chapter 4 (MP4). Exceptions to that meaning postulate were seek and conceive because these verbs do not express a relation between individuals. But also about these verbs something can be said in this respect. The first arguments have to be (intensions of) individuals: it is an individual that seeks, and not an individual concept. This is expressed by meaning postulate 5, that will be given below. For verbs of other categories a related postulate expresses that their subjects are not individual concepts, but individuals.

4. **Meaning postulate 5**

\[ \forall \exists \forall x [\delta(x,p) \leftrightarrow [^\vee]([^\vee]x)] \]

where \( \delta \in \{ \text{seek, conceive} \} \).

5. **Meaning postulate 6**

\[ \forall \exists \forall x [\delta(x,p) \leftrightarrow [^\vee]([^\vee]x)] \]

where \( \delta \in \{ \text{believe that, assert that} \} \).

6. **Meaning postulate 7**

\[ \forall \exists \forall x [\delta(x,p) \leftrightarrow [^\vee]([^\vee]x)] \]

where \( \delta \in \{ \text{try to, wish to} \} \).

6. END

These three meaning postulates do not give rise to new reduction rules because there are no generally accepted notations for the corresponding predicates with an individual as first argument.

The treatment of the temperature paradox was essentially based on the use of individual concepts.
This explains why all common nouns are translated into constants denoting predicates on individual concepts. Most common nouns express a predicate on individuals. This is formulated in a meaning postulate which I recall from chapter 4.

7. Meaning postulate 2

\[ \square [\delta(x) \rightarrow \exists u [x = \lambda u]] \]

where \( \delta \in \{ \text{man, woman, park, fish, pen, unicorn} \} \).

7. END

The meaning postulates for nouns and for verbs have a related aim: they both aim at excluding arbitrary individual concepts as argument and guaranteeing an individual as argument. So one might expect that there is a close relation between the consequences of the two meaning postulates. One might for instance expect that for nouns something holds like the formula in MP3. This is not the case, as is expressed in the following theorem.

8. THEOREM. Let \( \delta \in \mathcal{CN} : \langle \mathcal{S}, \mathcal{R} \rangle \).

Let (I) be the formula \( \square [\delta(x) \rightarrow \exists u [x = \lambda u]] \)

and (II) the formula \( \square [\delta(x) \leftrightarrow \delta(x)^{(\forall)}] \).

Then (i) (I) \( \not\vdash \) (II)

and (ii) (II) \( \not\vdash \) (I).

PROOF. (i) In chapter 3 we introduced the constant bigboss, which will be used here. Suppose that

\[ i_1 \vDash \text{bigboss} = \lambda \text{nixon} \quad \text{and} \quad i_2 \vDash \text{bigboss} = \lambda \text{bresnjev} \]

Then

\[ [\lambda \text{bigboss}] i_1 \vDash (i_2) = \lambda \text{bigboss} (i_1)(i_1) = \text{bigboss} (i_1) \]

bresnjev

and

\[ [\lambda \text{bigboss}] i_1 \vDash (i_1) = \lambda \text{bigboss} (i_1)(i_1) = \text{bigboss} (i_1) \]

nixon.

This means that \( \lambda \text{bigboss} \) is an expression of type \( \langle s, e \rangle \) which does not denote a constant function. Since \( [\lambda u] A, i, 1, x ; g (i_2) = [\lambda u] A, i, 1, x ; g (i_1) = \lambda g (u) \)
we have that for no \( g: g,i_1 \models \bigwedge u = \bigwedge^{\bigvee} \text{boss}. \) Suppose furthermore that \( \delta \) is a constant for which \( \text{MP}_1 \) holds, say \( \text{man} \).

Then (I) is satisfied.

So \( g,i_1 \models \text{man}(\bigwedge^{\bigvee} \text{boss}) \rightarrow \exists u[^u \bigwedge^{\bigvee} \text{boss} = \bigwedge u] \).

Due to the just proved property of \( \bigwedge^{\bigvee} \text{boss} \), the consequence is never true.

So for no \( g \) \( g,i_1 \models \text{man}(\bigwedge^{\bigvee} \text{boss}) \).

Suppose moreover that the predicate \( \text{man}_x \) holds for \( \text{Nixon} \).

So \( g,i_1 \models \text{man}_x(\text{Nixon}) \).

Since \( g,i_1 \models \bigwedge^{\bigvee} \text{boss} = \text{Nixon} \)

we have \( g,i_1 \models \text{man}_x(\bigwedge^{\bigvee} \text{boss}) \). Consequently

for no \( g \) \( g,i_1 \models \text{man}(\bigwedge^{\bigvee} \text{boss}) \leftrightarrow \text{man}_x(\bigwedge^{\bigvee} \text{boss}) \).

So if \( g(x) = [\bigwedge^{\bigvee} \text{boss}]^A,i_1,g \) statement (II) is not true. Finally, note that it is easy to design a model in which \( \text{boss} \) and \( \text{man} \) have the assumed properties. Hence we have proven (I).

PROOF. (ii) Let \( \text{boss} \) be as above. Assume now that \( \delta \) is a constant for which \( \text{MP}_3 \) holds, say \( \text{walk} \). Then (II) holds for \( \delta \).

Suppose now \( i_1 \models \text{walk}_x(\text{Nixon}) \).

So \( i_1 \models \text{walk}_x(\bigwedge^{\bigvee} \text{boss}) \).

Let \( g(x) = [\bigwedge^{\bigvee} \text{boss}]^A,i_1,g \).

Then it is not true that

\[
\text{i} \not\models \text{walk}_x(x) \rightarrow \exists u[x = \bigwedge u]
\]

because the antecedence is true, whereas the consequence is false. A model in which \( \text{walk} \) and \( \text{boss} \) have the desired properties can easily be defined and that is a counterexample to the implication.

8. END

Consequences of the above theorem are:

1. The formulations of the meaning postulates for Common Nouns and for Intransitive Verbs cannot be transformed into each other.

2. The following statement from PTQ (MONTAGUE 1973, p.265,+19) is incorrect:
\[ \square [\delta(x) \rightarrow \delta_x(\bigvee x)] \text{ if } \delta \text{ translates a basic common noun other than } \text{price} \text{ or } \text{temperature} \]

3. The meaning postulate for common nouns does not allow for replacing in all contexts an individual concept variable by the extension of this variable. This result was independently found by LINK (1979, p.224).
Next I will prove that in certain contexts the meaning postulate for common nouns does allow us to replace bound variables of type \(<s,e>\) by variables of type \(e\). It is contexts created by these translation rules of the PTQ-fragment: the translation rule for the determiner CN and the determiner-CN-rel.clause constructions. In the sequel \(\delta\) stands for the translation of a CN for which \(MP_2\) holds, and \(\phi\) for the translation of a relative clause. This \(\phi\) may be omitted in the formulation of the theorems.

9. **Lemma.** If \(A,i,g \models \forall \psi\) then \(A,i,g \models \forall u[\wedge u/x]\psi\)

If \(A,i,g \models \exists u[\wedge u/x]\psi\) then \(A,i,g \models \exists \psi\).

**Proof.** \(\{m \in D_{<s,e>}[m = [\wedge u]A,i,g] \subset \{m \in D_{<s,e>}[m = xA,i,g]\}\).

9. **End**

The next theorem deals with terms in which the determiner is \(a\).

10. **Theorem.** \(A,i,g \models \exists x[\delta(x) \land \phi \land \forall P(x)]\)

If \(A,i,g \models \exists u[\delta(u) \land [\wedge u/x] \phi \land \forall [\wedge u]]\).

**Proof.** Suppose that

1. \(A,i,g \models \exists x[\delta(x) \land \phi \land \forall P(x)]\).

Then there is a \(m \in D_{<s,e>}\) such that

2. \(A,i,[x+m]g \models \delta(x) \land \phi \land \forall P(x)\).

From \(MP_2\) and (2) follows

3. \(A,i,[x+m]g \models \exists u[\wedge u]\).

So there is a \(a \in D_e\) such that

4. \(A,i,[x+m,u+a]g \models x = \wedge u\).

From (4) and (2) follows

5. \(A,i,[x+m,u+a]g \models \delta(\wedge u) \land [\wedge u/x] \phi \land \forall [\wedge u]\).
So

\[(6) \quad \text{A}, i, g, g \models \exists u[\delta(u) \wedge \left[ ^u / x \right] \phi \wedge \forall P(u)].\]

Reversely (6) implies (1), as follows from the above lemma.

10. END

The terms with determiner \textit{every} are dealt with in theorem 11.

11. THEOREM. \textit{A}, i, g \models \forall x[\delta(x) \wedge \phi \rightarrow \forall P(x)]

\[\iff \text{A}, i, g \models \forall u[\delta(u) \wedge \left[ ^u / x \right] \phi \rightarrow \forall P(u)].\]

PROOF. One direction of the theorem follows immediately from Lemma 9.
The other direction is proved by contra-position. Assume that was not true that

\[(1) \quad \text{A}, i, g \models \forall x[\delta(x) \wedge \phi \rightarrow \forall P(x)].\]

This means

\[(2) \quad \text{A}, i, g \models \neg \forall x[\delta(x) \wedge \phi \rightarrow \forall P(x)].\]

This is equivalent with

\[(3) \quad \text{A}, i, g \models \exists x[\delta(x) \wedge \phi \wedge \neg \forall P(x)].\]

Application of the argumentation of theorem 10 gives

\[(4) \quad \text{A}, i, g \models \exists u[\delta(u) \wedge \left[ ^u / x \right] \phi \wedge \neg \forall P(u)].\]

Therefore it is not true that

\[(5) \quad \text{A}, i, g \models \forall u[\delta(u) \wedge \left[ ^u / x \right] \phi \rightarrow \forall P(u)].\]

So (5) implicates (1).

11. END

The next two theorems deal with terms with determiner \textit{the}. 
12. **Theorem.** If \( A, i, g \models \exists y[\forall x[\delta(x) \land \phi] \iff x = y] \land \lor p(y) \). Then \( A, i, g \models \exists u[\forall v[\delta'(\lor v) \land [\lor v/x]v] \iff v = u] \land \lor p(u) \). 

**Proof.** Suppose

(1) \( A, i, g \models \exists y[\forall x[\delta(x) \land \phi] \iff x = y] \land \lor p(y) \). 

This means that there is a \( m \in D_{<b, e>} \) such that (2) and (3) hold

(2) \( A, i, [y = m] g \models \forall x[\delta(x) \land \phi] \iff x = y \). 

(3) \( A, i, [y = m] g \models \lor p(y) \).

From (2) follows (4), and therefore (5) holds.

(4) \( A, i, [y = m] g \models \delta(y) \land [y/x] \phi \iff y = y \).

(5) \( A, i, [y = m] g \models \delta(y) \land [y/x] \phi \).

From (5) and \( MP_2 \) follows that there is an \( a \in D_e \) such that (6)

(6) \( A, i, [y = m, u = a] g \models y = \lor u \).

From (3) and (6) follows (7)

(7) \( A, i, [y = m, u = a] g \models \lor p(\lor u) \).

Apply lemma 9 to (2) and substitute \( \lor v \) for \( y \). Since (6) holds it follows that (8) holds

(8) \( A, i, [y = m, u = a] g \models \forall v[\delta(\lor v) \land [\lor v/y] \phi \iff v = \lor u] \).

Since \([u = v] A, i, g \) equals \( [u = v] A, i, g \), we may replace in (8) \( \lor v = \lor u \) by \( v = u \). Combination of (8) with (7) yields (9)

(9) \( A, i, [y = m, u = a] g \models \forall v[\delta(\lor v) \land [\lor v/y] \phi \iff v = u] \land \lor p(\lor u) \).
From this the theorem follows.

12. END

13. **THEOREM.** If $A, i, g \models \exists v[\forall u[\delta(u) \land [^u/x]\phi] \leftrightarrow u=v] \land \check{v} \land P(^v)$

then $A, i, g \models \exists y[\forall x[\delta(x) \land \phi] \leftrightarrow x=y] \land \check{v} \land P(x)$.

**PROOF.** Suppose

(1) $A, i, g \models \exists v[\forall u[\delta(u) \land [^u/x]\phi] \leftrightarrow u=v] \land \check{v} \land P(^v)$.

Then there is an $a \in D_e$ such that (2) and (3) hold

(2) $A, i, [\nu\cdot a] g \models \forall u[\delta(u) \land [^u/x]\phi] \leftrightarrow u=v$]

(3) $A, i, [\nu\cdot a] g \models \check{v} \land P(^v)$.

Let $m \in D_{<s,e>}$ be such that (4) holds.

(4) $A, i, [x\cdot m] g \models \delta(x) \land \phi$.

Then from MP$_2$ follows that there is a $b$ such that

(5) $A, i, [x\cdot m, u\cdot b] g \models x = ^u$.

From (5) and (2) follows (6)

(6) $A, i, [x\cdot m, \nu\cdot a, u\cdot b] g \models \delta(x) \land \phi \leftrightarrow x = ^v$.

Since (4) holds, it follows from (6).

(7) $A, i, [x\cdot m, \nu\cdot a] g \models x = ^v$.

It follows from (4) and (7) that (8) holds

(8) $A, i, [\nu\cdot a] g \models \forall x[\delta(x) \land \phi \rightarrow x = ^v]$.

Let now $m \in D_{<s,e>}$ be such that (9) holds
From (2) it then follows that (10) holds

(10) \[ A, i, [\nu^\rightarrow, x^\rightarrow]g \models \delta(x) \land \phi. \]

From (9) and (10) follows (11)

(11) \[ A, i, [\nu^\rightarrow]g \models \forall x[x^\nu \rightarrow \delta(x) \land \phi]. \]

From (3), (8) and (11) the theorem follows.

13. END

The above theorems constitute the justification for the following reduction rule.

14. REDUCTION RULE 12

Let \( \delta \) be the translation of a common noun for which meaning postulate \( MP_2 \) holds. Let be given a formula of one of the following forms,

- \( \exists x[\delta(x) \land \phi \land \forall P(x)] \)
- \( \forall x[\delta(x) \land \phi \rightarrow \forall P(x)] \)
- \( \exists y[\forall x[\delta(x) \land \phi \leftrightarrow x=y] \land \forall P(y)] \).

Then replace this formula by respectively

- \( \exists u[\delta^u \land [\hat{u}/x]\phi \land \forall P^u] \)
- \( \forall u[\delta^u \land [\hat{u}/x]\phi \rightarrow \forall P^u] \)
- \( \exists v[\forall u[\delta^u \land [\hat{u}/x]\phi \leftrightarrow u=v] \land \forall P^v] \)

(provided that \( \phi \) does not contain a free occurrence of \( u \) or \( v \)).

CORRECTNESS PROOF
See the theorems.

14. END

The theorems mentioned above, allow us to change the types of bound variables in a lot of contexts which arise if one deals with sentences from the PTQ-fragment. But they do not cover all contexts arising in this fragment. If the rule of quantification into a CN phrase (i.e. $S_{15,n}$) is used, then no reduction rule is applicable. An example is (14) in the reading in which every has wider scope than a. The corresponding translation is (15), and although none of the reduction rules is applicable, it is equivalent with (16).

(14) Every man such that he loses a pen such that he finds it, runs.

(15) $\forall x[\exists u[\text{pen}_x(u) \land \text{man}(x) \land \text{lose}_x(\forall y, u) \land \text{find}_x(\forall y, u)] \rightarrow \text{run}_x(\forall y)]$

(16) $\forall v[\exists u[\text{pen}_v(u) \land \text{man}_v(v) \land \text{lose}_v(v, u) \land \text{find}_v(v, u)] \rightarrow \text{run}_v(v)]$.

One would like to have a reduction rule which is applicable to constructions in which quantification into a CN is used. However, not in all such contexts reduction is possible. This was discovered by FRIEDMAN & WARREN (1980a). Consider sentence (17).

(17) A unicorn such that every woman loves it changes.

Suppose that 17 is obtained by quantification of every woman into unicorn such that he loves it. Then the translation of (17) reduces to (18); Friedman & Warren call this 'a rather unusual reading'.

(18) $\exists x[\forall u[\text{woman}_x(u) \rightarrow \text{unicorn}(x) \land \text{love}_x(u, \forall z) \land \text{change}(x)]]$.

This translation is, however not equivalent with (19).

(19) $\exists v[\forall u[\text{woman}_v(u) \rightarrow \text{unicorn}_v(v) \land \text{love}_v(u, v) \land \text{change}_v(v)]]$.

This situation might rise doubts about rule $S_{15,n}$. However, see chapter 9, section 7.2 for an example where the rule is seeded.
APPENDIX 2

SET MANIPULATION IN SYNTAX

In chapter 6 I provided a system of categories for dealing with syntactic variables. The rules given there implicitly assume that the reader knows what sets are, and what $u$, $\text{with}$ and $-$ mean. This is set theoretical knowledge, and not knowledge of the grammatical system. In the present section I will formulate syntactic rules which allow for replacing expressions like $(1,2) \cdot 1$ by $(2)$ and $(1,2) \cup 3$ by $(1,2,3)$. So we aim at rules which remove the symbols $u$, $\text{with}$ and $-$ from the formulation of the rules. The collection of rules performing this task is rather complex. I wish to emphasize that the rules do not arise from the requirement of using total rules, but from grammatical formalism. A related situation would arise when using partial rules. Such rules would mention a condition like 'contains an occurrence of $he_n$'. Since 'containing' is not a notion defined by grammatical means, a formalist might wish to do so. Then rules are needed which are related to the rules below since they have to perform related tasks. Once it is shown that the set-theoretical notions $u$, $\text{with}$ and $-$ are definable by means of grammatical tools, there is no objection against using them in the grammar even when not explicitly defined in this way.

Let $G$ be a grammar with a collection hyperrules $H$, and let the elements of $H$ contain expressions like $set - n$. Then the actual rules of the grammar are defined as the result of performing the following actions in order.

1. replace the metavariables in the hyperrules by some terminal metaproduction of the meta-grammar
2. replace subexpressions in the rules by other expressions, according to the rules given below, until there are no occurrences of the non-acceptable symbols ($U$, $\text{with}$, $-$).

The rules eliminating the non acceptable symbols introduce some non-acceptable symbols themselves. These are $\ast$, $is$, $\underline{unless}$, $\underline{true}$ and $\underline{false}$; these symbols have to be added to those mentioned in point 2 above. The collection of rules performing the task of eliminating these symbols is infinite, and will be defined by means of a two-level grammar. The hyperrules describing the elimination of the unacceptable symbols are unrestricted rewriting rules with metavariables. These variables are mentioned below, together with some examples of their terminal productions. Different examples are
separated by a bar symbol: /, and ε denotes the empty string.

\[ \text{set} : \{1,2\} / \{3,1\} / \emptyset. \]
\[ \text{seq} : \ 1,2 / 3,1. \]
\[ \text{lseq} : \ 1 , / 3,1 , / \varepsilon. \]
\[ \text{rseq} : \ ,2 , / ,1,5 , / \varepsilon. \]
\[ n : \ 1 , / 5. \]

The metarules for these metavariables are as follows (again a bar / separates alternatives, the non-terminal symbols are in italics).

\[ \text{set} \rightarrow \text{seq} / \emptyset. \]
\[ \text{seq} \rightarrow n / n, \text{seq}. \]
\[ n \rightarrow 1/2/3/4/5/6/7/8/9/nn/n0. \]
\[ \text{lseq} \rightarrow \text{seq}, / \varepsilon. \]
\[ \text{rseq} \rightarrow ,\text{seq} / \varepsilon. \]

The rules for \textbf{with} have to allow for replacing \{1,2\} \textbf{with} 1 by \{1,2\}, whereas they should not allow for replacing \{2,3\} \textbf{with} 1 by \{2,3\}. The hyperrule describing such replacements is

\[ \{\text{lseq},n,rseq\} \textbf{with} n \rightarrow \{\text{lseq},n,rseq\}. \]

An example of a rule derived from this hyperrule is

\[ \{1,3,5\} \textbf{with} 3 \rightarrow \{1,3,5\}. \]

Thus the expression \textbf{with} 3 is eliminated. In case one meets the subexpression \{2,3\} \textbf{with} 1 there is no rule which can be obtained from this hyperrule and which can be applied to this subexpression. So we cannot get rid of the non-acceptable symbol \textbf{with}, as was required in point 2. So we do not obtain an actual rule and the derivation cannot continue. This 'blind alley' technique is due to SINTZOFF (1967).

The rules for eliminating the - sign have to replace \{1,2\} - 2 by \{1\}, and \{1,2\} - 3 by \{1,2\}. The rules have to check whether the number preceded by the - sign occurs in the set mentioned before the sign. For this purpose, we need grammatical means to check whether two numbers are equal or different. It is easy to design a rule which can be applied only if two numbers are equal: a hyperrule with two occurrences of the meta-variable \(n\) can be transformed into a real rule only by substituting for both occurrences
the same number. If a hyperrule contains metavariables \( n_j \) and \( n_k \), then it can be transformed into a real rule by substituting for \( n_j \) and \( n_k \) different numbers. But nothing prevents us to substitute the same number. It is difficult to guarantee that two numbers are different, but we need such rules. The rules which do so use the blind alley technique again, now on the symbol unless. The hyperrules are as follows.

\[
\begin{align*}
0 \text{ is } 0 & \rightarrow \text{true} & 1 \text{ is } 0 & \rightarrow \text{false} & 2 \text{ is } 0 & \rightarrow \text{false} & \cdots \\
0 \text{ is } 1 & \rightarrow \text{false} & 1 \text{ is } 1 & \rightarrow \text{true} & 2 \text{ is } 1 & \rightarrow \text{false} \\
& \vdots & & \vdots & & \vdots \\
0 \text{ is } 9 & \rightarrow \text{false} & 1 \text{ is } 9 & \rightarrow \text{false} \\
1n_j \text{ is } 1n_k \rightarrow n_j \text{ is } n_k & \rightarrow \text{false} & 2n_j \text{ is } 1n_k \rightarrow \text{false} \\
1n_j \text{ is } 2n_k \rightarrow \text{false} & \rightarrow \text{false} & 2n_j \text{ is } 2n_k \rightarrow n_j \text{ is } n_k \\
& \vdots & & \vdots \\
1n_j \text{ is } 9n_k \rightarrow \text{false} & \rightarrow \text{false} & \text{unless } true \rightarrow false \\
\text{unless false } \rightarrow \text{true} \\
\text{true } \rightarrow \varepsilon.
\end{align*}
\]

The above rules have the effect that an expression of the form unless \( a \) is \( b \) reduces to unless true and next to unless in case \( a \) equals \( b \). This unless constitutes a blind alley. If \( a \) is not equal to \( b \), the expression reduces to unless false and through true to the empty string. Then the test is eliminated, and the production may proceed.

The rules for the - sign have to check for all elements of the mentioned set whether they are equal to the element that has to be removed. The element for which equality is tested (in a step of the testing process) is the last element of the sequence describing the set. If equality is found, the element is removed. If a check shows that the numbers are different, then the element which has been checked, is put at the beginnings of the sequence, and the new 'last element' is checked. By means of the * sign the numbers are separated which are already checked from the numbers which are not yet checked. This rotation technique is due to Van WIJNGAARDEN (1974). The hyperrules introducing and removing the * sign are as follows.
\[ \{ \text{seq} \} - n \rightarrow \{ \star, \text{seq} \} - n \]
\[ \emptyset - n \rightarrow \emptyset \]
\[ \{ \text{seq}, \star \} - n \rightarrow \{ \text{seq} \} \]
\[ \{ \star \} - n \rightarrow \emptyset. \]

The hyperrule removing an element is
\[ \{ \text{lsseq} \star \text{rseq}, n \} - n \rightarrow \{ \text{lsseq} \star \text{rseq} \} - n. \]

The rule rotating the sequence is
\[ \{ \text{lsseq} \star \text{rseq}, n \} - n \rightarrow \{ n \} - n \rightarrow \{ n \} - n \text{ unless } n \text{ is } n. \]

We use the \text{unless} phrase to guarantee that \( n \) and \( n \) are different. If the numbers are different, then the phrase reduces to the empty string. If they are equal the \text{unless} phrase reduces to the expression \text{unless}, and we cannot get rid of this phrase. This means that we are in a blind alley: we do not get an actual rule.

The rules for \( \cup \) use the \( - \) sign. It would be easy to reduce \( \{1,2\} \cup \{2\} \) to \( \{1,2,2\} \) but, in order to avoid this repetition of elements, I first remove the 2 from the leftmost set and then add 2 to the set thus obtained. The rules are as follows.

\[ \text{set} \cup \{ n, \text{rseq} \} \rightarrow \text{set} - n \rightarrow n \cup \{ \text{rseq} \} \]
\[ \{ \text{seq} \} + n \rightarrow \{ \text{seq}, n \} \]
\[ \emptyset + n \rightarrow n \]
\[ \text{set} \cup \{ \emptyset \} \rightarrow \text{set} \]
\[ \text{set} \cup \{ \} \rightarrow \text{set}. \]

This completes the description of the set cf rules needed for dealing with set-theoretical notions by grammatical means.