Compositionality and Model-Theoretic Interpretation

Herman Hendriks
Utrecht Institute of Linguistics OTS, Utrecht University, The Netherlands
ILLC/Department of Philosophy, University of Amsterdam, The Netherlands

Abstract

The present paper[1] studies the general implications of the principle of *compositionality* for the organization of grammar. It will be argued that Janssen's (1986) requirement that syntax and semantics be *similar* algebras is too strong, and that the more liberal requirement that syntax be *interpretable* into semantics leads to a formalization that can be motivated and applied more easily, while it avoids the complications that encumber Janssen's formalization. Moreover, it will be shown that this alternative formalization even allows one to further complete the formal theory of compositionality, in that it is capable of clarifying the role played by *translation*, *model-theoretic interpretation* and *meaning postulates*, of which the latter two aspects received little or no attention in Montague (1970) and Janssen (1986).

1 Compositionality

In its most general form, the principle of compositionality states the following: 'The meaning of an expression is a function of the meanings of its parts and of the way they are syntactically combined.' (Partee 1984, p. 281.) In other words: the meaning of an expression is determined completely by the meanings of its parts plus the information which syntactic rules have been used to build that expression out of those parts. The principle of compositionality is also known as 'Frege's principle'.[2] We will give a formalization of the principle along the lines of Janssen (1986), which, in turn, is based on Montague's seminal paper 'Universal Grammar' (UG, 1970).[3]

[1]This paper is a thoroughly rewritten version of Chapter 2 of Hendriks (1993), to which the reader is occasionally referred for technical details and examples.

[2]Janssen (1986) argues that this attribution is at best a tribute. See also the contributions of Janssen and Pelletier to the present issue of the *Journal of Logic, Language and Information*.

[3]The framework defined in Montague's UG and Janssen's formalization are, roughly speaking, 'different views of the same mathematical object' (Janssen 1986, Part 1, p. 91). The main difference is that Janssen employs many-sorted algebras, whereas Montague uses one-sorted algebras (though with much additional structure). As a consequence of this, Janssen's ap-

As for the syntax, the principle presupposes some set A of expressions and some set F of syntactic rules. This set A includes a set H that consists of the non-compound, lexical expressions. In keeping with the customary assumption within theories of formal grammar that linguistic expressions belong to different syntactic *categories*, we will suppose that the set of expressions is an indexed family of sets: $A = (A_s)_{s \in S}$, where S is the set of *sorts*, which model the syntactic categories, and for each $s \in S$, the set A_s is the set of expressions of category s, or the *carrier* of sort s. This also holds for the set of lexical expressions: $H = (H_s)_{s \in S}$, where $H_s \subseteq A_s$ for all $s \in S$. Since we are dealing with expressions, we will assume that the members of the carriers are strings over some alphabet. But there are no further restrictions on the carriers; they may overlap, be empty, include one another, etcetera. Syntactic rules, or *operators*, $F_\gamma \in F$ yield a unique compound expression a_{n+1} when they apply to a number of expressions $a_1, \ldots, a_n$, the (immediate) parts of a_{n+1}: $F_\gamma(a_1, \ldots, a_n) = a_{n+1}$. We will assume that every F_γ has a fixed number n of expressions to which it applies (where $n \in \mathbb{N}^+$, i.e., $n \in \mathbb{N}$ and $n > 0$). A syntactic rule does not have to yield a expression for every sequence $\langle a_1, \ldots, a_n \rangle$ of expressions in $\cup A \times \ldots \times \cup A$. It may be that F_γ only produces an outcome a_{n+1} for sequences $\langle a_1, \ldots, a_n \rangle$ of which the components belong to certain sorts $s_1, \ldots, s_n$, that is, for $a_1 \in A_{s_1}, \ldots, a_n \in A_{s_n}$. But if this is the case, we assume that F_i does this for *all* $\langle a_1, \ldots, a_n \rangle \in A_{s_1} \times \ldots \times A_{s_n}$, and the outcomes a_{n+1} must without exception belong to the carrier of one sort, $A_{s_{n+1}}$, say. Thus every syntactic operator is a total function $F_\gamma : A_{s_1} \times \ldots \times A_{s_n} \rightarrow A_{s_{n+1}}$, for some $n \in \mathbb{N}^+$, $A_{s_1} \in A$, $\ldots$, $A_{s_n} \in A$, and $A_{s_{n+1}} \in A$. There are no further restrictions on the members of F. The operators may do anything: concatenate, insert, permute, delete, introduce syncategorematic material that does not occur in the arguments, or what have you. The picture of syntax that emerges from these considerations is that of a *many-sorted algebra of signature* π. (The phrase

proach has the following advantages (of which (a) through (c) are also noted in Janssen 1986, Part 1, pp. 90-92):

(a) In UG, the operators in the algebraic sense are untyped, but the syntactic rules are typed. UG requires for each operator a single corresponding semantic operation. However, sometimes one might want to be able to interpret the same operator in different ways, for instance depending on the type of the expressions involved.

(b) Both frameworks require that the operators be total. In the one-sorted context of UG this means that an algebraic operator has to be defined for all elements of the algebra, also for those elements to which the corresponding syntactic rule will never be applied. And worse, even a semantic interpretation has to be specified for the resulting non-expressions.

(c) Janssen's formalization establishes a natural and straightforward relation between the disambiguated language (the members of the term algebra) and the generated language: one obtains an expression of the generated language from an expression in the term algebra by simply evaluating the latter. UG only requires some (further unspecified) relation R between the disambiguated language and the generated language.

(d) While in Janssen's approach syntactic rules operate on expressions of the generated language, they operate on expressions of the disambiguated language in UG. This restricts the set of possible syntactic operations in an unnatural way. For example, an operation $F : A_s \times A_s \rightarrow A_s$ of simple concatenation is not an admissible structural operation in a disambiguated language, since it makes an expression $\alpha\beta\gamma$ ambigous between $F(\alpha, F(\beta, \gamma))$ and $F(F(\alpha, \beta), \gamma)$ (cf. Halvorsen and Ladusaw 1979, footnote 17, p. 221).

'many-sorted algebra of signature π' will often be abbreviated as 'π-algebra'.[4])

(1) $\langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ is a *many-sorted algebra* of *signature* π iff
  (a) S is a non-empty set (of *sorts*);
  (b) $(A_s)_{s\in S}$ is an indexed family of sets (A_s is the *carrier* of s);
  (c) Γ is a set (of *operator indices*);
  (d) π (the *type-assigning function*) assigns to each $\gamma \in \Gamma$ a pair $\langle\langle s_1, \ldots, s_n\rangle, s_{n+1}\rangle$, where $n \in \mathbb{N}^+$, $s_1 \in S, \ \ldots, \ s_{n+1} \in S$; and
  (e) $(F_\gamma)_{\gamma\in\Gamma}$ is an indexed family (of *operators*) such that if $\pi(\gamma) = \langle\langle s_1, \ldots, s_n\rangle, s_{n+1}\rangle$, then $F_\gamma : A_{s_1} \times \ldots \times A_{s_n} \to A_{s_{n+1}}$.

More specifically, the syntactic component is a π-algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ with generating family $H = (H_s)_{s\in S}$, where S is the set of syntactic categories and for each $s \in S$, the set A_s is the set of expressions of category s; Γ is the set of indices of syntactic rules and for each $\gamma \in \Gamma$, syntactic rule F_γ of type $\pi(\gamma) = \langle\langle s_1, \ldots, s_n\rangle, s_{n+1}\rangle$ is a total function $A_{s_1} \times \ldots \times A_{s_n} \to A_{s_{n+1}}$ that yields a unique compound expression a_{n+1} of category s_{n+1} for every sequence $a_1, \ldots, a_n$ of expressions of respective categories $s_1, \ldots, s_n$; and for each $s \in S$, the set H_s is the set of non-compound (lexical) expressions of category s. (A concise survey of the concepts and facts from the theory of many-sorted algebra that will be employed below is given in the Appendix of the present paper.)

At this point it should be observed that natural languages are generally *syntactically ambiguous*. This means that natural language expressions may belong to more than one sort—**walk**, for example, is both a verb and a noun and denotes different sets of individuals depending on its sort (walkers and walks, respectively)—, but also that they may be associated with different syntactic analyses: the expression **old men and women**, for example, may be analyzed as [[**old men**] **and women**] and as [**old** [**men and women**]], two analyses that are responsible for non-equivalent interpretations; likewise, an expression such as **kick the bucket** may be analyzed as an idiomatic lexical expression with a concomitant figurative meaning, but must also be analyzed as a compound expression that has a literal meaning (where literal is figurative for verbal.) As a consequence of the phenomenon of syntactic ambiguity, one cannot in general speak of the meaning of an expression, but only of the meaning of an expression with respect to a certain sort and a certain syntactic analysis, that is: with respect to a certain so-called *derivational history*. In order to be able to refer to derivational histories of expressions, we invoke the concept of a *term algebra*. Term algebras play an important role in the formalization of the compositionality principle. The carriers of term algebras consist of symbols, 'syntactic terms', which can be seen as representations of the derivational histories of the generated algebra with which they are associated. Accordingly, term algebras are invariably syntactically unambiguous, or *free* algebras (formal definitions of these notions are given in the Appendix). Now, since the meaning of an expression depends on its sort and syntactic analysis, the meanings of the members

[4]The term 'many-sorted algebra' stems from Adj (1977). Our terminology deviates from Janssen (1986, Part 1, p. 43), where a pair $\langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ meeting the requirements in (1) is called a 'many-sorted algebra of signature (S, Γ, π)'.

of the carriers of the syntactic algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ with generating family $(H_s)_{s\in S}$ are defined on the members of the carriers of the corresponding term algebra $T_{A,H} = \langle (T_{A,H,s})_{s\in S}, (F^T_\gamma)_{\gamma\in\Gamma}\rangle$, an algebra in which these aspects are represented.

What about these meanings? By analogy with the idea that linguistic expressions belong to different categories, it has become customary to assume that their meanings, or interpretations, inhabit various semantic *types*, and that, moreover, the interpretations of different expressions of the same category belong to the same semantic type. Firstly, we will therefore assume that also the members of the semantic domain constitute an indexed family of sets $(B_t)_{t\in T}$, where T is the set of semantic sorts, which model the types, and for each $t \in T$, the set B_t is the set of semantic objects of sort t, or the carrier of sort t; and that, furthermore, there is a function σ which associates every syntactic sort s with a semantic sort $\sigma(s)$, so that semantic objects of sort $\sigma(s)$ can serve as interpretations of syntactic expressions of sort s. Secondly, given this much, the principle of compositionality—according to which the meaning of a compound expression is a *function* of the meanings of its constituent parts and the way they are syntactically combined[5]—literally requires that for every way of syntactically combining expressions, i.e., for every syntactic operator F_γ, there is a semantic *function* G_δ such that for every sequence $a_1, \ldots, a_n$ of expressions, the meaning of the compound expression built up from these expressions, i.e., the meaning of the expression $F_\gamma(a_1, \ldots, a_n)$ which results from applying F_γ to $a_1, \ldots, a_n$, is equal to the value which the function G_δ assigns to the meanings of $a_1, \ldots, a_n$. More formally: let h denote 'the meaning of'. Then:

(2) $\quad h(F_\gamma(a_1, \ldots, a_n)) = G_\delta(h(a_1), \ldots, h(a_n))$

This means that the semantics, too, is a many-sorted algebra. In addition to a family of sets indexed by sorts, it includes a number of operators: functions from the Cartesian product of a number of semantic carriers to some semantic carrier.[6] Hence, thirdly, besides the function σ mapping categories to types, we need a function ρ which associates every n-ary operator F_γ in the syntactic algebra with an n-ary operator $G_{\rho(\gamma)}$ in the semantic algebra. And since every syntactic sort s is associated with a semantic sort $\sigma(s)$, it must hold that $\omega(\rho(\gamma)) = \langle\langle\sigma(s_1), \ldots, \sigma(s_n)\rangle, \sigma(s_{n+1})\rangle$ whenever $\pi(\gamma) = \langle\langle s_1, \ldots, s_n\rangle, s_{n+1}\rangle$, where π and ω are the respective type-assigning functions of the syntactic and semantic algebra.

These considerations can be formalized by means of the following notions. Let $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ be a π-algebra, let $B = \langle (B_t)_{t\in T}, (G_\delta)_{\delta\in\Delta}\rangle$ be an ω-algebra, and let $\sigma : S \to T$ and $\rho : \Gamma \to \Delta$ be functions. Then:

(3) $\quad A$ is *(σ, ρ)-interpretable in B* iff for all $\gamma \in \Gamma$: if $\pi(\gamma) = \langle\langle s_1, \ldots, s_n\rangle, s_{n+1}\rangle$, then $\omega(\rho(\gamma)) = \langle\langle\sigma(s_1), \ldots, \sigma(s_n)\rangle, \sigma(s_{n+1})\rangle$.

[5]That is, strictly speaking, the meaning of an expression relativized to a particular sort and derivational history of that expression.

[6]We will assume that these are total functions, just as the functions in the syntactic algebra.

Moreover, let π-algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ be (σ,ρ)-interpretable in ω-algebra $B = \langle (B_t)_{t\in T}, (G_\delta)_{\delta\in\Delta}\rangle$ and let $h[A_s]$ denote the set $\{h(a) \mid a \in A_s\}$. Then:

(4) $h : \bigcup_{s\in S} A_s \to \bigcup_{t\in T} B_t$ is a (σ,ρ)-*homomorphism from* A *to* B iff
  (i) for all $s \in S$: $h[A_s] \subseteq B_{\sigma(s)}$ (h respects the sorts); and
  (ii) if $\pi(\gamma) = \langle\langle s_1,\ldots,s_n\rangle, s_{n+1}\rangle$ and $a_1 \in A_{s_1},\ldots,a_n \in A_{s_n}$,
  then $h(F_\gamma(a_1,\ldots,a_n)) = G_{\rho(\gamma)}(h(a_1),\ldots,h(a_n))$
  (h respects the operators).

Summing up, then, the principle of compositionality dictates the following: (a) the syntax is a π-algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ with generating family $H = (H_s)_{s\in S}$; (b) the semantic domain is an ω-algebra $B = \langle (B_t)_{t\in T}, (G_\delta)_{\delta\in\Delta}\rangle$ such that A is (σ,ρ)-interpretable in B for some functions $\sigma : S \to T$ and $\rho : \Gamma \to \Delta$; and (c) meaning assignment is a (σ,ρ)-homomorphism from $T_{A,H}$, the term algebra of A with respect to H, to B.

2 Similarity versus Interpretability

The above formalization of the compositionality principle is more or less the same as the one given by Janssen (1986), except for a seemingly minor point which will turn out to have rather far-reaching consequences.

Janssen's definition of (σ,ρ)-homomorphism is identical to the one given in (4), but (σ,ρ)-homomorphisms are allowed to exist only between algebras A and B which are '(σ,ρ)-similar'. Here is the definition of that notion (Janssen 1986, pp. 67–8). Let $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ be a π-algebra, let $B = \langle (B_t)_{t\in T}, (G_\delta)_{\delta\in\Delta}\rangle$ be an ω-algebra, and let $\sigma : S \to T$ and $\rho : \Gamma \to \Delta$ be *bijections*. Then:

(5) A and B are (σ,ρ)-*similar* iff for all $\gamma \in \Gamma$: $\omega(\rho(\gamma)) =$
  $\langle\langle \sigma(s_1),\ldots,\sigma(s_n)\rangle, \sigma(s_{n+1})\rangle$ *iff* $\pi(\gamma) = \langle\langle s_1,\ldots,s_n\rangle, s_{n+1}\rangle$.

So, (σ,ρ)-similarity differs from (σ,ρ)-interpretability in two respects: the former notion requires (a) that the functions σ and ρ be *bijections*; and (b) that for all $\gamma \in \Gamma$: $\omega(\rho(\gamma)) = \langle\langle \sigma(s_1),\ldots,\sigma(s_n)\rangle, \sigma(s_{n+1})\rangle$ if *and only if* $\pi(\gamma) = \langle\langle s_1,\ldots,s_n\rangle, s_{n+1}\rangle$. It may be noted, however, that the 'only if'-part of (b) is superfluous given (a) and the 'if'-part of (b),[7] so that the only real difference between the two notions is (a).

We will now argue that the requirement of (σ,ρ)-similarity leads to complications, and show that these undesirable consequences can be avoided by replacing the requirement of (σ,ρ)-similarity between the syntactic algebra and the semantic algebra by the requirement that the syntactic algebra be (σ,ρ)-*interpretable* in the semantic algebra. First, note that (σ,ρ)-similarity is too strong in view of the explication of the intuitive idea of compositionality that we have given in the previous section, but that, on the other hand, (σ,ρ)-interpretability is a notion that *can* be motivated in this way: it suffices to require that there be

[7] Cf. Hendriks 1993, footnote 15, p. 146.

functions (and not necessarily bijections) $\sigma : S \to T$ and $\rho : \Gamma \to \Delta$ that connect the sorts and operator indices of the syntactic algebra to the sorts and the operator indices of the semantic algebra. Second, since bijections are functions, we have that π-algebra A is (σ, ρ)-interpretable in ω-algebra B whenever A and B are (σ, ρ)-similar. The converse does not hold. Therefore, (σ, ρ)-similarity is stronger than (σ, ρ)-interpretability, so that the latter notion is more easily applicable in principle. And third, the requirement that the domains of syntax and semantics constitute similar algebras is responsible for technical complications in that it does in fact lead to actual problems of applicability, because in practice it is generally not the case that there are bijections $\sigma : S \to T$ from the syntactic categories to the semantic types and $\rho : \Gamma \to \Delta$ from the syntactic operator indices to the semantic operator indices that are respected by the meaning assignment homomorphism. Usually, the syntactic and the semantic algebra fail to be (σ, ρ)-similar, since (a) some semantic types do not correspond to syntactic categories, so that σ is not surjective; (b) different syntactic categories correspond to one and the same semantic type, so that σ is not injective; (c) some semantic operators do not figure as the counterpart of a syntactic operator, so that ρ is not surjective; and (d) different syntactic operators correspond to one and the same semantic operator, so that ρ is not injective. Moreover, as will be shown below, if a formal logical language is used as an auxiliary translation language, syntactic operators may correspond to semantic operators that—though definable in terms of the operators of the semantic algebra—are themselves not actually present in the semantic algebra, so that ρ is not even a function.

Consider, by way of illustration, the grammar fragment in Montague's 'The Proper Treatment of Quantification in Ordinary English' (PTQ, 1973), a paper which has acquired a paradigmatic status within the framework of compositional model-theoretic semantics. The set of syntactic sorts of this 'PTQ fragment' is defined as the smallest set S such that e and t are in S; and whenever A and B are in S, then A/B and $A/\!\!/B$ are also in S (Montague 1973, p. 249). The set of semantic sorts is defined as the smallest set T such that e and t are in T; whenever a and b are in T, then (a, b) is in T; and whenever a is in T, then (s, a) is in T (Montague 1973, p. 256). And the function σ that associates the syntactic sorts in S with the semantic sorts in T is defined by $\sigma(\mathrm{e}) = \mathrm{e}$, $\sigma(\mathrm{t}) = \mathrm{t}$, and $\sigma(A/B) = \sigma(A/\!\!/B) = ((\mathrm{s}, \sigma(B)), \sigma(A))$ (Montague 1973, p. 260). Observe, first, that σ is not a surjection, since there are semantic sorts such as (s, e) and (e, t) that do not correspond to a syntactic sort—in fact, only e, t and sorts of the form $((\mathrm{s}, a), b)$ are the σ-value of some syntactic sort—; and, second, that σ is not an injection, since there are different syntactic sorts that correspond to one and the same semantic sort: the syntactic sorts t/e (of intransitive verb phrases) and $\mathrm{t}/\!\!/\mathrm{e}$ (of common noun phrases), for example, correspond both to the semantic sort $((\mathrm{s}, \mathrm{e}), \mathrm{t})$.

Speaking in strict many-sorted algebraic terms, moreover, the PTQ fragment contains syntactic operators such as the following ones (see Montague 1973, pp. 251–3, for details; in (6), IV abbreviates the syntactic sort t/e):

(6) $F_{\text{S8-F6}} : A_{\text{IV}/\!\!/\text{IV}} \times A_{\text{IV}} \to A_{\text{IV}}$; $F_{\text{S11-F8}} : A_{\text{t}} \times A_{\text{t}} \to A_{\text{t}}$; and
$F_{\text{S10-F7}} : A_{\text{IV/IV}} \times A_{\text{IV}} \to A_{\text{IV}}$; $F_{\text{S11-F9}} : A_{\text{t}} \times A_{\text{t}} \to A_{\text{t}}$.

The semantic operators of PTQ correspond to the interpretations of the clauses for the construction of meaningful expressions of the logical language IL ('Intensional Logic'; see Montague 1973, pp. 256–60, for the semantic interpretation of IL). Here are some examples of these clauses (in (7), T denotes the set of semantic sorts; and for $a \in T$, the set of meaningful IL expressions of sort a is denoted by B_a):

(7) $K_{()\text{-}a\text{-}b} : B_{(a.b)} \times B_a \to B_b$ (*application*) for $a, b \in T$, where
$K_{()\text{-}a\text{-}b}(\alpha, \beta) = [\alpha(\beta)]$
$K_{\text{in-}a} : B_a \to B_{(s,a)}$ (*intension*) for $a \in T$, where $K_{\text{in-}a}(\alpha) = {}^\wedge\alpha$;
$K_\wedge : B_{\text{t}} \times B_{\text{t}} \to B_{\text{t}}$ (*conjunction*), where $K_\wedge(\alpha, \beta) = [\alpha \wedge \beta]$; and
$K_\vee : B_{\text{t}} \times B_{\text{t}} \to B_{\text{t}}$ (*disjunction*), where $K_\vee(\alpha, \beta) = [\alpha \vee \beta]$.

Focusing on the correspondence ρ between the syntactic and the semantic operators (see Montague 1973, pp. 261–2), we may note that ρ is not an injection, since it turns out that, for example, the syntactic operators $F_{\text{S8-F6}} : A_{\text{IV}/\!\!/\text{IV}} \times A_{\text{IV}} \to A_{\text{IV}}$ and $F_{\text{S10-F7}} : A_{\text{IV/IV}} \times A_{\text{IV}} \to A_{\text{IV}}$ both correspond to the semantic operator that applies the *application* operator to its first argument and the result of applying the *intension* operator to its second argument. In addition to this, it can be observed that ρ is not a surjection either, because except for $K_\wedge : B_{\text{t}} \times B_{\text{t}} \to B_{\text{t}}$ and $K_\vee : B_{\text{t}} \times B_{\text{t}} \to B_{\text{t}}$, which figure as the semantic counterparts of the respective syntactic operators $F_{\text{S11-F8}} : A_{\text{t}} \times A_{\text{t}} \to A_{\text{t}}$ and $F_{\text{S11-F9}} : A_{\text{t}} \times A_{\text{t}} \to A_{\text{t}}$, none of the semantic operators is associated with one of the syntactic operators in the PTQ fragment. As a matter of fact, the correspondence ρ between the syntactic and the semantic operators even fails to be a function, since apart from $F_{\text{S11-F8}} : A_{\text{t}} \times A_{\text{t}} \to A_{\text{t}}$ and $F_{\text{S11-F9}} : A_{\text{t}} \times A_{\text{t}} \to A_{\text{t}}$, all syntactic operators correspond to semantic operators that are indeed definable as non-trivial compositions of semantic operators, but do not belong to the semantic algebra proper. Thus we just noted that both $F_{\text{S8-F6}} : A_{\text{IV}/\!\!/\text{IV}} \times A_{\text{IV}} \to A_{\text{IV}}$ and $F_{\text{S10-F7}} : A_{\text{IV/IV}} \times A_{\text{IV}} \to A_{\text{IV}}$ correspond to a semantic operator $O : B_{((\text{s},((\text{s},\text{e}),\text{t})),((\text{s},\text{e}),\text{t}))} \times B_{((\text{s},\text{e}),\text{t})} \to B_{((\text{s},\text{e}),\text{t})}$ that can be considered the composition of the application operator $K_{()\text{-}(\text{s},((\text{s},\text{e}),\text{t}))\text{-}((\text{s},\text{e}),\text{t})}$ and the intension operator $K_{\text{in-}((\text{s},\text{e}),\text{t})}$, in that $O(\alpha, \beta) = K_{()\text{-}(\text{s},((\text{s},\text{e}),\text{t}))\text{-}((\text{s},\text{e}),\text{t})}(\alpha, K_{\text{in-}((\text{s},\text{e}),\text{t})}(\beta)) = [\alpha({}^\wedge\beta)]$. *Mutatis mutandis*, the same holds for the semantic operators that correspond to all other syntactic operators. It will be shown below that the 'addition' of this kind of operators is always unproblematic.

In order to bridge such gaps of dissimilarity between syntactic and semantic algebras, Janssen invokes the notion of a 'safe deriver'. This notion is introduced in the course of giving a definition of a Montague grammar, which, in its most simple form, consists of a many-sorted algebra and a homomorphic interpretation. 'However, one always uses, in practice, some formal (logical) language as auxiliary language, and the language of which one wishes to define the meanings is translated into this formal language. Thus the meaning assignment is performed indirectly. The aspect of translating into an auxiliary language is, in my opinion, unavoidable for practical reasons, and I therefore wish to incorporate this aspect in the definition of a Montague grammar.' (Janssen 1986, Part 1, p. 81.) This definition is given in (8), and the situation it describes can be sketched as in (9) (cf. Janssen 1986, Part 1, pp. 75 and 82):

(8) A *Montague grammar* consists of:
a *syntactic* π-algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ generated by $H = (H_s)_{s\in S}$;
a *logical* ω-algebra $B = \langle (B_t)_{t\in T}, (K_\delta)_{\delta\in\Delta}\rangle$;
a *semantic* ω-algebra $M = \langle (M_t)_{t\in T}, (G_\delta)_{\delta\in\Delta}\rangle$ similar to B;
an *interpretation* homomorphism $\mathcal{I}$ from B to M;
an algebra $D(B)$ similar to A, where D is a *safe deriver*; and
a *translation* homomorphism *tr* from $T_{A,H} = \langle (T_{A,H,s})_{s\in S}, (F^T_\gamma)_{\gamma\in\Gamma}\rangle$, the term algebra of A with respect to H, to $D(B)$.

(9)
$$\begin{array}{ccc} & & T_{A,H} \\ & & \downarrow tr \\ B & \Longrightarrow & D(B) \\ \downarrow \mathcal{I} & & \downarrow \mathcal{I} \\ M & \Longrightarrow & M' \end{array}$$

In general, a *deriver* D is a function from algebras to algebras: 'a method to obtain new algebras from old ones', and:[8]

(10) A deriver D is *safe* for algebra A iff for all algebras B and all surjective homomorphisms $\mathcal{I}$ from A to B there is a unique algebra B' such that for the restriction $\mathcal{I}'$ of $\mathcal{I}$ to $D(A)$ it holds that $\mathcal{I}'$ is a surjective homomorphism from $D(A)$ to B'. (Janssen 1986, Part 1, p. 76)

Janssen's deriver D is the composition of four basic derivers, viz., AddOp, AddSorts, DelOp and DelSorts,[9] which, by adding operators, adding sorts, deleting operators and deleting sorts, respectively, transform the logical algebra B into an algebra $D(B) = \text{DelSorts}(\text{DelOp}(\text{AddSorts}(\text{AddOp}(B))))$ which is similar to

[8] Janssen offers no arguments why (10) should define the safeness of a deriver. The only motivation given is the following: 'The requirement that $\mathcal{I}'$ is a surjective homomorphism is important. If we would not require this, then B' would in most cases not be unique. An extreme example arises when $D(A)$ is an empty algebra. Then there are infinitely many algebras B' such that $\mathcal{I}'$ is a homomorphism from $D(A)$ to B', but only one such that $\mathcal{I}'$ is a surjective homomorphism from $D(A)$ to B'.' (Janssen 1986, Part 1, p. 76). In the context of the present paper it is perhaps interesting to observe that an operator Φ_* over an algebra A is universally $\mathcal{I}$-functional (cf. Section 3 below) if and only if the deriver AddOp$[\{\Phi_*\}]$ is safe for A in the sense of definition (10) above, but that, as will be shown below, and contrary to what Janssen's motivation for the notion of safeness suggests, it is not so much the *uniqueness* as the *existence* of the algebra B' which is at stake. (Notice, by the way, that definition (10) does not say how $D(A)$ relates to A, so that it is not clear what $\mathcal{I}'$ denotes, given $\mathcal{I}$.)

[9] In fact, the deriver DelSorts replaces the more complicated and problematic deriver SubAlg actually proposed by Janssen (see Hendriks 1993, pp. 162–3, for motivation and details).

the syntactic algebra A. As to the question whether is it really necessary to incorporate this laborious process of deriving an algebra $D(B)$ similar to the syntactic algebra A in four steps from the original logical algebra B into the general definition of a Montague grammar, it can be noted that Janssen emphasizes repeatedly that the possibility of a homomorphism presupposes similarity: 'A mapping is called a homomorphism if it respects the structures of the algebras involved. This is only possible if the two algebras have a similar structure.' (Janssen 1986, Part 1, pp. 21–22; see also pp. 67–70.) Nevertheless, it can also be observed that if, instead of similarity, interpretability is assumed, we are done in one step: we only need to consider the 'addition' of operators to the logical algebra. Regarding this aspect of the derivation of a new algebra from the algebra of the logical language, Janssen concludes: 'In one respect this attempt [to formalize the compositionality principle] probably has not been successful: the description of how to obtain new algebras out of old ones. There is no general theory which I could use here, and I had to apply *ad hoc* methods.' (Janssen 1986, Part 1, p. 42; see also p. 83.)

Contrary to this, however, we feel that the appropriate conclusion to be drawn is that the very notion of a 'safe deriver' is *ad hoc*, since it is an artefact created by the requirement of similarity—a requirement which, as we pointed out above, is itself undermotivated in view of the conditions imposed by the compositionality principle. Accordingly, we will now show that the addition of operators to the logical algebra is not, as Janssen puts it, the 'most important' deriver, but the only 'deriver' that has to be taken into account at all. In order to demonstrate this, we will first discuss the use of a formal logical language as an auxiliary translation language and then clarify the role played by model-theoretic interpretation and meaning postulates.

3 Translation, Models and Meaning Postulates

The basic idea of using a formal logical language as an auxiliary translation language is simply that a syntactic term in the term algebra of the generated syntactic algebra is indirectly assigned the interpretation $\mathcal{I}(\beta)$ of the expression β of the logical language that serves as the translation of the term. Thus, each syntactic term τ is associated with a unique translation $tr(\tau)$, and this translation induces the interpretation $\mathcal{I}(tr(\tau))$: 'the principal use of translations is the semantical one of inducing interpretations' (Montague 1970, p. 232). For such an indirect interpretation assignment to be compositional, the composition[10] $tr \circ \mathcal{I}$ of the translation and interpretation step has to be a homomorphism, i.e., a function, which entails that the logical language must be unambiguous. In general, formal logical languages and their semantic interpretations are defined by specifying (i) a generated algebra of well-formed logical expressions; and (ii) an interpretation homomorphism from this generated algebra to a semantic algebra. This homomorphism is not specified by stating its values for all

[10]The composition $f \circ g$ of two functions f and g is defined by $f \circ g(x) = g(f(x))$.

arguments (since there are generally infinitely many well-formed logical expressions), but by (a) providing a mapping $\mathcal{I}$ which assigns a member of sort $\sigma(t)$ in the semantic algebra to each generator of sort t in the logical algebra; and (b) associating each logical operator K_δ of type $\langle\langle t_1,\ldots,t_n\rangle, t_{n+1}\rangle$ with a semantic operator $G_{\rho(\delta)}$ of type $\langle\langle\sigma(t_1),\ldots,\sigma(t_n)\rangle,\sigma(t_{n+1})\rangle$, whereby $\mathcal{I}(K_\delta(\beta_1,\ldots,\beta_n))$ is defined as $G_{\rho(\delta)}(\mathcal{I}(\beta_1),\ldots,\mathcal{I}(\beta_n))$.[11] Note that this procedure is only guaranteed to result—and does indeed result[12]—in a *homomorphism* $\mathcal{I}$ if the logical language is syntactically unambiguous. Therefore, the generated algebra of a logical language is as a rule a *free* algebra.[13]

Furthermore, formal logical languages usually have a *model-theoretic* interpretation, which means that their interpretation homomorphism $\mathcal{I}$ is defined pointwise: on the basis of a class[14] $\mathcal{M}$ of models for the logical language, the interpretation of logical expressions β is specified by separately defining $in_m(\beta)$ for each $m \in \mathcal{M}$, where $in_m(\beta)$ is given by (a) a specification of $in_m(\beta)$ for logical generators β; and (b) an assignment of a semantic operator $G_{m,\delta}$ to each logical operator K_δ, so that $in_m(K_\delta(\beta_1,\ldots,\beta_n))$ is defined as $G_{m,\delta}(in_m(\beta_1),\ldots,in_m(\beta_n))$. Of course, the point of this model-theoretic set-up is that a logical expression can have different interpretations in different models: there is not in general a single object that serves as the interpretation of a logical expression β in all models m. Hence, in order to be able to talk about 'the' interpretation $\mathcal{I}(\beta)$ of a logical expression β, one has to incorporate the models into the concept of interpretation: $\mathcal{I}(\beta)$ is that function from models to interpretations in models such that $\mathcal{I}(\beta)(m) = in_m(\beta)$ for all $m \in \mathcal{M}$.[15]

Observe that if $B = \langle(B_t)_{t\in T},(K_\delta)_{\delta\in\Delta}\rangle$ is a model-theoretically interpreted logical algebra of signature ω, then such an interpretation function $\mathcal{I}$ can be construed as a—surjective—homomorphism from B to the following semantic

[11] The correspondence between the sorts and operator indices of the logical and the semantic algebra is usually established by bijections (identity functions) σ and ρ. We will henceforth simply assume that σ and ρ are identity functions, and call $\mathcal{I}$ an (=,=)-homomorphism.

[12] Observe that $\mathcal{I}$ is designed so as to respect sorts and operators. Moreover, if the logical language is a syntactically unambiguous free algebra, then all generators belong to exactly one sort and are different from all non-generators, so that every generator is assigned exactly one value by $\mathcal{I}$. Besides, the operators are injections with disjoint ranges, which means that also the non-generators receive a unique value, and, consequently, that $\mathcal{I}$ is a homomorphism.

[13] Witness L.T.F. Gamut's adage: 'Logical languages wear their meanings on their sleeves' (p.c.).

[14] 'Class', since the collection $\mathcal{M}$ of models for a logical language is generally too large to be countenanced as a set in the sense of axiomatic set theory. The same holds for the notions $\mathcal{I}(\beta)$, $\mathcal{I}_t$, $\mathcal{G}_\delta$, $\mathcal{S}$, $\Phi^{\mathcal{I}}_\gamma$, $K^{\mathcal{I}}_\delta$, $\mathcal{S}$ and their *MP*-superscripted couterparts that will be defined below. The use of calligraphic letters is meant to visualize this set-theoretical proviso.

[15] This is another application of the *cylindrification* technique used for assigning interpretations in models to logical languages which involve well-formed expressions that may contain free variables, There, cylindrification amounts to the incorporation of the variable assignments into the concept of interpretation (cf. Montague 1970, p. 228; Janssen 1986, Part 1, pp. 28–35).

ω-algebra $\mathcal{S}$:

(11) $\mathcal{S} = \langle(\mathcal{I}_t)_{t\in T}, (\mathcal{G}_\delta)_{\delta\in\Delta}\rangle$, where
  (a) $\mathcal{I}_t = \{\mathcal{I}(\beta) \mid \beta \in B_t\}$; and
  (b) if $\omega(\delta) = \langle\langle t_1, \ldots, t_n\rangle, t_{n+1}\rangle$, then $\mathcal{G}_\delta : \mathcal{I}_{t_1} \times \ldots \times \mathcal{I}_{t_n} \to \mathcal{I}_{t_{n+1}}$, where $\mathcal{G}_\delta(\mathcal{I}(\beta_1), \ldots, \mathcal{I}(\beta_n)) = \mathcal{I}(K_\delta(\beta_1, \ldots, \beta_n))$.

So, let us assume that each syntactic term τ in a carrier of the term algebra $T_{A,H} = \langle(T_{A,H,s})_{s\in S}, (F^T_\gamma)_{\gamma\in\Gamma}\rangle$ of the syntactic π-algebra $A = \langle(A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ with respect to the generating family $(H_s)_{s\in S}$ is to be assigned a logical translation $tr(\tau)$ in some carrier of the logical ω-algebra $B = \langle(B_t)_{t\in T}, (K_\delta)_{\delta\in\Delta}\rangle$, a translation which is interpreted as $\mathcal{I}(tr(\tau))$ in some carrier of the semantic ω-algebra $\mathcal{S} = \langle(\mathcal{I}_t)_{t\in T}, (\mathcal{G}_\delta)_{\delta\in\Delta}\rangle$ via the $(=,=)$-homomorphism $\mathcal{I}$ (cf. footnote 11). Note that for such an indirect interpretation assignment to be compositional, the composition $tr \circ \mathcal{I}$ of the translation and interpretation function has to be a homomorphism. This entails that $tr \circ \mathcal{I}$ has to respect the sorts, so there must be a function σ from the sorts S of $T_{A,H}$ to the sorts T of $\mathcal{S}$ such that $tr \circ \mathcal{I}[T_{A,H,s}] \subseteq \mathcal{I}_{\sigma(s)}$ for all $s \in S$. However, since the correspondence between the sorts of B and $\mathcal{S}$ is established by an injection (the identity function $=$), this means that the translation tr by itself must also respect the sorts, i.e.: $tr[T_{A,H,s}] \subseteq B_{\sigma(s)}$ for all $s \in S$.

The assignment of translations $tr(\tau)$ to syntactic terms τ in the term algebra $T_{A,H}$ of a generated syntactic algebra proceeds in a way analogous to the assignment of interpretations to expressions in a logical language: because there are generally infinitely many syntactic terms to be translated, the translation function is not specified by stating its values for all terms, but by providing a mapping tr which (a) associates each syntactic term τ that corresponds to a generator h of category s in the syntactic algebra with some expression of type $\sigma(s)$ in the logical algebra B; and (b) associates each term algebra operator F^T_γ of type $\langle\langle s_1, \ldots, s_n\rangle, s_{n+1}\rangle$ with some function $\Phi_\gamma : B_{\sigma(s_1)} \times \ldots \times B_{\sigma(s_n)} \to B_{\sigma(s_{n+1})}$, whereby $tr(F^T_\gamma(\tau_1, \ldots, \tau_n))$ is defined as $\Phi_\gamma(tr(\tau_1), \ldots, tr(\tau_n))$. Since the term algebra of a generated syntactic algebra is a free algebra, this procedure is guaranteed to result in the assignment of a unique translation $tr(\tau)$ to each syntactic term τ. Besides, the π-algebra $T_{A,H} = \langle(T_{A,H,s})_{s\in S}, (F^T_\gamma)_{\gamma\in\Gamma}\rangle$ is (σ, ρ)-interpretable in the π'-algebra $B' = \langle(B_t)_{t\in T}, (\Phi_\gamma)_{\gamma\in\Gamma}\rangle$ for ρ and π' such that $\rho(\gamma) = \gamma$ and $\pi'(\gamma) = \langle\langle\sigma(s_1), \ldots \sigma(s_n)\rangle, \sigma(s)\rangle$ iff $\pi(\gamma) = \langle\langle s_1, \ldots s_n\rangle, s\rangle$, and the translation function tr is a (σ, ρ)-homomorphism from $T_{A,K}$ to B'.

It is worth mentioning here that the logical algebra is usually exploited 'at a higher level' in the process of translation. Thus terms corresponding to generators of the syntactic algebra need not be translated into generators of the logical algebra. In the PTQ fragment, for example, the syntactic generator **run** is assigned a generator—viz., a (non-logical) constant—of the corresponding logical sort as its translation, but the translation of the syntactic generator **be** is a (highly) compound expression. And, more importantly, the functions Φ_γ associated with the operators F^T_γ of the syntactic term algebra do not necessarily coincide with the operators K_δ that are actually present in the logical algebra.

We have already seen that in the PTQ fragment, for example, the syntactic operators $F_{\text{S11-F8}}$ and $F_{\text{S11-F9}}$ turn out to correspond to operators that belong to the logical algebra, viz., $K_\wedge$ and $K_\vee$, respectively, but that this does not hold for the other operators: the latter are all associated with a logical operator that is definable as a non-trivial composition of the operators present in the logical algebra, but does not itself belong to that algebra. Now, let $\Phi_\gamma : B_{\sigma(s_1)} \times \ldots \times B_{\sigma(s_n)} \to B_{\sigma(s_{n+1})}$ be such a logical operator. We define $\Phi_\gamma^{\mathcal{I}}$, *the relation* $\mathcal{I}$*-induced by* Φ_γ, as the following collection:

(12) $\{\langle\langle\mathcal{I}(\beta_1),\ldots,\mathcal{I}(\beta_n)\rangle,\mathcal{I}(\beta_{n+1})\rangle \mid \langle\langle\beta_1,\ldots,\beta_n\rangle,\beta_{n+1}\rangle \in \Phi_\gamma\}$

We will say that an operator Φ_γ is $\mathcal{I}$*-functional* iff the relation $\Phi_\gamma^{\mathcal{I}}$ $\mathcal{I}$-induced by Φ_γ is a function, i.e., iff there are no $\langle\langle\varepsilon_1,\ldots,\varepsilon_n\rangle,\varepsilon\rangle \in \Phi_\gamma^{\mathcal{I}}$ and $\langle\langle\varepsilon_1',\ldots,\varepsilon_n'\rangle,\varepsilon'\rangle \in \Phi_\gamma^{\mathcal{I}}$ such that $\langle\varepsilon_1,\ldots,\varepsilon_n\rangle = \langle\varepsilon_1',\ldots,\varepsilon_n'\rangle$ while $\varepsilon \neq \varepsilon'$. Note that for the operators K_δ of the logical algebra $B = \langle(B_t)_{t\in T},(K_\delta)_{\delta\in\Delta}\rangle$ itself it holds that $K_\delta^{\mathcal{I}} = \mathcal{G}_\delta$, where $\mathcal{G}_\delta$ is the function defined in (11) above. Hence, obviously, K_δ is $\mathcal{I}$-functional for all $\delta \in \Delta$ and $\mathcal{I}$ is an (=,=)-homomorphism from B to the semantic algebra $\mathcal{S} = \langle(\mathcal{I}_t)_{t\in T},(\mathcal{G}_\delta)_{\delta\in\Delta}\rangle = \langle(\mathcal{I}_t)_{t\in T},(K_\delta^{\mathcal{I}})_{\delta\in\Delta}\rangle$. In general, as regards the compositionality of an indirect interpretation assignment in terms of a translation homomorphism *tr* from the syntactic term algebra $T_{A,H} = \langle(T_{A,H,s})_{s\in S},(F_\gamma^T)_{\gamma\in\Gamma}\rangle$ to some 'derived' logical algebra $B' = \langle(B_t)_{t\in T},(\Phi_\gamma)_{\gamma\in\Gamma}\rangle$ and an interpretation homomorphism $\mathcal{I}$ from the logical algebra $B = \langle(B_t)_{t\in T},(K_\delta)_{\delta\in\Delta}\rangle$ to the semantic algebra $\mathcal{S} = \langle(\mathcal{I}_t)_{t\in T},(K_\delta^{\mathcal{I}})_{\delta\in\Delta}\rangle$, it may be noted that the structure $\mathcal{S}' = \langle(\mathcal{I}_t)_{t\in T},(\Phi_\gamma^{\mathcal{I}})_{\gamma\in\Gamma}\rangle$ is an algebra—and $\mathcal{I}$, consequently, a homomorphism from B' to $\mathcal{S}'$—if and only if for all $\gamma \in \Gamma$ it holds that Φ_γ is $\mathcal{I}$-functional.[16] Since the composition of two homomorphisms is again a homomorphism, this means that we have that $tr \circ \mathcal{I}$ is a homomorphism from $T_{A,H}$ to $\mathcal{S}'$ if and only if Φ_γ is $\mathcal{I}$-functional for all $\gamma \in \Gamma$.

Summing up: in order for the composition $tr \circ \mathcal{I}$ of a homomorphism *tr* from $T_{A,H}$ to B' and a homomorphism $\mathcal{I}$ from B to $\mathcal{S}$ to be a homomorphism from $T_{A,H}$ to $\mathcal{S}'$, all functions Φ_γ in B' that are associated with operators F_γ^T in $T_{A,H}$ must $\mathcal{I}$-induce a *function* $\Phi_\gamma^{\mathcal{I}}$. This raises the following question: given a homomorphism $\mathcal{I}$ from the logical algebra $B = \langle(B_t)_{t\in T},(K_\delta)_{\delta\in\Delta}\rangle$ to a semantic algebra $\mathcal{S} = \langle(\mathcal{I}_t)_{t\in T},(K_\delta^{\mathcal{I}})_{\delta\in\Delta}\rangle$, which operators $\Phi_\gamma : B_{t_1} \times \ldots \times B_{t_n} \to B_{t_{n+1}}$ are $\mathcal{I}$-functional? A partial answer to this question is given in the Appendix, where it is shown that the class of operators that are $\mathcal{I}$-functional for *all* homomorphisms $\mathcal{I}$ from B to some algebra $\mathcal{S}$ includes the *polynomial* operators over the algebra B. The class of polynomial operators over B consists of elementary operators—projection functions and constant functions—plus operators that are definable as compositions of these elementary operators and the operators in $(K_\delta)_{\delta\in\Delta}$. On the other hand, it is also obvious that for a *particular* homomorphism $\mathcal{I}$ from the logical algebra B to a *specific* semantic algebra $\mathcal{S}$, there are always[17] non-polynomial $\mathcal{I}$-functional operators. For either there are

[16] Strictly speaking, however, cf. Hendriks 1993, footnote 42, p. 170.
[17] Well, almost always, cf. Hendriks 1993, footnote 44, p. 171.

no $t \in T$, $\beta \in B_t$ and $\beta' \in B_t$ such that $\mathcal{I}(\beta) = \mathcal{I}(\beta')$, or there are such $t \in T$, $\beta \in B_t$ and $\beta' \in B_t$. In the former—peculiar[18]—situation every operator Φ_γ is necessarily $\mathcal{I}$-functional, since then $\langle \mathcal{I}(\beta_1), \ldots, \mathcal{I}(\beta_n)\rangle = \langle \mathcal{I}(\beta'_1), \ldots, \mathcal{I}(\beta'_n)\rangle$ entails that $\langle \beta_1, \ldots, \beta_n\rangle = \langle \beta'_1, \ldots, \beta'_n\rangle$, so that $\Phi^{\mathcal{I}}_\gamma$ inherits its being a function from Φ_γ. In the latter situation, where for some $t \in T$, $\beta \in B_t$ and $\beta' \in B_t$ it holds that $\mathcal{I}(\beta) = \mathcal{I}(\beta')$, it can be noted that the operator $\Phi_\gamma : B_t \to B_t$ such that $\Phi_\gamma(\beta) = \beta'$, $\Phi_\gamma(\beta') = \beta$ and $\Phi_\gamma(\beta'') = \beta''$ for $\beta'' \in (B_t - \{\beta, \beta'\})$ is $\mathcal{I}$-functional but non-polynomial.[19]

Nonetheless, there are good reasons for disregarding operators over the logical algebra B that are only $\mathcal{I}$-functional for *some* homorphism $\mathcal{I}$ from B to $\mathcal{S}$. For even though formal logical languages B usually come with a particular class of models $\mathcal{M}$ which determines a specific semantic algebra $\mathcal{S}$ and a specific homomorphism $\mathcal{I}$ from B to $\mathcal{S}$,[20] this is generally not the class of models in which the translations of the expressions in the syntactic term algebra are interpreted. This is because most Montague grammar fragments contain a set MP of so-called *meaning postulates*,[21] sentences of the logical language which are intended to reduce the class $\mathcal{M}$ of all models to the subclass $\mathcal{M}^{MP}$ of models in which all meaning postulates in MP are true:[22]

$$(13) \quad \mathcal{M}^{MP} = \{m \in \mathcal{M} \mid \text{for all } \varphi \in MP: in_m(\varphi) = \{\langle a, 1\rangle \mid a \in A\}\}$$

The interpretation $\mathcal{I}(\beta)$ of logical expressions β is reduced accordingly:

$$(14) \quad \mathcal{I}^{MP}(\beta) = \{\langle m, in_m(\beta)\rangle \mid m \in \mathcal{M}^{MP}\}.$$

The interpretation function $\mathcal{I}^{MP}$ can be construed as a—surjective—homomorphism from the logical ω-algebra $B = \langle (B_t)_{t \in T}, (K_\delta)_{\delta \in \Delta}\rangle$ to the following

[18] It seems to be characteristic for a logic that it contains at least some equivalent expressions.

[19] That Φ_γ is $\mathcal{I}$-functional follows from the fact that for all β'' and β''' either $\mathcal{I}(\beta'') \neq \mathcal{I}(\beta''')$ or $\mathcal{I}(\Phi_\gamma(\beta'')) = \mathcal{I}(\Phi_\gamma(\beta'''))$. That Φ_γ is non-polynomial is shown in Hendriks 1993, footnote 44, p. 171. Hendriks 1993, footnote 46, p. 171, provides an example of such an operator.

[20] There is some latitude. E.g., typed logics have 'standard' as well as 'generalized' models, etc.

[21] See Montague 1973, pp. 263–4, for the nine meaning postulates of the PTQ fragment.

[22] This is only true for extensional logics, where sentences denote (a constant function from assignments to) a truth value in every model, viz. 1 ($\{\langle a, 1\rangle \mid a \in A\}$) or 0 ($\{\langle a, 0\rangle \mid a \in A\}$). In the case of intensional logics, where the interpretation of a sentence in a model is (a constant function from assignments to) a function from (sequences of) indices to truth values, the class $\mathcal{M}$ of all models is reduced to the subclass $\mathcal{M}^{MP}$ of models in which all meaning postulates in MP are *valid*, i.e., the class of models in which they denote (the constant function from assignments to) the constant function from (sequences of) indices to the truth value 1. Meaning postulates are meant to restrict 'the interpretations of the [non-logical] constants of the logic' (Janssen 1986, Part 1, p. 98). Since the boundaries are not always clear, we adopt the liberal policy of accepting any expression of type t without free variables as a legitimate meaning postulate. (Given their function of reducing the class of models, for that matter, it does not even seem essential that meaning postulates are expressions of—or expressible in—the logical language.)

semantic algebra $\mathcal{S}^{MP}$ of signature ω:

(15) $\mathcal{S}^{MP} = \langle (\mathcal{I}_t^{MP})_{t\in T}, (\mathcal{G}_\delta^{MP})_{\delta\in\Delta}\rangle$, where
(a) $\mathcal{I}_t^{MP} = \{\mathcal{I}^{MP}(\beta) \mid \beta \in B_t\}$; and
(b) if $\omega(\delta) = \langle\langle t_1,\ldots,t_n\rangle, t_{n+1}\rangle$, then $\mathcal{G}_\delta^{MP} : \mathcal{I}_{t_1}^{MP} \times \ldots \times \mathcal{I}_{t_n}^{MP} \to \mathcal{I}_{t_{n+1}}^{MP}$, where $\mathcal{G}_\delta^{MP}(\mathcal{I}^{MP}(\beta_1),\ldots,\mathcal{I}^{MP}(\beta_n)) = \mathcal{I}^{MP}(K_\delta(\beta_1,\ldots,\beta_n))$.

The addition of meaning postulates affects the class of $\mathcal{I}$-functional operators in a fairly inscrutable manner: given an initial homomorphism $\mathcal{I}$ and some set *MP* of meaning postulates, the $\mathcal{I}^{MP}$-functionality of an operator over a logical algebra B cannot be straightforwardly predicted from its $\mathcal{I}$-functionality.[23] That is: the fact that Φ_γ is $\mathcal{I}$-functional does not entail that it will be $\mathcal{I}^{MP}$-functional, while an operator Φ_γ which fails to be $\mathcal{I}$-functional may very well be $\mathcal{I}^{MP}$-functional. (Examples are provided in Hendriks 1993, pp. 173–4.) Hence it is a safe strategy to allow only those operators over B which are $\mathcal{I}$-functional for *all* homomorphisms $\mathcal{I}$. We noted above that the class of these universally $\mathcal{I}$-functional operators always includes the polynomial operators over the logical algebra B. Moreover, for the languages of typed logic which are commonly used in Montague grammar fragments and of which the syntax constitutes a free algebra B in which each type contains infinitely many generators (viz., the variables of that type), there is a complete characterization of this class, since it can be shown for such algebras that the polynomial operators over B actually exhaust the class of universally $\mathcal{I}$-functional operators.[24] The restriction to universally $\mathcal{I}$-functional—i.e., polynomial—operators can be incorporated as follows. Let $B = \langle (B_t)_{t\in T}, (K_\delta)_{\delta\in\Delta}\rangle$ be an algebra, let POL^B denote the set of polynomial symbols over B, and let $p^\#$ denote the polynomial operator determined by a polynomial symbol p.[25] Then $\Pi(B)$, the *polynomial closure* of B, is the following algebra:

(16) $\langle (B_t)_{t\in T}, \{p^\# \mid p \in \mathrm{POL}^B\}\rangle$

Observe that if $\mathcal{I}$ is a homomorphism from an algebra B to an algebra C, then $\mathcal{I}$ is also a homomorphism from $\Pi(B)$, the polynomial closure of B, to $\Pi(C)$, the polynomial closure of C. This holds on account of the fact that there is a function τ from the polynomial symbols over B to the polynomial symbols over C such that for all $\beta_1,\ldots,\beta_n$: $\mathcal{I}(p_\delta^\#(\beta_1,\ldots,\beta_n)) = (\tau(p_\delta))^\#(\mathcal{I}(\beta_1),\ldots,\mathcal{I}(\beta_n))$.[26]

[23] Some results in this area can be distilled from Van Benthem (1980), Section 3.

[24] For the proof, cf. Hendriks 1993, footnote 52, pp. 174–5, and Janssen 1986, Part 1, pp. 189–92.

[25] These notions are defined in the Appendix.

[26] See the Appendix for the definition of τ. Note that if $\mathcal{I}$ is an (=,=)-homomorphism from B to C, then $\mathcal{I}$ is an $(=,\tau)$-homomorphism from $\Pi(B)$ to $\Pi(C)$.

The above considerations lead to the situation depicted in (17):

(17)

$$\begin{array}{cc} & T_{A,H} \\ & \downarrow tr \\ B & \Pi(B) \\ \downarrow \mathcal{I} & \downarrow \mathcal{I}^{MP} \\ \mathcal{S} & \Pi(\mathcal{S}^{MP}) \end{array}$$

In (17), B represents a logical algebra $\langle (B_t)_{t\in T}, (K_\delta)_{\delta\in\Delta}\rangle$ which is interpreted on the basis of a class of models $\mathcal{M}$: the interpretation $\mathcal{I}(\beta)$ of each logical expression β is a function which associates each $m \in \mathcal{M}$ with $in_m(\beta)$, the interpretation of β in m. The interpretation function $\mathcal{I}$ is an $(=,=)$-homomorphism from B to a semantic algebra $\mathcal{S} = \langle (\mathcal{I}_t)_{t\in T}, (\mathcal{G}_\delta)_{\delta\in\Delta}\rangle$, where $\mathcal{I}_t = \{\mathcal{I}(\beta) \mid \beta \in B_t\}$ and for all $\delta \in \Delta$: $\mathcal{G}_\delta(\mathcal{I}(\beta_1), \ldots, \mathcal{I}(\beta_n)) = \mathcal{I}(K_\delta(\beta_1, \ldots, \beta_n))$.

In the right-hand side of (17), $T_{A,H}$ represents the term algebra of a syntactic algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ with generating family $H = (H_s)_{s\in S}$, while $\Pi(B)$ represents the polynomial closure $\langle (B_t)_{t\in T}, \{p^\# \mid p \in \mathrm{POL}^B\}\rangle$ of the logical algebra B. Every sort s of $T_{A,H}$ is assigned a sort $\sigma(s)$ of $\Pi(B)$, every operator F^T_γ of $T_{A,H}$ is assigned an operator $p^\#_{\rho(\gamma)}$ of $\Pi(B)$, and the translation function tr is a (σ,ρ)-homomorphism from $T_{A,H}$ to $\Pi(B)$. The set of meaning postulates MP restricts the class $\mathcal{M}$ of models for the logical algebra B to the class $\mathcal{M}^{MP}$ of models in which all meaning postulates in MP are true (or valid), and the restriction $\mathcal{I}^{MP}$ of $\mathcal{I}$ to $\mathcal{M}^{MP}$ is defined by $\mathcal{I}^{MP}(\beta) = \{\langle m, in_m(\beta)\rangle \mid m \in \mathcal{M}^{MP}\}$. Since $\mathcal{I}^{MP}$ is an $(=,=)$-homomorphism from B to the semantic algebra $\mathcal{S}^{MP} = \langle (\mathcal{I}^{MP}_t)_{t\in T}, (\mathcal{G}^{MP}_\delta)_{\delta\in\Delta}\rangle$, where $\mathcal{I}^{MP}_t = \{\mathcal{I}^{MP}(\beta) \mid \beta \in B_t\}$ and for all $\delta \in \Delta$: $\mathcal{G}^{MP}_\delta(\mathcal{I}^{MP}(\beta_1), \ldots, \mathcal{I}^{MP}(\beta_n))$ $= \mathcal{I}^{MP}(K_\delta(\beta_1, \ldots, \beta_n))$, we have that $\mathcal{I}^{MP}$ is an $(=,\tau)$-homomorphism from $\Pi(B)$ to the polynomial closure $\Pi(\mathcal{S}^{MP}) = \langle (\mathcal{I}^{MP}_t)_{t\in T}, \{p^\# \mid p \in \mathrm{POL}^{\mathcal{S}^{MP}}\}\rangle$ of $\mathcal{S}^{MP}$, with τ as specified above. Consequently, the composition $tr \circ \mathcal{I}^{MP}$ of the translation homomorphism tr and the interpretation homomorphism $\mathcal{I}^{MP}$ is a $(\sigma\circ{=}, \rho\circ\tau)$-homomorphism from the syntactic term algebra $T_{A,H}$ to the semantic algebra $\Pi(\mathcal{S}^{MP})$.

4 Conclusion

Summing up, the main advantage of the picture sketched in (17) over the approach outlined in (9) above seems to be that there is no need for a separate process of explicitly deriving algebras. On the one hand, there is a model-theoretically interpreted logic which determines the translation algebra. On the other hand, there is a grammar fragment consisting of a generated syntactic algebra, a translation homomorphism from its term algebra to the translation algebra, and a set of meaning postulates. Given the grammar fragment, both the interpretation algebra and the interpretation homomorphism from the translation algebra to the interpretation algebra are induced automatically. This makes the relationship between the grammar of our fragment and the logic that we use in specifying its semantics not only more perspicuous, but also more general: there is no need to readjust our logical tools to every fragment in which we may wish to employ them, apparently as intended by Montague, who 'viewed the use of an intermediate language as motivated by [...] the expectation (which has been amply realized in practice) that a sufficiently well-designed language such as his Intensional Logic with a known semantics could provide a convenient tool for giving the semantics of various fragments of various natural languages.' (Partee 1997, p. 24.)

Appendix: Many-Sorted Algebra

The notion 'many-sorted algebra of signature π' ('π-algebra') is defined in (1) above. A *subalgebra* of π-algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ is a π-algebra $\langle (B_s)_{s\in S}, (F'_\gamma)_{\gamma\in\Gamma}\rangle$ such that $B_s \subseteq A_s$ for all $s \in S$, and for all $\gamma \in \Gamma$: if $\pi(\gamma) = \langle\langle s_1, \ldots, s_n\rangle, s_{n+1}\rangle$, then F'_γ is the *restriction* of F_γ to $(B_s)_{s\in S}$, that is: $F'_\gamma = F_\gamma \cap ((B_{s_1} \times \ldots \times B_{s_n}) \times B_{s_{n+1}})$. Thus, a subalgebra of an algebra A involves a collection of subsets of the carriers of A which is closed under the restrictions of the operators of A to those subsets. Let $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ be a π-algebra that *includes* $H = (H_s)_{s\in S}$, that is: $H_s \subseteq A_s$ for all $s \in S$. The smallest subalgebra $\langle (B_s)_{s\in S}, (F'_\gamma)_{\gamma\in\Gamma}\rangle$ of A that includes H is called the *subalgebra generated by* H, which we will write as $\langle [H], (F'_\gamma)_{\gamma\in\Gamma}\rangle$, where $[H]$ indicates the indexed family $(B_s)_{s\in S}$ of carriers of that subalgebra, and for $F'_\gamma \in (F'_\gamma)_{\gamma\in\Gamma}$, F'_γ is the restriction of F_γ to $(B_s)_{s\in S}$. There is always a unique algebra $\langle [H], (F'_\gamma)_{\gamma\in\Gamma}\rangle$, since it can be characterized as the intersection of all subalgebras of A that include H. Moreover, let algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ include $H = (H_s)_{s\in S}$. Then H is a *generating family* for A (or: A is *generated by* H) iff $\langle [H], (F'_\gamma)_{\gamma\in\Gamma}\rangle = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$.

If a π-algebra $\langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ generated by $(H_s)_{s\in S}$ has the following three properties, it is called a *free* algebra: (1) the members of the generating family $(H_s)_{s\in S}$ are not in the range of some operator F_γ in $(F_\gamma)_{\gamma\in\Gamma}$: if $a_{n+1} \in H_{s_{n+1}}$, then for all F_γ with $\pi(\gamma) = \langle\langle s_1, \ldots, s_n\rangle, s_{n+1}\rangle$ and for all $a_1 \in A_{s_1}, \ldots, a_n \in A_{s_n}$: $a_{n+1} \neq F_\gamma(a_1, \ldots, a_n)$; (2) the operators in $(F_\gamma)_{\gamma\in\Gamma}$ are injections that have disjoint ranges: if $F_\gamma(a_1, \ldots, a_n) = F_{\gamma'}(a'_1, \ldots, a'_m)$, then $\langle a_1, \ldots, a_n\rangle = \langle a'_1, \ldots, a'_m\rangle$ and $F_\gamma = F_{\gamma'}$; and (3) every member of a member of $(A_s)_{s\in S}$ is a member of exactly one carrier A_s: if $a \in A_s$ and $a \in A_{s'}$, then $s = s'$.[27]

(18) Let π-algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ be generated by $H = (H_s)_{s\in S}$. Then $T_{A,H}$, the *term algebra of A with respect to H*, is the π-algebra $\langle (T_{A,H,s})_{s\in S}, (F^T_\gamma)_{\gamma\in\Gamma}\rangle$, where for all $s \in S$ and for all $\gamma \in \Gamma$:
(*a*) $T_{A,H,s}$ is the smallest set such that $\{\lfloor h\rfloor_s \mid h \in H_s\} \subseteq T_{A,H,s}$, and if $t_1 \in T_{A,H,s_1}, \ldots, t_n \in T_{A,H,s_n}$ and $\pi(\gamma) = \langle\langle s_1, \ldots, s_n\rangle, s\rangle$, then $F^T_\gamma(t_1, \ldots, t_n) \in T_{A,H,s}$; and (*b*) $F^T_\gamma(t_1, \ldots, t_n) = \lfloor\gamma\ t_1 \ldots t_n\rfloor_s$.

[27] Clause (1) and (2) constitute the definition of the notion of a free algebra for traditional ('one-sorted') algebras (cf. Montague 1970, p. 225). Clause (3) is a natural addition in the context of many-sorted algebras, as extensively argued in Hendriks 1993, footnote 10, p. 141.

Observe that the term algebra $T_{A,H}$ of a π-algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ with generating family $H = (H_s)_{s\in S}$ can be characterized as a generated algebra, viz., as $\langle [\,(\{\lfloor h\rfloor_s | h \in H_s\}_s)_{s\in S}\,], (F^T_\gamma)_{\gamma\in\Gamma}\rangle$, and that it is designed to be a free algebra. The members of the carriers of a term algebra $T_{A,H}$ are related to the members of the carriers of the original algebra A by means of the *evaluation* function ev defined by $ev(\lfloor h\rfloor_s) = h$ and $ev(\lfloor \gamma\; t_1 \ldots t_n\rfloor_s) = F_\gamma(ev(t_1), \ldots, ev(t_n))$: it holds for all $s \in S$ that $a \in A_s$ if and only if there is a term $t \in T_{A,H,s}$ such that $ev(t) = a$. A π-algebra A generated by some family H is (σ, ρ)-interpretable[28] in algebra B if and only if the term algebra $T_{A,H}$ is (σ, ρ)-interpretable in B, for note that A and $T_{A,H}$ invariably have the same set of sorts S, the same set of operator indices Γ, and the same type-assigning function π. An important fact is that the composition of two homomorphisms h and g is again a homomorphism. Suppose that A, B and C are algebras such that A is (σ, ρ)-interpretable in B and B is (σ', ρ')-interpretable in C, that h is a (σ, ρ)-homomorphism from A to B, and that g is a (σ', ρ')-homomorphism from B to C. Then A is $(\sigma\circ\sigma', \rho\circ\rho')$-interpretable in C and $h\circ g$ is a $(\sigma\circ\sigma', \rho\circ\rho')$-homomorphism from A to C. The proof is straightforward.

The set of *polynomial operators* over a π-algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ consists of projection functions and constant functions plus operators which are definable in terms of these elementary operators and the operators in $(F_\gamma)_{\gamma\in\Gamma}$. We present a definition in terms of so-called polynomial symbols.[29] Let $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ be a many-sorted algebra of signature π. We first define three sets of auxiliary symbols. Let $s \in S$ and let, for $n \in \mathbb{N}^+$, $\langle\langle s_1, \ldots, s_n\rangle, s\rangle \in S^n \times S$. Then:

(19) $\mathrm{VAR} = \{\xi^i \mid i \in \mathbb{N}^+\}$; $\quad \mathrm{CON}^A_s = \{a \mid a \in A_s\}$; and
$\mathrm{OP}^A_{\langle\langle s_1,\ldots,s_n\rangle,s\rangle} = \{\gamma \mid F_\gamma \in (F_\gamma)_{\gamma\in\Gamma}$ and $\pi(\gamma) = \langle\langle s_1, \ldots, s_n\rangle, s\rangle\}$.

VAR is the set of *polynomial variables*, CON^A_s is the set of *polynomial constants* over A of sort s, and $\mathrm{OP}^A_{\langle\langle s_1,\ldots,s_n\rangle,s\rangle}$ is the set of *polynomial operator symbols* over A of type $\langle\langle s_1, \ldots, s_n\rangle, s\rangle$. For $n \in \mathbb{N}^+$ and $\langle \vec{s}, s\rangle \in S^n \times S$, the set $\mathrm{POL}^A_{\langle\vec{s},s\rangle}$ of *polynomial symbols* over A of type $\langle\vec{s}, s\rangle$ is defined as the smallest set such that:

(20) if $\xi^i \in \mathrm{VAR}$ and $\vec{s} = \langle s_1, \ldots, s_{i-1}, s, s_{i+1} \ldots, s_n\rangle$, then $\lfloor\xi^i\rfloor_s \in \mathrm{POL}^A_{\langle\vec{s},s\rangle}$;
if $a \in \mathrm{CON}^A_s$, then $\lfloor a\rfloor_s \in \mathrm{POL}^A_{\langle\vec{s},s\rangle}$; and if $p_1 \in \mathrm{POL}^A_{\langle\vec{s},s_1\rangle}, \ldots,$
$p_k \in \mathrm{POL}^A_{\langle\vec{s},s_k\rangle}$ and $\gamma \in \mathrm{OP}^A_{\langle\langle s_1,\ldots,s_k\rangle,s\rangle}$, then $\lfloor\gamma\; p_1 \ldots p_k\rfloor_s \in \mathrm{POL}^A_{\langle\vec{s},s\rangle}$.

We let $\mathrm{POL}^A = \bigcup\{\,\mathrm{POL}^A_{\langle\vec{s},s\rangle} \mid \langle\vec{s}, s\rangle \in S^n \times S$ and $n \in \mathbb{N}^+\}$. A polynomial symbol $p \in \mathrm{POL}^A_{\langle\vec{s},s\rangle}$ uniquely determines a *polynomial operator* $p^\#$ of type

[28] The notions '(σ, ρ)-interpretability' and '(σ, ρ)-homomorphism' are defined in (3) and (4).

[29] Cf. Janssen (1986, pp. 56–61). A more direct—one-sorted—definition is given in Montague (1970, p. 224). The mediation of polynomial symbols leads to a more neatly arranged bureaucracy in the many-sorted context.

$\langle \vec{s}, s\rangle$. Let $\vec{s}$ be the sequence $\langle s_1, \ldots, s_n\rangle$ and let $a_1 \in A_{s_1}, \ldots, a_n \in A_{s_n}$. Then:

(21) $\lfloor \xi^i \rfloor_s^{\#}(a_1, \ldots, a_n) = a_i$; $\lfloor c \rfloor_s^{\#}(a_1, \ldots, a_n) = c$; and
$\lfloor \gamma\ p_1 \ldots p_k \rfloor_s^{\#}(a_1, \ldots, a_n) = F_\gamma(p_1^{\#}(a_1, \ldots, a_n), \ldots, p_k^{\#}(a_1, \ldots, a_n))$.

Thus (i) polynomial symbols $\lfloor \xi^i \rfloor_s$ of type $\langle\langle s_1, \ldots s_n\rangle, s\rangle$ determine a *projection function* $\lfloor \xi^i \rfloor_s^{\#} : A_{s_1} \times \ldots \times A_{s_n} \to A_s$ which yields its i-th argument as a result; (ii) polynomial symbols $\lfloor c \rfloor_s$ of type $\langle\langle s_1, \ldots s_n\rangle, s\rangle$ determine a *constant function* $\lfloor c \rfloor_s^{\#} : A_{s_1} \times \ldots \times A_{s_n} \to A_s$ which yields $c \in A_s$ as a result; and (iii) polynomial symbols $\lfloor \gamma\ p_1 \ldots p_k \rfloor_s$ of type $\langle\langle s_1, \ldots s_n\rangle, s\rangle$ determine a function $\lfloor \gamma\ p_1 \ldots p_k \rfloor_s^{\#} : A_{s_1} \times \ldots \times A_{s_n} \to A_s$ which is a *composition* of F_γ and the functions $p_1^{\#}, \ldots, p_k^{\#}$ determined by the respective polynomial symbols $p_1, \ldots, p_k$.[30]

Let $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ be a π-algebra with generating family $H = (H_s)_{s\in S}$. It can be shown that polynomial operators over A are h-functional for all homomorphisms h from A to some algebra B. That is: let $B = \langle (B_t)_{t\in T}, (G_\delta)_{\delta\in\Delta}\rangle$ be an arbitrary ω-algebra such that A is (σ, ρ)-interpretable in B for some $\sigma : S \to T$ and $\rho : \Gamma \to \Delta$, and let h be a (σ, ρ)-homomorphism from A to B. Then for all polynomial operators $p^{\#}$ over A it holds that $(p^{\#})^h$, the relation h-induced by $p^{\#}$ as defined in (22), is a function.

(22) $(p^{\#})^h = \{\langle\langle h(a_1), \ldots, h(a_n)\rangle, h(a_{n+1})\rangle \mid \langle\langle a_1, \ldots, a_n\rangle, a_{n+1}\rangle \in p^{\#}\}$

This claim is proven by defining a function τ which assigns a polynomial symbol $\tau(p) \in \mathrm{POL}^B_{\langle\langle\sigma(s_1),\ldots\sigma(s_n)\rangle,\sigma(s)\rangle}$ to each polynomial symbol $p \in \mathrm{POL}^A_{\langle\langle s_1,\ldots s_n\rangle,s\rangle}$ and by showing that for all $a_1 \in A_{s_1}, \ldots, a_n \in A_{s_n}$ it holds that $h(p^{\#}(a_1, \ldots, a_n)) = (\tau(p))^{\#}(h(a_1), \ldots, h(a_n))$. The latter means that $(p^{\#})^h$ is nothing but the set $\{\langle\langle h(a_1), \ldots, h(a_n)\rangle, (\tau(p))^{\#}(h(a_1), \ldots, h(a_n))\rangle \mid \langle\langle a_1, \ldots, a_n\rangle, a_{n+1}\rangle \in p^{\#}\}$, which, of course, cannot fail to be a function. The function τ is defined by $\tau(\lfloor \xi^i \rfloor_s) = \lfloor \xi^i \rfloor_{\sigma(s)}$; $\tau(\lfloor c \rfloor_s) = \lfloor h(c) \rfloor_{\sigma(s)}$; and $\tau(\lfloor \gamma\ p_1 \ldots p_k \rfloor_s) = \lfloor \rho(\gamma)\ \tau(p_1) \ldots \tau(p_k) \rfloor_{\sigma(s)}$. The proof proceeds by a straightforward induction on the complexity of $p \in$

[30]Projection functions yield one of their arguments as a result, constant functions yield a member of a carrier as a result, and the collection $(A_s)_{s\in S}$ is closed under the operations $F_\gamma \in (F_\gamma)_{\gamma\in\Gamma}$. Hence the addition of polynomial operators to an algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ always yields an algebra with the same carriers as A itself. Note that the operators F_γ in $\langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ are themselves expressed by a polynomial symbol: if $\pi(\gamma) = \langle\langle s_1, \ldots, s_n\rangle, s\rangle$, then $F_\gamma = \lfloor \gamma\ \lfloor \xi^1 \rfloor_{s_1} \ldots \lfloor \xi^n \rfloor_{s_n} \rfloor_s^{\#}$, where $\lfloor \xi^1 \rfloor_{s_1}, \ldots, \lfloor \xi^n \rfloor_{s_n}$ have the respective types $\langle\langle s_1, \ldots, s_n\rangle, s_1\rangle, \ldots, \langle\langle s_1, \ldots, s_n\rangle, s_n\rangle$. (Incidentally, the members of the carriers of the term algebra $T_{A,H}$ of a π-algebra $A = \langle (A_s)_{s\in S}, (F_\gamma)_{\gamma\in\Gamma}\rangle$ with respect to generating family $H = (H_s)_{s\in S}$ are precisely the polynomial symbols which can be built up from polynomial constants corresponding to members of members of $(H_s)_{s\in S}$ and operator symbols corresponding to members of $(F_\gamma)_{\gamma\in\Gamma}$, but contain no polynomial variables).

POL^A:

$$h(\lfloor\xi^i_s\rfloor^\#(a_1,\ldots,a_n)) = h(a_i) = \lfloor\xi^i_{\sigma(s)}\rfloor^\#(h(a_1),\ldots,h(a_n)) =$$
$$(\tau(\lfloor\xi^i_s\rfloor))^\#(h(a_1),\ldots,h(a_n));$$
$$h(\lfloor c_s\rfloor^\#(a_1,\ldots,a_n)) = h(c) = \lfloor h(c)_{\sigma(s)}\rfloor^\#(h(a_1),\ldots,h(a_n)) =$$
$$(\tau(\lfloor c_s\rfloor))^\#(h(a_1),\ldots,h(a_n)); \text{ and}$$
$$h(\lfloor\gamma\ p_1\ldots p_k\rfloor^\#(a_1,\ldots,a_n)) =$$
$$h(F_\gamma(p_1^\#(a_1,\ldots,a_n),\ldots,p_k^\#(a_1,\ldots,a_n))) =$$
$$G_{\rho(\gamma)}(h(p_1^\#(a_1,\ldots,a_n)),\ldots,h(p_k^\#(a_1,\ldots,a_n))) =$$
$$G_{\rho(\gamma)}((\tau(p_1))^\#(h(a_1),\ldots,h(a_n)),\ldots,(\tau(p_k))^\#(h(a_1),\ldots,h(a_n))) =$$
$$\lfloor\rho(\gamma)\ \tau(p_1)\ldots\tau(p_k)\rfloor^\#(h(a_1),\ldots,h(a_n)) =$$
$$(\tau(\lfloor\gamma\ p_1\ldots p_k\rfloor))^\#(h(a_1),\ldots,h(a_n)). \text{ [QED]}$$

References

Adj (J.A. Goguen, J.W. Thatcher, E.G. Wagner, J.B. Wright) (1977). 'Initial Algebra Semantics and Continuous Algebras'. *Journal of the Association for Computing Machinery* 24, 68–95.

Benthem, J. van (1980). 'Universal Algebra and Model Theory. Two Excursions on the Border.' Report ZW–7908. Department of Mathematics, Groningen University.

Gamut, L.T.F. (J. van Benthem, J. Groenendijk, D. de Jongh, M. Stokhof, H. Verkuyl) (1991). *Logic, Language and Meaning. Volume I: Introduction to Logic. Volume II: Intensional Logic and Logical Grammar*. University of Chicago Press, Chicago and London.

Halvorsen, P.-K., and W. Ladusaw (1979). 'Montague's "Universal Grammar": an Introduction for the Linguist'. *Linguistics and Philosophy* 3, 185–223.

Hendriks, H. (1993). *Studied Flexibility. Categories and Types in Syntax and Semantics*. ILLC Dissertation Series 1993–5. ILLC, University of Amsterdam.

Janssen, T. (1986). *Foundations and Applications of Montague Grammar. Part 1: Philosophy, Framework, Computer Science. Part 2: Applications to Natural Language*. CWI Tracts 19 and 28, Amsterdam.

Montague, R. (1970). 'Universal Grammar'. *Theoria* **36**, 373–398. Page references concern the reprint as Chapter 7 of R. Thomason (ed.) (1974), *Formal Philosophy. Selected Papers of Richard Montague*. Yale University Press, New Haven.

Montague, R. (1973). 'The Proper Treatment of Quantification in Ordinary English'. In J. Hintikka, J. Moravcsik and P. Suppes (eds.) (1973), *Approaches to Natural Language. Proceedings of the 1970 Stanford Workshop on Grammar and Semantics*. Reidel, Dordrecht. Page references concern the reprint as Chapter 8 of Thomason (ed.) (1974), *Formal Philosophy. Selected Papers of Richard Montague*. Yale University Press, New Haven.

Partee, B. (1984). 'Compositionality'. In F. Landman and F. Veltman (eds.) (1984), *Varieties of Formal Semantics. Proceedings of the Fourth Amsterdam Colloquium*. Foris, Dordrecht.

Partee, B., with H. Hendriks (1997). 'Montague Grammar'. Chapter 1 in J. van Benthem and A. ter Meulen (eds.), *Handbook of Logic and Language*, Elsevier Science Publishers, Amsterdam, 5–92.