A Coalgebraic Modelling of Head-Driven Phrase Structure Grammar

Stephan Kepser
SFB 441, University of Tübingen
Wilhelmstr. 113 72074 Tübingen, Germany
kepser@sfs.nphil.uni-tuebingen.de

Abstract

This paper provides a coalgebraic modelling of Head-Driven Phrase Structure Grammar. HPSG is a licensing theory in the sense that grammaticality is defined in terms of abstract grammar principles, well-formedness conditions on linguistic analyses rather then by a set of rules that would explicitely generate an analysis. Since coalgebras are very well suited for modelling licensing theories, we provide a conceptually particularly adequate formalisation for HPSG by showing that HPSG can indeed be modeled with coalgebras. We also show that the category of grammar models contains a final coalgebra. This final coalgebra we propose as *the* model of an HPSG grammar, because it models the analyses of the structurally different readings of all and only those utterances licensed by the grammar while eliminating spurious ambiguities.

1 Introduction

Linguistic theories in general fall into one out of two classes. They can be *generating* theories in the sense that the grammar actually generates utterances and analyses of utterances. Typical examples of such theories are the early transformational grammar framework or GPSG. The second class of theories are the *licensing* theories. These theories do not generate utterances or their analyses. Rather they define well-formedness conditions often layed out as abstract general principles. An analysis of an utterance is admissible, if it obeys to all these principles. How this analysis is originally obtained is of no concern for the grammar, it may "fall from heaven".

HPSG clearly is such a licensing theory as the whole grammar is given as a set of universal and language specific principles. We also know that this appearance of the grammar is not merely a coincidence, its authors intended it to look that way. So, for example, Pollard and Sag (1994) write on page 31:

> "But how do we specify the well-formed phrases of a given language? The answer we give is similar to the one given in GB theory: a candidate phrase will be well formed provided it satisfies all the principles of grammar, including both universal principles and language-specific principles."

We think it is therfore an important task of any formalisation of HPSG to capture the idea of licensing within its main concepts. And in particular, we think these concepts should differ from those used in the formalisation of generating theories to be able to capture the difference between these two types of linguistic theories within the way models look.

As the name implies, in a generating theory, the internal structure of a linguistic analysis is generated by the grammar. Consequently, the internal structure is completely visible for the outside world. A linguist does not only know what properties the internal structure may have, ideally, he knows exactly what it looks like and works directly with it.

If a licensing theory sets out well-formedness conditions, we need a different concept. The internal structure of a linguistic analysis is no longer of importance. We do not need to know it. What the linguist needs to know instead are only certain observable properties of the analysis. Because whatever the internal structure of the analysis may be, as long as its observable properties, its features (in the non-technical sense) are compatible with the well-formedness conditions the grammar sets out, the analysis is grammatical. The internal structure becomes a hidden, because finally irrelevant aspect of the analysis.

That does not mean that a licensing theory does not have any data structures for linguistic analyses. They sure have, and, as we know, HPSG uses feature structures. But how these feature structures are internally constructed and represented is no longer relevant. What counts are their formal, observable properties, that is their sorts and features and the relations of the two. An HPSG principle like the Head Feature Principle makes a statement on the relation of certain sorts and features. And if we can observe that this relation holds in a given feature structure, we know it obeys to the principle. The principle makes no statement how it should come along that the relation is true for the feature structure in question. It does not care about the "internal affairs" of the feature structure, as long as the relation holds true.

Hence for a modelling of HPSG that conforms with the idea of a licensing theory we need a formal framework that allows us to talk about observable properties of data structures in a mathematically precise sense. Universal algebra provides a modelling concept that does not deal with constructing data types but with observing properties of data types, namely the concept of (final) coalgebras and coinduction. It is this concept that we propose for modelling HPSG.

2 Coalgebras

The theory of coalgebras is a methodology from universal algebra (see Cohn (1965); Grätzer (1979)) and theoretical computer science to make the notions of observation and observability formally precise. In this introduction to coalgebras, we will need some simple notions from category theory that we cannot introduce here. The interested reader is referred to Pierce (1991), for example. Our explanation of coalgebras closely follows the introductory paper by Jacobs and Rutten (1997).

Definition 1 Let T be a functor. A *coalgebra* is a pair (U, c) consisting of a set U and a function $c : U \to T(U)$.

The set U is called *universe*, *carrier* or *state space*. The function c is called the *structure* of the coalgebra.

Example 2 A typical example of a system that can fruitfully be modeled as a coalgebra is a state transition system. The universe U is the set of different states the system can be in. The system is a black box, the internal structure of a state is hidden and unavailable from outside. All we have are observers that can tell us some information about states. In our simple example, we suppose states to have a colour and potentially a successor state. These properties can be observed with the observers $\texttt{colour} : U \to \{\mathsf{red}, \mathsf{green}, \mathsf{blue}\}$ and $\texttt{next} : U \to U \uplus \{*\}$ where $*$ is used to indicate that a state has no successor. Since the internal representation is unavailable, what we can see from outside about a state u is $\texttt{colour}(u)$, $\texttt{colour}(\texttt{next}(u))$, $\texttt{colour}(\texttt{next}(\texttt{next}(u)))$, and so on. What we perceive of a state is a sequence of colours, the colour of the state and the colours of its successors. So, each state is associated with an observation chain, e.g., $(\mathsf{red},\mathsf{red},\mathsf{blue},\mathsf{red},\mathsf{green},\mathsf{blue},\mathsf{green})$. It can be that in a given coalgebra, two states have the same observation chain. That is there can be two different states $u_1 \neq u_2$, which are still such that $\texttt{colour}(u_1) = \texttt{colour}(u_2)$ and $\texttt{colour}(\texttt{next}^n(u_1)) = \texttt{colour}(\texttt{next}^n(u_2))$ for all n. That means an outside observer cannot distinguish the two states. For him, they appear to be one and the same. So, why should there be two different states at all? The answer is that there is not just one model of the state space, rather there exists a whole collection of them. The collection are all the coalgebras for the given functor.

In this example, the functor is $T(X) = 1 + X \times \{\mathsf{red}, \mathsf{green}, \mathsf{blue}\}$. There are many different coalgebras for this functor, all of them modelling the state transition system in a different way. Together with homomorphisms, they form a category of coalgebras.

Definition 3 Let T be a functor.
A *homomorphism* of coalgebras from a T-coalgebra (U_1, c_1) to another T-coalgebra (U_2, c_2) consists of a function $f : U_1 \to U_2$ between the carrier sets which commutes with the operations: $c_2 \circ f = T(f) \circ c_1$ as expressed by the following diagram.

$$\begin{tikzcd} U_1 \arrow[r, "f"] \arrow[d, "c_1"'] & U_2 \arrow[d, "c_2"] \\ T(U_1) \arrow[r, "T(f)"'] & T(U_2) \end{tikzcd}$$

A *final coalgebra* (W, d) is a coalgebra such that for every algebra (U, c) there exists a unique homomorphism $f : (U, c) \to (W, d)$.

To continue the example, there exists a distinguished T-coalgebra in the category of coalgebraic models of our state transition system. It contains states for all and only those observations that can be made. Hence there are no different, but indistinguishable states, each chain of observations belongs to a unique element. This coalgebra is the final coalgebra. The carrier of the coalgebra is the set of all finite and infinite sequences of the colours $\{\mathsf{red}, \mathsf{green}, \mathsf{blue}\}$. The `colour` of such a state is just the first element or the head of the sequence. And if the sequence is more than one element long, then the successor state is the tail of the sequence, that is the sequence obtained by deleting the leftmost element of the given sequence. If the sequence is just one element long, it has no successor. So, e.g., $\mathtt{colour}((\mathsf{red,red,blue,red,green,blue,green})) = \mathsf{red}$ and $\mathtt{next}((\mathsf{red,red,blue,red,green,blue,green})) = (\mathsf{red,blue,red,green,blue,green})$.

To see that this coalgebra is final, consider a T-coalgebra (U, c). As explained above, each state $u \in U$ is assigned a unique observation chain. This observation chain obviously exists, too, in the final coalgebra. Hence we map each state to its observation chain. This map is clearly a homomorphism: The colour of a state is the same as the first colour of its observation chain. And the observation chain of the successor of a state is the observation chain of the state with the leftmost element removed.
To see that this homomorphism is unique, consider any function from the set of states to sequences of colours. In order to be a homomorphism it must firstly map a state to a sequence that starts with the colour of that state. And further, if the state has a successor then it must be mapped to a sequence that has as second element the colour of the successor. If it has no successor, it can only be mapped to the one element sequence that consists just of the colour of the state. Now, we can iterate the argument with the potential successor, and we see that there exists indeed only a single homomorphism from the set of states to sequences of colours namely the one that maps each state to its observation chain.

Final coalgebras play a role in coalgebraic theory that is dual to the role of initial algebras in algebraic theory. They provide coinduction, the dual of induction, a means for defining functions on and proving properties of coalgebras. Hence the existence of a final coalgebra is similarly desirable and equally important as the existence of an initial algebra.

Proposition 4 *(i) Final coalgebras, if they exist, are uniquely determined (up to isomorphism).*
(ii) A final coalgebra $W \to T(W)$ is a fixed point $W \stackrel{\cong}{\to} T(W)$ of the functor T.

3 Speciate Re-entrant Logic

Speciate Re-entrant Logic (SRL, see King (1989, 1999)) is a complete (King (1989)) and decidable (Kepser (1994)) sorted feature logic for HPSG developed by Paul King. Here we review only those aspects of the formal language of SRL that are germane to the present paper, and do not discuss the logical properties of SRL. We follow mainly the exposition of SRL as given in King et al. (1999).

Definition 5 A *signature* $\Sigma = (\mathcal{S}, \mathcal{F}, \mathcal{A})$ for this logic consists of a set $\mathcal{S}$ of sorts, a set $\mathcal{F}$ of features, and an appropriateness function $\mathcal{A} : \mathcal{S} \times \mathcal{F} \to \wp(\mathcal{S})$[1] that states for each sort which features are appropriate and which sorts may follow these features.

For notational facility we henceforth assume that none of the symbols $:, \sim, =, \neg, \wedge, \vee, \to, ($ or $)$ is a sort or a feature.

Definition 6 An *interpretation* for a signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathcal{A})$ is a triple (U, S, F) where

U is a set,

S is a total function from U to $\mathcal{S}$, and

F is a total function form $\mathcal{F}$ to the set of partial function from U to U, and for each $\alpha \in \mathcal{F}, u \in U$

$F(\alpha)(u)$ is defined iff $\mathcal{A}(S(u), \alpha) \neq \emptyset$, and

if $F(\alpha)(u)$ is defined then $S(F(\alpha)(u)) \in \mathcal{A}(S(u), \alpha)$.

Suppose $I = (U, S, F)$ is a Σ-interpretation. We call U the universe of I and its elements objects; S is the sort assignment function that assigns each object $o \in U$ to a unique sort in $\mathcal{S}$. And F is the feature interpretation function. Each feature is interpreted by a partial function on U. The additional clauses in the definition of F ensure that the interpretation obeys to the appropriateness function $\mathcal{A}$ and that the resulting feature structures are totally well-typed and sort resolved.

The language of the logic is a description language for feature structures. Terms are feature sequences. That is to say the reserved symbol ':' is a term, and if τ is a term and $\alpha \in \mathcal{A}$ a feature then $\tau\alpha$ is a term. We write T_Σ for the set of all Σ-terms. Let $I = (U, S, F)$ be a structure. The feature interpretation function F can naturally be extended to terms in the following way. $F(:) = \mathrm{id}_U$ and $F(\tau\alpha) = F(\tau) \circ F(\alpha)$ where $\circ$ is composition of partial functions.

Definition 7 Let Σ be a signature. The set $\mathcal{D}_=$ of *descriptions* is defined as follows.

If $\tau \in T_\Sigma$ and $s \in \mathcal{S}$ then $\tau \sim s \in \mathcal{D}_=$,

if $\tau, \tau' \in T_\Sigma$ then $\tau = \tau' \in \mathcal{D}_=$,

if $d \in \mathcal{D}_=$ then $\neg d \in \mathcal{D}_=$,

if $d, d' \in \mathcal{D}_=$ then $(d \wedge d') \in \mathcal{D}_=$,

if $d, d' \in \mathcal{D}_=$ then $(d \vee d') \in \mathcal{D}_=$,

if $d, d' \in \mathcal{D}_=$ then $(d \to d') \in \mathcal{D}_=$.

$\tau \sim s$ is called a sort statement, $\tau = \tau'$ a path equation. These two are atomic descriptions.

The descriptions are supposed to describe elements of the universe of discourse. Hence they are interpreted as true or false of an object.

Definition 8 Let $I = (U, S, F)$ be an interpretation. The denotation of a description is defined as follows ($\tau, \tau' \in T_\Sigma, s \in \mathcal{S}, d, d' \in \mathcal{D}_=$):

[1] We write $\wp(M)$ for the power set of a set M.

$$\begin{aligned}
&[\![\tau \sim s]\!] = \{u \in U \mid F(\tau) \text{ is defined on } u \text{ and } S(F(\tau)(u)) = s\},\\
&[\![\tau = \tau']\!] = \left\{ u \in U \;\middle|\; \begin{array}{l} F(\tau) \text{ and } F(\tau') \text{ are defined on } u \\ \text{and } F(\tau)(u) = F(\tau')(u) \end{array} \right\},\\
&[\![\neg d]\!] = U \setminus [\![d]\!],\\
&[\![d \wedge d']\!] = [\![d]\!] \cap [\![d']\!],\\
&[\![d \vee d']\!] = [\![d]\!] \cup [\![d']\!],\\
&[\![d \rightarrow d']\!] = (U \setminus [\![d]\!]) \cup [\![d']\!].
\end{aligned}$$

The denotation of a sort statement is the set of all those objects on which the path τ is defined and for which the object at the end of the path is of sort s. The denotation of a path equation is the set of those objects on which both paths τ and τ' are defined and lead to one and the same object. As one can see, the connectives have classical meaning of set complement, intersection, and union.

Let Σ be a signature , $I = (U, S, F)$ be an interpretation, and d a description. We say I *satisfies* d iff $[\![d]\!] \neq \emptyset$, that is if the description is true of at least one object in the universe. We say I *models* d or I is a *model* of d iff $[\![d]\!] = U$, that is if the description d is true of every object in the universe.

Principles of HPSG can be expressed in the description language. For example, the Head Feature Principle is, simplifying slightly, rendered as

$(:\sim \textit{phrase} \wedge \text{:daughters} \sim \textit{headed-structure}) \rightarrow$
$\text{:synsem local category head} = \text{:daughters head-dtr synsem local category head}$

An HPSG grammar Γ is then a conjunction of descriptions. For a given interpretation I, the grammar is true of some of the entities in its universe, but not necessarily of all. Naturally, the linguist is interested in Γ-models, because interpretations that contain objects violating the principles can never be representations for grammatical utterances or their analyses. And not all Γ-models are equally important. The extreme case of an uninteresting Γ-model is one with an empty universe. Of particular interest are the so-called exhaustive models. To define them, we need the notion of a subinterpretation below an object.

Definition 9 Let $I = (U, S, F)$ be an interpretation and $u \in U$. Define the *subinterpretation* $\langle u \rangle = (U_u, S_u, F_u)$ *below* u as follows

$U_u = \{o \in U \mid \exists \tau \in T_\Sigma : o = F(\tau)(u)\}$,
$S_u = S_{|U_u}, \quad F_u = F_{|U_u}$.[2]

We sometimes call the subinterpretation below u also the substructure below u. u is called its root.

Now, a Γ-model $\mathfrak{A}$ is called an *exhaustive* model iff for every Γ-model $\mathfrak{B}$ and each $u \in U_{\mathfrak{B}}$ there exists an isomorphic image of the substructure $\langle u \rangle$ in $\mathfrak{A}$. In other words, an exhaustive model contains every feature structure that is compatible with the grammar. The strong generative capacity of HPSG can be defined using exhaustive models (Pollard (1999)).

4 SRL-Models as Coalgebras

King himself certainly does not regard SRL-interpretations as coalgebras. But it is very natural to interprete them in this way. The universe of an SRL-interpretation is a kind of hidden state space. And indeed, King makes no claim about the inner structure of the entities in the universe. The sort function replaces the `colour`-function of the introductory example. And instead of a single successor function we now have a whole bunch of them in terms of the features. So, we are not looking at the internal structure of a feature structure. Rather we use sorts and features to make certain observations about it. The appropriateness function $\mathcal{A}$ is a kind of theory for the class of all coalgebras for a signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathcal{A})$. Let the cardinality of the set $\mathcal{F}$ of features be κ. Then the endofunctor is $T(X) = \mathcal{S} \times (X + 1)^\kappa$. An SRL-interpretation (U, S, F) is a coalgebra

[2] We write $f_{|M}$ for the restriction of the function f to the set M.

$\mathsf{srl} : U \to \mathcal{S} \times (U+1)^{\kappa}$. The leftmost component of srl is the sort function S. And for each feature $f \in \mathcal{F}$ there is a function $F(f) : U \to U + 1$ which returns the successor of an element under the feature f, if it is defined, and $*$ otherwise. Hence $\mathsf{srl} = S \times \prod_{f \in \mathcal{F}} F(f)$.

Of course, it is not so much the whole class of all coalgebras for a given signature that is interesting. More interesting is the subclass that for a given grammar Γ contains all and only its Γ-models. With homomorphisms, this subclass forms a category.

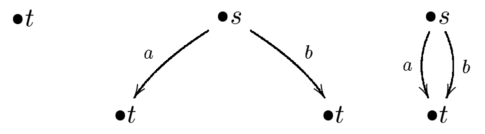
It would be nice, if this category had final coalgebras. But unfortunately, we observe the following

Theorem 10 *In the language $\mathcal{D}_=$, the category of Γ-models does generally not contain final coalgebras.*

We give a simple example with a very small signature and a simple grammar to show that in general the existence of final objects cannot be expected.

Example 11 The signature contains two sorts: s and t, two features: a and b, and the appropriateness function: $s \quad a \quad t \qquad\qquad s \quad b \quad t$.

There are three prototypical structures that are compatible with the given signature:



$A \qquad\qquad B \qquad\qquad C$

Any structure is constructed out of these elements.

Now, let the grammar $\Gamma = \{:\sim s \to : a \neq : b\}$.
This explicitely forbids structure C. Structure A cannot be the final object, because there is no homomorphism from B to A. But structure B cannot be the final object either, because there are two different homomorphisms from A to B. Any other candidate for a final object would have to contain an element of sort s in its universe, because otherwise there cannot be a homomorphism from B into the candidate. But then the candidate already contains B as a substructure, and is therefore unsuitable as final object for the same reason that B is.

The result is not very surprising considering the fact that our description language contains full negation. We can use path disequations in a grammar Γ to express that two elements in the universe have to be different. At the same time, we can set up the signature and the rest of the grammar in such a fashion that these two different entities must be observationally equivalent. We know that a final coalgebra will always identify entities that are observationally equivalent in other coalgebras within the same category. But since the grammar prohibits this identification, the final coalgebra cannot be in the category of Γ-models.

So, we are clearly faced with the fact that the description language is too powerful. Being able to express that there should be two different but observationally equivalent entities violates the spirit of coalgebraic modelling. To solve the problem, we can try and restrict the language. We do so by restricting the use of negation to the negation of atomic sort statements. We thus have $\tau \sim s, \neg\, \tau \sim s$ and $\tau = \tau'$ as statement literals and then conjunction and disjunction (and nothing else) to form complex descriptions. With the language that restricted, we can prove the following

Theorem 12 *The category of Γ-models contains final coalgebras, if only atomic sort statements can be negated.*

It is interesting to note that the language restriction is not as severe as it may seem at first. Since the connectives are interpreted classically, any description can be transformed into an equivalent one where only atomic statements are negated. Thus all we prohibit are path disequations. In particular, implications of the form '$\tau_1 \sim s \rightarrow \tau_2 = \tau_3$' are still allowed. And many principles of HPSG have exactly that form. The Head Feature Principle is the prototypical example for such principle. Path disequations are rarely directly used in Pollard and Sag (1994). The two places where they play an important role are the Binding theory and the Control theory. Although the Binding theory is not completely expressed in terms of feature structures, one can see that path disequations are the means of choice to express that a personal pronoun of Principle B or a nonpronoun of Principle C must be (locally) o-free. And in the Control theory, there is an overt use of a path disequation in so far as there is a path equation in the antecedent of an implication. Thus although quite a few principles can be expressed without path disequations it is finally not possible to capture the whole of HPSG without them. We therefore have to find another way to cope with path disequations, and not just prohibit them.

In general, if path disequations are used in HPSG, then as a convenient means to express that two sub-feature structures are *really* two different structures, that means they should also be observationally different, while the malicious counterexample above demands that two observationally indistinguishable structures be different. Therefore another way to remedy the problem with path disequation is to offer a different semantics for path equations. SRL-interpretations model path equations as true identity. Following the ideas of licensing theories, we propose to model them as observational indiscernability. We introduce a new logical constant $\approx$ and its negation $\not\approx$. As a replacement of Definition 7 we now have the following definition of descriptions.

Definition 13 Let Σ be a signature. The set $\mathcal{D}_\approx$ of *descriptions* is defined as follows.
If $\tau \in T_\Sigma$ and $s \in \mathcal{S}$ then $\tau \sim s \in \mathcal{D}_\approx$,
if $\tau, \tau' \in T_\Sigma$ then $\tau \approx \tau' \in \mathcal{D}_\approx$,
if $d \in \mathcal{D}_\approx$ then $\neg d \in \mathcal{D}_\approx$,
if $d, d' \in \mathcal{D}_\approx$ then $(d \wedge d') \in \mathcal{D}_\approx$,
if $d, d' \in \mathcal{D}_\approx$ then $(d \vee d') \in \mathcal{D}_\approx$,
if $d, d' \in \mathcal{D}_\approx$ then $(d \rightarrow d') \in \mathcal{D}_\approx$.

Before we can define the denotation of a path equation, we have to make the notion precise that two objects in the domain of an interpretation are indiscernible or observationally equal.

Definition 14 Let $I = (U, S, F)$ be an interpretation. An element $b \in U$ is called an *immediate successor* of $a \in U$ iff there exists a feature α such that $F(\alpha)(a) = b$.

Let I be an interpretation. For each element $a \in U$ define recursively its *sort-successors-pair* as the pair containing the sort of a as its left component and as its right component the set containing the sort-successors-pairs of a's immediate successors. $\mathsf{ssp}(a)$ denotes the sort-successors-pair of a.

Note that a sort-successors-pair may be an infinite object. Now, two objects are indiscernible, if they have the same sort-successors-pair. With the notion of a sort-successors-pair we can now define the denotation of a description in $\mathcal{D}_\approx$.

Definition 15 Let $I = (U, S, F)$ be an interpretation. The denotation of a description is defined as follows ($\tau, \tau' \in T_\Sigma, s \in \mathcal{S}, d, d' \in \mathcal{D}_\approx$):
$[\![\tau \sim s]\!] = \{u \in U \mid F(\tau) \text{ is defined on } u \text{ and } S(F(\tau)(u)) = s\}$,
$$[\![\tau \approx \tau']\!] = \left\{ u \in U \,\middle|\, \begin{array}{l} F(\tau) \text{ and } F(\tau') \text{ are defined on } u \\ \text{and } \mathsf{ssp}(F(\tau)(u)) = \mathsf{ssp}(F(\tau')(u)) \end{array} \right\},$$
$[\![\neg d]\!] = U \setminus [\![d]\!]$,
$[\![d \wedge d']\!] = [\![d]\!] \cap [\![d']\!]$,
$[\![d \vee d']\!] = [\![d]\!] \cup [\![d']\!]$,
$[\![d \rightarrow d']\!] = (U \setminus [\![d]\!]) \cup [\![d']\!]$.

Since we have classical negation, it follows that

$$[\![\tau \not\approx \tau']\!] = \left\{ u \in U \;\middle|\; \begin{array}{l} F(\tau) \text{ or } F(\tau') \text{ is not defined on } u \\ \text{or } \mathsf{ssp}(F(\tau)(u)) \neq \mathsf{ssp}(F(\tau')(u)) \end{array} \right\}.$$

The language $\mathcal{D}_{\approx}$ where HPSG path equalities are interpreted as indiscernability of objects has the desirable property of offering unrestricted negation on the one hand and still providing final coalgebras in the category of grammar models on the other.

Theorem 16 *In language $\mathcal{D}_{\approx}$, the category of Γ-models contains a final coalgebra.*

The interpretation of path equations as observational indiscernability may at first seem somewhat unusual to linguists. But it is not only natural if the underlying model is coalgebraic in nature. It is also in line with the idea of a licensing theory. If grammaticality is defined in terms of well-formedness conditions, then it should be observable from the outside whether a candidate structure obeys to these well-formedness conditions. It should not be a potentially unobservable property of its internal representation. Understood in this manour, a path equation expresses that the structures at the end of the paths *behave* exactly alike, that they cannot be distinguished with the expressive means we have. In such a situation it is no longer material whether these two structures are really one and the same or not.

On the other hand, there is little doubt that Pollard and Sag (1994) expect path equations to be interpreted as literal identity. For example, on page 19 they write "It is important to be clear that structure sharing involves token identity of values, not just values that are structurally identical feature structures." So how can we come in terms with the linguists' expectations? Concerning the relation of $=$ and $\approx$ it is simple to see that if two objects are strictly equal, they are indiscernible. And if they are observationally different, they are of course distinct. For a coalgebra in general, the two notions differ. But in a final coalgebra, the notions coincide, because for any observation chain we can find there exists exactly one element in the final coalgebra. Since there are independent reasons to consider the final coalgebra as the model of choice, we can interprete path equations as observational indiscernability, as is desireable for licensing theories, and at the same time meet the linguists' intuitions that path equations denote true identity.

Theorem 17 *In a final coalgebra of the category of Γ-models, if two objects are observationally equal, they are identical. If two objects are distinct, they are observationally different.*

We regard the final coalgebra of a category of Γ-models as the model of choice for an HPSG grammar Γ. As explained at length above, coalgebraic models are the right models for a licensing theory such as HPSG, because licensing theories set out well-formedness conditions of observable properties of linguistic structures and coalgebras model observable properties abstracting from internal structures. The final coalgebra now contains feature structures for the linguistic analyses of all and only the grammatical sentences. That is because for every grammatical utterance there is a coalgebra in the category of Γ-models that models that utterance and its analysis. And by definition, the final coalgebra contains a homomorphic image of the coalgebra. Since the grammaticality properties are preserved by the homomorphism, the final coalgebra models that utterance, too. Natural language utterances are inherently ambiguous. A grammar captures this fact by providing different analyses for readings of the utterance that differ in structure (as opposed to simple lexical ambiguity). Consequently, there will also be different coalgebras in the category of models for the grammar that model the different structural analyses. And for the same reason as above, all these different models find their homomorphic counterpart in the final coalgebra. By definition, a homomorphism can only map a substructure onto a substructure that is observationally equal. And differences in structural analyses lead obviously to observably different feature structures. Thus all structural ambiguities can also be found in the final coalgebra, no reading gets lost.

The final coalgebra of a category of Γ-models will in general not be an exhaustive model. The reason for this is that an exhaustive model will contain some feature structures that are observationally equal, but have different internal structures, while the final coalgebra will contain only a single one of them. As we argued in the introduction, it does not make much sense to distinguish between observationally equal feature structures within the framework of a licensing theory. If the linguistic principles cannot make a distinction between these structures, why should the linguist do so? The final coalgebra contains all and only those feature structure that are justified by the licensing principles of the grammar. Hence we regard it as the model of choice.

5 Proof of the Existence-Theorems

To simplify proofs we will use yet another description language, in which we have true equality $=$ and observational indiscernability $\approx$, but restricted negation. Only atomic sort statements and atomic path indiscernability statements can be negated.

Definition 18 Let Σ be a signature. The set $\mathcal{D}$ of *descriptions* is defined as follows.
If $\tau \in T_\Sigma$ and $s \in \mathcal{S}$ then $\tau \sim s \in \mathcal{D}$,
if $\tau \in T_\Sigma$ and $s \in \mathcal{S}$ then $\tau \not\sim s \in \mathcal{D}$,
if $\tau, \tau' \in T_\Sigma$ then $\tau = \tau' \in \mathcal{D}$,
if $\tau, \tau' \in T_\Sigma$ then $\tau \approx \tau' \in \mathcal{D}$,
if $\tau, \tau' \in T_\Sigma$ then $\tau \not\approx \tau' \in \mathcal{D}$,
if $d, d' \in \mathcal{D}$ then $(d \wedge d') \in \mathcal{D}$,
if $d, d' \in \mathcal{D}$ then $(d \vee d') \in \mathcal{D}$,

The denotation of a description in $\mathcal{D}$ is just the same as it was defined for $\mathcal{D}_=$ and $\mathcal{D}_\approx$. We repeat is here for completeness.

Definition 19 Let $I = (U, S, F)$ be an interpretation. The denotation of a description is defined as follows ($\tau, \tau' \in T_\Sigma, s \in \mathcal{S}, d, d' \in \mathcal{D}$):

$$[\![\tau \sim s]\!] = \{u \in U \mid F(\tau) \text{ is defined on } u \text{ and } S(F(\tau)(u)) = s\},$$
$$[\![\tau \not\sim s]\!] = \{u \in U \mid F(\tau) \text{ is not defined on } u \text{ or } S(F(\tau)(u)) \neq s\},$$
$$[\![\tau = \tau']\!] = \left\{ u \in U \;\middle|\; \begin{array}{l} F(\tau) \text{ and } F(\tau') \text{ are defined on } u \\ \text{and } F(\tau)(u) = F(\tau')(u) \end{array} \right\},$$
$$[\![\tau \approx \tau']\!] = \left\{ u \in U \;\middle|\; \begin{array}{l} F(\tau) \text{ and } F(\tau') \text{ are defined on } u \\ \text{and } \mathsf{ssp}(F(\tau)(u)) = \mathsf{ssp}(F(\tau')(u)) \end{array} \right\},$$
$$[\![\tau \not\approx \tau']\!] = \left\{ u \in U \;\middle|\; \begin{array}{l} F(\tau) \text{ or } F(\tau') \text{ is not defined on } u \\ \text{or } \mathsf{ssp}(F(\tau)(u)) \neq \mathsf{ssp}(F(\tau')(u)) \end{array} \right\},$$
$$[\![d \wedge d']\!] = [\![d]\!] \cap [\![d']\!],$$
$$[\![d \vee d']\!] = [\![d]\!] \cup [\![d']\!].$$

We fix the signature Σ and the grammar Γ as parameters and consider the category $\mathcal{K}$ of all Γ-models with coalgebra-homomorphisms as morphisms. We will show that for the conjoint description language $\mathcal{D}$ the category $\mathcal{K}$ has final coalgebras.

In subsequent proofs, we will simplify notation of feature paths a little bit. If $I = (U, S, F)$ is an interpretation and $u \in U$ then instead of writing $F(\tau)(u)$ we write τu, and instead of $F(\alpha)(F(\tau)(u))$ we write $\tau\alpha u$.

We start by showing some important properties of homomorphisms. Homomorphisms preserve modelhood and they are confluent. But first three useful technical lemma.

Lemma 20 *Let $\mathfrak{A}$ and $\mathfrak{B}$ be Σ-structures and $h : \mathfrak{A} \to \mathfrak{B}$ a homomorphism. Let $\tau \in T_\Sigma$ be a feature path.*

1. *If $u \in U_{\mathfrak{A}}$ is such that τu is defined then $h(\tau u) = \tau h(u) \in \mathfrak{B}$.*
2. *If $u \in U_{\mathfrak{A}}$ is such that τu is defined then $S_A(\tau u) = S_b(h(\tau u))$.*
3. *If $u \in U_{\mathfrak{A}}$ is such that τ is defined on $h(u) \in U_{\mathfrak{B}}$ then τ is defined on u.*

Proof. (1) can be shown by a simple induction on the length of the path.
(2) Again a simple induction on the length of the path: If $\tau = :$ then $S_A(u) = S_B(h(u))$, because h is a homomorphism.
Let $\tau = \tau'\alpha$. By induction hypothesis, $S_A(\tau' u) = S_B(h(\tau' u))$. Thus feature α is defined on $\tau' u$ and $h(\tau' u)$, because $\mathfrak{A}$ and $\mathfrak{B}$ are totally well-typed. Then $S_A(\tau'\alpha u) = S_B(h(\tau'\alpha u))$ because h is a homomorphism.
(3) Let $\tau \in T_\Sigma$ be a path and $u \in U_{\mathfrak{A}}$ such that τ is defined on $h(u)$. Then τ is defined on u.
In the induction step, let $\tau = \tau'\alpha$. By the induction hypothesis, τ' is defined on u, and by (2), $S_A(\tau' u) = S_B(h(\tau' u)) = s$ for some sort s. Since τ is defined on $h(u)$, we know α is appropriate for s. Thus α is defined on $\tau' u$, because $\mathfrak{A}$ is totally well-typed. Hence τ is defined on u. ∎

Lemma 21 *Let $\mathfrak{A}$ and $\mathfrak{B}$ be Σ-structures and $h : \mathfrak{A} \to \mathfrak{B}$ be a homomorphism and $u \in U_{\mathfrak{A}}$. Then $\mathsf{ssp}(u) = \mathsf{ssp}(h(u))$.*

Proof. This is an immediate consequence of the above lemma. Let τ be any feature path. By Lemma 20 (1) and (3), τ is defined on u iff it is defined on $h(u)$ and by (2) $S_A(\tau u) = S_B(h(\tau u))$. Hence $\mathsf{ssp}(u) = \mathsf{ssp}(h(u))$. ∎

Corollary 22 *let $\mathfrak{A}$ and $\mathfrak{B}$ be Σ-structures and $h : \mathfrak{A} \to \mathfrak{B}$ be a homomorphism. For all $u, v \in U_{\mathfrak{A}}$, if $h(u) = h(v)$ then the sort-successors-pairs of u and v are identical.*

Proof. $h(u) = h(v) \Longrightarrow \mathsf{ssp}(h(u)) = \mathsf{ssp}(h(v))$ and $\mathsf{ssp}(u) = \mathsf{ssp}(h(u))$ and $\mathsf{ssp}(v) = \mathsf{ssp}(h(v))$ by the above lemma. Thus $\mathsf{ssp}(u) = \mathsf{ssp}(v)$. ∎

Lemma 23 *Let $\mathfrak{A}$ and $\mathfrak{B}$ be Σ-structures and $\mathfrak{A}$ in particular a Γ-model. Let $h : \mathfrak{A} \to \mathfrak{B}$ be a surjective homomorphism. Then $\mathfrak{B}$ is a Γ-model.*

Proof. Let $u \in U_{\mathfrak{A}}$. We show that for every description d, if $u \in [\![d]\!]^{\mathfrak{A}}$ then $h(u) \in [\![d]\!]^{\mathfrak{B}}$ by a structural induction over descriptions starting with the five different types of literals.

Let $u \in [\![\tau \sim s]\!]^{\mathfrak{A}}$. Then τ is defined on $h(u)$ and $s = S_A(\tau u) = S_B(h(\tau u))$ by (1) and (2) of the above lemma. Hence $h(u) \in [\![\tau \sim s]\!]^{\mathfrak{B}}$.

Let $u \in [\![\tau \not\sim s]\!]^{\mathfrak{A}}$. We distinguish two cases.
Firstly, let τ be defined on u. Then there is a sort $t \in \mathcal{S}$ with $t \neq s$ and $S_A(\tau u) = t$. Thus τ is defined on $h(u)$ and $t = S_A(\tau u) = S_B(h(\tau u))$ by Lemma 20 (1) and (2).
Hence $h(u) \in [\![\tau \not\sim s]\!]^{\mathfrak{B}}$.
Secondly, let τ not be defined on u. Then τ is not defined on $h(u)$ by Lemma 20 (3).
So $h(u) \in [\![\tau \not\sim s]\!]^{\mathfrak{B}}$.

Let $u \in [\![\tau = \tau']\!]^{\mathfrak{A}}$. Then $\tau u = \tau' u$. Then τ and τ' are defined on $h(u)$ and $\tau h(u) = h(\tau u) = h(\tau' u) = \tau' h(u)$ by Lemma 20 (1). Hence $h(u) \in [\![\tau = \tau']\!]^{\mathfrak{B}}$.

Let $u \in [\![\tau \approx \tau']\!]^{\mathfrak{A}}$. Then $\mathsf{ssp}(\tau u) = \mathsf{ssp}(\tau' u)$. Then τ and τ' are defined on $h(u)$ and $\tau h(u) = h(\tau u)$ and $h(\tau' u) = \tau' h(u)$ by Lemma 20 (1). By Lemma 21, $\mathsf{ssp}(\tau h(u)) = \mathsf{ssp}(\tau' h(u))$. Hence $h(u) \in [\![\tau \approx \tau']\!]^{\mathfrak{B}}$.

Let $u \in [\![\tau \not\approx \tau']\!]^{\mathfrak{A}}$. We distinguish two cases.
Firstly, let τ be undefined on u. Then τ is undefined on $h(u)$ by Lemma 20 (3). Hence $h(u) \in [\![\tau \not\approx \tau']\!]^{\mathfrak{B}}$.

Secondly, let τ and τ' be defined on u and $\mathsf{ssp}(\tau u) \neq \mathsf{ssp}(\tau' u)$. Then τ and τ' are defined on $h(u)$ and $\tau h(u) = h(\tau u)$ and $h(\tau' u) = \tau' h(u)$ by Lemma 20 (1). And $\mathsf{ssp}(\tau h(u)) = \mathsf{ssp}(\tau u) \neq \mathsf{ssp}(\tau' u) = \mathsf{ssp}(\tau' h(u))$ by Lemma 21.
Hence $h(u) \in [\![\tau \not\approx \tau']\!]^{\mathfrak{B}}$.

Now for the induction step.
Let d and $d' \in \mathcal{D}$ be descriptions $u \in [\![d \wedge d']\!]^{\mathfrak{A}}$. Then $u \in [\![d]\!]^{\mathfrak{A}}$ and $u \in [\![d']\!]^{\mathfrak{A}}$. By induction hypothesis $h(u) \in [\![d]\!]^{\mathfrak{B}}$ and $h(u) \in [\![d']\!]^{\mathfrak{B}}$. Therefore $h(u) \in [\![d \wedge d']\!]^{\mathfrak{B}}$.

Let d and $d' \in \mathcal{D}$ be descriptions $u \in [\![d \vee d']\!]^{\mathfrak{A}}$. Then $u \in [\![d]\!]^{\mathfrak{A}}$ or $u \in [\![d']\!]^{\mathfrak{A}}$. By induction hypothesis $h(u) \in [\![d]\!]^{\mathfrak{B}}$ or $h(u) \in [\![d']\!]^{\mathfrak{B}}$. Therefore $h(u) \in [\![d \vee d']\!]^{\mathfrak{B}}$. ■

Lemma 24 (Local confluence lemma) *Let $\mathfrak{A}, \mathfrak{B}$, and $\mathfrak{C}$ be Γ-models and $h_{AB} : \mathfrak{A} \to \mathfrak{B}$ and $h_{AC} : \mathfrak{A} \to \mathfrak{C}$ be surjective homomorphisms.*
Then there exists a Γ-model $\mathfrak{D}$ and surjective homomorphisms $h_{BD} : \mathfrak{B} \to \mathfrak{D}$ and $h_{CD} : \mathfrak{C} \to \mathfrak{D}$ such that the following diagram commutes:

$$\begin{tikzcd}
& \mathfrak{A} \arrow[dl, "h_{AB}"'] \arrow[dr, "h_{AC}"] & \\
\mathfrak{B} \arrow[dr, "h_{BD}"'] & & \mathfrak{C} \arrow[dl, "h_{CD}"] \\
& \mathfrak{D} &
\end{tikzcd}$$

Proof. Define equivalence classes on $U_{\mathfrak{A}}$ by setting for each $u \in U_{\mathfrak{A}}$ $[u] = \{v \in U_{\mathfrak{A}} \mid h_{AB}(u) = h_{AB}(v)$ or $h_{AC}(u) = h_{AC}(v)\}$.
Define $\mathfrak{D} = (D, S_D, F_D)$ by
$D = \{[u] \mid u \in U_{\mathfrak{A}}\}$
$S_D([u]) = S_{\mathfrak{A}}(u)$
$F_D : \alpha[u] = [v]$ iff $\alpha u = v$ in $\mathfrak{A}$.
Fact 1: $\mathfrak{D}$ is a proper Σ-structure.
Since homomorphisms are sort preserving, if $h(u) = h(v)$ then $S_A(u) = S_A(v)$. Hence, S_D is well-defined.
Let $\alpha u = v$ and $u' \in [u]$. Since $S_A(u) = S_A(v)$ and $\mathfrak{A}$ is totally well-typed, α is defined on u'. Let $h_{AB}(u) = h_{AB}(u')$ (– the argument for $h_{AC}(u) = h_{AC}(u')$ is analogue).
$\Longrightarrow \alpha h_{AB}(u) = \alpha h_{AB}(u')$
$\Longrightarrow h_{AB}(\alpha u) = h_{AB}(\alpha u')$
$\Longrightarrow \alpha u' \in [v]$.
Thus F_D is well-defined.

Let $S_D([u]) = s, S_D([v]) = t$, and $\alpha[u] = [v]$.
Then $S_A(u) = s, S_A(v) = t$, and $\alpha u = v$ by definition of $\mathfrak{D}$.
Then $\langle s, \alpha, t\rangle \in \mathcal{F}$ of Σ, because $\mathfrak{A}$ is totally well-typed.

Let $\langle s, \alpha, t\rangle \in \mathcal{F}$ of Σ and $S_D([u]) = s$.
Then $S_A(u) = s$. α is defined on u and $S_A(\alpha u) = t$, because $\mathfrak{A}$ is totally well typed.
Then α is defined on $[u]$ and $S_D([\alpha u]) = t$ by definition of $\mathfrak{D}$.

Thus $\mathfrak{D}$ is a proper Σ-structure.

Fact 2: There exists a surjective homomorphism $h_{AD} : \mathfrak{A} \to \mathfrak{D}$.
Define $h_{AD} : u \mapsto [u]$ for all $u \in U_{\mathfrak{A}}$.
Let $u \in U_{\mathfrak{A}}$. $S_D(h_{AD}(u)) = S_D([u]) = S_A(u)$ by definition of $\mathfrak{D}$.
$\alpha h_{AD}(u) = \alpha[u] = [\alpha u]$ by definition of F_D
$= h_{AD}(u)$.

So, h_{AD} is a homomorphism.
Let $d \in D$. Then there is a $u \in U_{\mathfrak{A}}$ such that $d = [u]$. By definition of h_{AB} we have $h_{AB}(u) = d$. Thus h_{AB} is surjective.

Fact 3: $\mathfrak{D}$ is a Γ-model.
Follows immediately from Fact 2 and the fact that homomorphisms preserve modelhood.

Fact 4: There exists a surjective homomorphism $h_{BD} : \mathfrak{B} \to \mathfrak{D}$.
Define $h_{BD} : h_{AB}(u) \mapsto [u]$ for all $u \in U_{\mathfrak{A}}$.
Since h_{AB} is surjective, h_{BD} is a total on $U_{\mathfrak{B}}$. It is a function, because if $h_{AB}(u) = h_{AB}(v)$ then $v \in [u]$ by definition.
Let $b \in U_{\mathfrak{B}}$. There exists a $u \in U_{\mathfrak{A}}$ such that $b = h_{AB}(u)$.

$$
\begin{aligned}
S_B(b) = S_B(h_{AB}(u)) &= S_A(u) \qquad \text{because } h_{AB} \text{ is a homomorphism} \\
&= S_D([u]) \qquad \text{by definition of } S_D \\
&= S_D(h_{BD}(b)).
\end{aligned}
$$

Let $c \in U_{\mathfrak{B}}$ and $\alpha b = c$. There exists a $v \in U_{\mathfrak{A}}$ with $c = h_{AB}(v)$ and $\alpha u = v$.
Now $\alpha h_{BD}(b) = \alpha[u] = [\alpha u]$ by definition of F_D.
And $h_{BD}(\alpha b) = h_{BD}(c) = [v] = [\alpha u]$.
Thus h_{BD} is a homomorphism.
Let $d \in D$. Then there is a $u \in U_A$ with $d = [u]$. By definition of $h_{BD} : h_{AB}(u) \mapsto d$. Thus h_{BD} is surjective.

Fact 5: There exists a surjective homomorphism $h_{CD} : \mathfrak{C} \to \mathfrak{D}$.
The argument is analogue to the proof for Fact 4. ∎

Lemma 25 *There exists a Γ-model $\mathfrak{A}$ in $\mathcal{K}$ such that for each Γ-model $\mathfrak{B}$ in $\mathcal{K}$ and for each $u \in U_{\mathfrak{B}}$ it is true that $\mathfrak{A}$ contains a homomorphic image of $\langle u \rangle$.*

This is a consequence of the theorem by King that every grammar has an exhaustive model (King (1999)). The following line of argument leads to Proposition 34, a strengthening of the above lemma stating that there is a *unique* homomorphic image of $\langle u \rangle$.

Definition 26 A Γ-model A is called *minimal* iff for every Γ-model $\mathfrak{B}$ and homomorphism $h : \mathfrak{A} \to \mathfrak{B}$ it is true that h is injective.

Let $\mathfrak{A}$ and $\mathfrak{B}$ be Γ-models. $\mathfrak{B}$ is called a minimal model for $\mathfrak{A}$ iff there exists a surjective homomorphism $h : \mathfrak{A} \to \mathfrak{B}$.

Lemma 27 *The minimal models of a Γ-model are unique up to isomorphism.*

Proof. Let $\mathfrak{A}$ be a Γ-model, $\mathfrak{B}_1$ and $\mathfrak{B}_2$ two minimal models of $\mathfrak{A}$, and $h_{1,2} : \mathfrak{A} \to \mathfrak{B}_{1,2}$ surjective homomorphisms.
By the local confluence lemma, there exists a Γ-model $\mathfrak{D}$ and surjective homomorphisms $g_{1,2} : \mathfrak{B}_{1,2} \to \mathfrak{D}$.
$g_{1,2}$ is injective, because $\mathfrak{B}_{1,2}$ is minimal.
As a homomorphism, $g_{1,2}$ is sort-preserving in both directions, hence $g_{1,2}$ is an isomorphism.
So, $\mathfrak{B}_{1,2}$ is isomorphic to $\mathfrak{D}$ and thus $\mathfrak{B}_1$ is isomorphic to $\mathfrak{B}_2$. ∎

The next task is to show the existence of minimal models.

Definition 28 Let $\mathfrak{A}$ be a Γ-model. Define a structure $\mathfrak{A}{\downarrow} = (U, S, F)$ by

U is the set of all sort-successors-pairs of $\mathfrak{A}$,
S associates with each sort-successors-pair its left component,
$A : (u_1, u_2) \in \alpha_{\mathfrak{A}\downarrow}$ iff $(v_1, v_2) \in \alpha_{\mathfrak{A}}$,
u_1 is the sort-successors pair of v_1, and
u_2 is the sort-successors pair of v_2.

Lemma 29 $\mathfrak{A}\downarrow$ *is a proper* Σ*-interpretation.*

Proof. What we have to show is that F obeys to the appropriateness function $\mathcal{A}$ of the signature Σ. Let $(s, \alpha, t) \in \mathcal{A}$ and $u \in U_{\mathfrak{A}\downarrow}$ with $S(u) = s$. Then there is a $v \in U_{\mathfrak{A}}$ with sort s and sort-successor-pair u. Since $\mathfrak{A}$ is totally well-typed, there is a $v' \in U_{\mathfrak{A}}$ such that v' is of sort t and $\alpha v = v'$. By construction of $\mathfrak{A}\downarrow$, there is a $u' \in U_{\mathfrak{A}\downarrow}$ that is the sort-successors-pair of v' and hence has sort t. By definition of F, we have $\alpha u = u'$.

Let $u, u' \in U_{\mathfrak{A}\downarrow}, S(u) = s, S(u') = t$, and $\alpha u = u'$. By definition of $\mathfrak{A}\downarrow$ there is a $v \in U_{\mathfrak{A}}$ with $S_{\mathfrak{A}}(v) = s$ and sort-successors-pair u, a $v' \in U_{\mathfrak{A}}$ with $S_{\mathfrak{A}}(v') = t$ and sort-successors-pair u', and $\alpha v = v'$. Since $\mathfrak{A}$ obeys to $\mathcal{F}$, we know $(s, \alpha, t) \in \mathcal{F}$. ■

Lemma 30 *There exists a surjective homomorphism* $h : \mathfrak{A} \to \mathfrak{A}\downarrow$.

Proof. Define h as follows: Map each $v \in U_{\mathfrak{A}}$ to its sort-successors-pair. Then h is obviously a surjective function.
Let $v \in U_{\mathfrak{A}}$ and $S_{\mathfrak{A}}(v) = s$. Let u be the sort-successors-pair of v. Then S assigns s to u by definition of S.
Let $v \in U_{\mathfrak{A}}$ and α a feature appropriate for v and $\alpha v = v'$. Let u be the sort-successors-pair of v (i.e., $hv) = u$) and u' be the sort-successors-pair of v' (i.e., $h(v') = u'$). Thus $h(\alpha v) = h(v') = u'$. By definition of F: $(u, u') \in \alpha_{\mathfrak{A}\downarrow}$. Therefore $\alpha(h(v)) = \alpha u = u'$. ■

Corollary 31 $\mathfrak{A}\downarrow$ *is a* Γ*-model.*

Proof. This follows immediately from the above lemma and Lemma 23. ■

Lemma 32 *Let* $\mathfrak{A}$ *be a* Γ*-model. Every* $u \in U_{\mathfrak{A}\downarrow}$ *is identical with its own sort-successors-pair.*

Proof. Let $u \in U_{\mathfrak{A}\downarrow}$. There is a $v \in U_{\mathfrak{A}}$ such that u is the sort-successors-pair of v. Simply by definition of $\mathfrak{A}\downarrow$ for every feature path $\tau \in T_{\Sigma}$ we know τ is defined on v if and only if it is defined on u, and if it is defined then $S_{\mathfrak{A}}(\tau v) = S_{\mathfrak{A}\downarrow}(\tau u)$. Hence by definition of a sort-successors-pair, u and v have the same sort-successors-pair. Since u is the sort-successors-pair of v it follows that u is identical with its own sort-successors-pair. ■

Lemma 33 $\mathfrak{A}\downarrow$ *is a minimal model of* $\mathfrak{A}$.

Proof. Let $\mathfrak{B}$ be a Γ model and $h : \mathfrak{A}\downarrow \to \mathfrak{B}$ be a homomorphism. Let $u, v \in U$ such that $h(u) = h(v)$. By Lemma 22, the sort-successors-pair of u is identical with that of v. By the above lemma, u is identical with its own sort-successors-pair, and v is identical with its own sort-successors-pair. Therefore $u = v$. This shows that $\mathfrak{A}\downarrow$ is a minimal model. That it is a minimal model *for* $\mathfrak{A}$ follows from Lemma 30. ■

Proposition 34 *There exists a* Γ*-model* $\mathfrak{A}$ *in* $\mathcal{K}$ *such that for each* Γ*-model* $\mathfrak{B}$ *in* $\mathcal{K}$ *and for each* $u \in U_{\mathfrak{B}}$ *it is true that* $\mathfrak{A}$ *contains a* unique *homomorphic image of* $\langle u \rangle$.

Proof. By Lemma 25, there exists a Γ-model $\mathfrak{B}$ in $\mathcal{K}$ such that for each Γ-model $\mathfrak{C}$ in $\mathcal{K}$ and each $u \in U_{\mathfrak{C}}$: $\mathfrak{B}$ contains a homomorphic image of $\langle u \rangle$. We will show that $\mathfrak{B}{\downarrow}$, the minimal model of $\mathfrak{B}$ is the desired Γ-model.
Let $\mathfrak{C}$ be a Γ-model in $\mathcal{K}$ and $u \in U_{\mathfrak{C}}$. By Lemma 30, there exists a homomorphism $h : \mathfrak{B} \to \mathfrak{B}{\downarrow}$. Hence $\mathfrak{B}{\downarrow}$ contains a homomorphic image of $\langle u \rangle$.
For uniqueness, let $h_1 : \langle u \rangle \to \mathfrak{B}{\downarrow}$ and $h_2 : \langle u \rangle \to \mathfrak{B}{\downarrow}$ be two homomorphisms. On their images $h_1(\langle u \rangle)$ and $h_2(\langle u \rangle)$, h_1 and h_2 are of course surjective. By the local confluence lemma, there exists a Γ-model $\mathfrak{D}$ and surjective homomorphisms $g_1 : h_1(\langle u \rangle) \to \mathfrak{D}$ and $g_2 : h_2(\langle u \rangle) \to \mathfrak{D}$. g_1 and g_2 are injective. (If g_1 was not, it could be extended to a non-injective homomomorphism on $\mathfrak{B}{\downarrow}$, which contradicts the minimality of $\mathfrak{B}{\downarrow}$. Hence g_1 and g_2 are isomorphisms, and thus $h_1(\langle u \rangle)$ and $h_2(\langle u \rangle)$ are isomorphic. By definition of a sort-successors-pair, $h_1(u)$ and $h_2(u)$ have the same sort-successors-pair. By Lemma 32, $h_1(u)$ and $h_2(u)$ are identical with its sort-successors-pair. Hence $h_1(u) = h_2(u)$, and their is one unique homomorphic image of $\langle u \rangle$. ∎

Definition 35 A Γ-model $\mathfrak{A}$ as in Proposition 34 is called a *fit* model.

Proposition 36 *Let $\mathfrak{A}$ be a fit model in the category $\mathcal{K}$. Then $\mathfrak{A}$ is a final coalgebra in $\mathcal{K}$.*

Proof. Let $\mathfrak{A}$ be a fit Γ-model and $\mathfrak{B}$ be a Γ-model in $\mathcal{K}$. We first show the existence of a homomorphism from $\mathfrak{B}$ to $\mathfrak{A}$. For each $b \in U_{\mathfrak{B}}$ there exists a unique homomorphism $h_b : \langle b \rangle \to \mathfrak{A}$. Define $h = \bigcup_{b \in U_{\mathfrak{B}}} h_b$.

Fact 1: h is a function.
Let $b, c \in U_{\mathfrak{B}}$ such that $c \in \langle b \rangle$. Then $\langle c \rangle$ is a substructure of $\langle b \rangle$ by definition of substructures. Since $\mathfrak{A}$ contains exactly one homomorphic image of $\langle c \rangle$ it follows that $h_b(\langle c \rangle) = h_c(\langle c \rangle)$ and a fortiori $h_b(c) = h_c(c)$.

Fact 2: h is a homomorphism.
For each $b \in U_{\mathfrak{B}}$ we know that h and h_b agree on $\langle b \rangle$. h_b is a homomorphism, so h is a homomorphism on $\langle b \rangle$, too.

Now we show uniqueness of the homomorphism from $\mathfrak{B}$ to $\mathfrak{A}$. Let $h : \mathfrak{B} \to \mathfrak{A}$ and $g : \mathfrak{B} \to \mathfrak{A}$ be two homomorphisms and $b \in U_{\mathfrak{B}}$. Since $\mathfrak{A}$ contains a *unique* homomorphic image of $\langle b \rangle$ it follows that h and g agree on $\langle b \rangle$. So h and g agree on $U_{\mathfrak{B}}$. ∎

Theorem 12 stating that category $\mathcal{K}$ in a language with true equality but without disequality has final coalgebras is now a simple consequence of Proposition 36. Also Theorem 16 stating that $\mathcal{K}$ in the language $\mathfrak{D}_{\approx}$ has final coalgebras is a consequence of this proposition. Theorem 17 stating that in the final coalgebra the notions of path equality and indiscernability coincide follows from Proposition 36 and Lemma 32.

6 Conclusion

We provided a coalgebraic modelling of Head-Driven Phrase Structure Grammar. We showed that this modelling is conceptually and technically very well suited for HPSG. Conceptually, because HPSG is a licensing theory, and licensing theories define grammaticality in terms of well-formedness conditions of observabilities of linguistic structures. Coalgebras are *the* tools for modelling observable properties of systems. Technically, because the final coalgebra of the category of all coalgebraic models of a grammar contains a feature structural analysis of each structurally different reading of every grammatical sentence while eliminating spurious ambiguities that other models tend to have. And also because in the final coalgebra the at first maybe somewhat unusual denotation of path equations coincides with the intuitions that linguists have about path equations. Coalgebraic models provide the first conceptually adequate formalisation of licensing theories like HPSG while fulfilling the technical demands HPSG grammarians have.

Acknowledgements

I would like to thank Uwe Mönnich, Detmar Meurers, Frank Richter, Tilman Höhle, Kai-Uwe Kühnberger, and Jens Michaelis for helpful discussions.
This research is funded by the German Research Foundation (DFG) as part of the Special Research Programme (Sonderforschungsbereich) 441.

References

Carpenter, B. (1992). *The Logic of Typed Feature Structures.* Cambridge University Press.

Cohn, P. M. (1965). *Universal Algebra.* Harper & Row, New York.

Grätzer, G. (1979). *Universal Algebra.* Springer-Verlag, Berlin, second edition.

Jacobs, B. and Rutten, J. (1997). A tutorial on (co)algebras and (co)induction. *Bulletin of EATCS*, 62:222–259.

Kepser, S. (1994). A satisfiability algorithm for a typed feature logic. Master's thesis, Seminar für Sprachwissenschaft, University of Tübingen, Arbeitspapiere des SFB 340, Bericht Nr. 60.

King, P. J. (1989). *A Logical Formalism for Head-Driven Phrase Struture Grammar.* PhD thesis, University of Manchester.

King, P. J. (1999). Towards truth in HPSG. In Kordoni, V., editor, *Tübingen Studies in Head-Driven Phrase Structure Grammar*, volume 2, pages 301–352. Arbeitspapiere des SFB 340, Bericht Nr. 132, Tübingen, Germany.

King, P. J., Simov, K. I., and Aldag, B. (1999). The complexity of modellability in finite and computable signatures of a constraint logic for head-driven phrase structure grammar. *Journal of Logic, Language and Information*, 8(1):83–110.

Mal'cev, A. I. (1971). *The Metamathematics of Algebraic Systems*, volume 66 of *Studies in Logic.* North-Holland Publishing Company.

Mal'cev, A. I. (1973). *Algebraic Systems*, volume 192 of *Die Grundlehren der mathematischen Wissenschaften in Einzeldarstellung.* Springer-Verlag, Berlin.

Pierce, B. C. (1991). *Basic Category Theory for Computer Scientists.* MIT Press, Cambridge, USA.

Pollard, C. (1999). Strong generative capacity in HPSG. In Webelhuth, G., Koenig, J.-P., and Kathol, A., editors, *Lexical and Constructional Aspects of Linguistic Explanation*, pages 281–297. CSLI.

Pollard, C. and Sag, I. A. (1987). *Information Based Syntax and Semantics, Vol. 1: Fundamentals.* Number 13 in Lecture Notes. CSLI.

Pollard, C. and Sag, I. A. (1994). *Head-Driven Phrase Structure Grammar.* University of Chicago Press.

Rutten, J. (1996). Universal coalgebra: A theory of systems. Technical Report CS-R9652, CWI, Amsterdam.