Arbeitspapiere des SFB 340
Bericht Nr. 136, Juli 1999

COMPILING FEATURE-BASED CONSTRAINTS WITH COMPLEX ANTECEDENTS

GERALD PENN AND KAREN STEINICKE

ABSTRACT. This paper defines a description language for implicational constraints with complex antecedents such as could be used to formalise a large class of HPSG (Pollard and Sag 1987; Pollard and Sag 1994) principles, and describes how to transform constraints from that language to ones that are compatible with ALE signatures (Carpenter and Penn 1996).

CONTENTS

1. INTRODUCTION

In the ALE programming language (Carpenter and Penn 1996), universal implicational constraints are provided for specifying structurally more complicated constraints on feature structures in a grammar signature than appropriateness conditions, but they are still restricted to have only types in their antecedents. While that may be suitable for a significant number of linguistic principles from Head-driven Phrase Structure Grammar (Pollard and Sag 1987; Pollard and Sag 1994), there are also clearly cases where having the ability to specify complex antecedents (drawn from a feature description language) is desirable.

This paper has three purposes. The first is to define and provide a denotational semantics for a description language that can be used to specify principles in HPSG as constraints with complex antecedents. This language bears the closest syntactic resemblance to the description language

ISSN 0947-6954/99

defined in Carpenter 1992 for the logic of typed feature structures (LTFS), but it will assume, following King 1989 and Pollard and Sag 1994 that all objects in its interpretations belong to the set denoted by exactly one maximally-specific consistent type (sometimes called species). In this respect, it can be viewed as an extension of SRL (King 1989) to include variables and general inequations. The interpretations admitted by the signature of such a language are a proper subset of those admitted by ALE/LTFS signatures, and it will be assumed here that the implicational constraints necessary for those interpretations (specifically, constraints that, for each non-maximally-specific type, constrain a feature structure of that type to be of one of its maximally specific consistent extensions) would be included in an ALE/LTFS signature in an implementation of this work. Since these are already type-antecedent constraints, they will not be discussed further here.

The second purpose is to describe how to translate from general implicational constraints in which antecedents can be arbitrary inequation-free expressions from this description language to type-antecedent constraints, thus making them suitable for inclusion into an ALE/LTFS signature.

The third purpose is to point out a rather strange effect that occurs when general negation (effectively added by implication) and inequations of general descriptions are included in the same language such that negation can take scope over inequations. This does not appear in SRL because there are no general inequations (only path equations that can stand in the scope of general negation), it does not appear in the LTFS description language because there is no general negation operator and no general inequations (path inequations are explicitly added), and it does not appear in ALE because negation only exists implicitly in universal implicational constraints, and the antecedents to those constraints are restricted to be types. A related effect emerges if one combines general inequations, variables and existential quantification such that existential quantification can take scope over inequations. Both of these will be discussed in the last section.

2. The Signature and Description Language

The signature of this language and its admissible interpretations are taken to be the same as an SRL signature with a finite number of species:

Definition 1. $\Sigma = \langle \mathcal{S}, \mathcal{F}, \mathcal{A} \rangle$ *is a* (King) signature *iff:*

- $\mathcal{S}$ *is a finite set of* species,
- $\mathcal{F}$ *is a set of* features, *and*

- $\mathcal{A} : \mathcal{S} \times \mathcal{F} \rightarrow P(\mathcal{S})$ *is a total* appropriateness function.

DEFINITION 2. *Given a signature,* $\Sigma = \langle \mathcal{S}, \mathcal{F}, \mathcal{A} \rangle$, $I = \langle U, S, F \rangle$ *is an* interpretation of Σ *iff:*

- U *is a set, the* universe of objects,
- $S : U \rightarrow \mathcal{S}$ *is a total* species interpretation *function,*
- $F : \mathcal{F} \rightarrow (U \rightharpoonup U)$ *is a total* feature interpretation *function from* $\mathcal{F}$ *to the set of partial functions on* U*, and*
- *for each* $f \in \mathcal{F}$, $u \in U$:
  - $F(f)(u)\downarrow$ *iff* $\mathcal{A}(S(u), f) \neq \emptyset$*, and*
  - *if* $F(f)(u)\downarrow$*, then* $S(F(f)(u)) \in \mathcal{A}(S(u), f)$.

We also need a set of types to use in the description language, for which the set of species of the signature forms the set of maximally specific consistent types:

DEFINITION 3. *Given a signature,* $\Sigma = \langle \mathcal{S}, \mathcal{F}, \mathcal{A} \rangle$*, a* type basis for Σ *is a top-smashed finite bounded complete partial order,* $\langle T_\Sigma, \sqsubseteq \rangle$ *for which:*

- $\mathcal{S}$ *is the set of maximally-specific consistent types of* T_Σ*, i.e.,* $S \subseteq T_\Sigma$ *and for all* $s \in S, t \in T_\Sigma$*, if* $s \sqsubseteq t$*, then either* $t = s$ *or* $t = \top$*, and*
- *for all* $f \in \mathcal{F}$*, there is a unique least* $t \in \mathcal{T}$ *such that for all* $s \in S$ *for which* $t \sqsubseteq s$, $\mathcal{A}(s, f) \neq \emptyset$*. This least type is called Intro*(f)*, the* introducer of f.

This second condition is the assumption of unique feature introduction, taken from Carpenter 1992. Because type bases are actually independent constructs from King signatures and their interpretations, this is consistent with SRL. In this paper, however, the type, $\top$, will refer to inconsistency, and the type, $\bot$, will refer to truth.

DEFINITION 4. *Given a signature,* $\Sigma = \langle \mathcal{S}, \mathcal{F}, \mathcal{A} \rangle$*, a type basis for* Σ, T_Σ*, and a countable set of variables,* V*, the* consequent language for Σ and T_Σ, $D_{\Sigma,T}$*, is the smallest set containing:*

- x*, for all* $x \in V$,
- t*, for all* $t \in T_\Sigma$,
- $f{:}\phi$*, for all* $f \in \mathcal{F}$, $\phi \in D_{\Sigma,T}$,
- $\not\asymp\phi$*, for all* $\phi \in D_{\Sigma,T}$*, and*
- $\phi_1 \wedge \phi_2$ *and* $\phi_1 \vee \phi_2$*, for all* $\phi_1, \phi_2 \in D_{\Sigma,T}$.

DEFINITION 5. *Given a signature,* $\Sigma = \langle \mathcal{S}, \mathcal{F}, \mathcal{A} \rangle$*, a type basis for* Σ, T_Σ*, and a countable set of variables,* V*, the* antecedent language for Σ and T_Σ, $N_{\Sigma,T}$*, is the smallest set containing:*

- x*, for all* $x \in V$,

- *t, for all $t \in T_\Sigma$,*
- *$f{:}\phi$, for all $f \in \mathcal{F}$, $\phi \in N_{\Sigma,T}$,*
- *$\exists x.\phi$, for all $x \in V$, $\phi \in N_{\Sigma,T}$,*
- *$\phi_1 \wedge \phi_2$ and $\phi_1 \vee \phi_2$, for all $\phi_1, \phi_2 \in N_{\Sigma,T}$.*

The motivation for using variables rather than variable-free path equations / inequations is that variables in a feature logic with disjunction can make descriptions exponentially more compact (Kasper 1987). Narrow existential quantification is then required in the antecedent language, as an implicational constraint such as:

$$(f{:}x \wedge g{:}x) \rightarrow \psi$$

is logically equivalent to:

$$\neg \mathit{Intro}(f) \vee \neg \mathit{Intro}(g) \vee f{:}\not\approx x \vee g{:}\not\approx x \vee \psi$$

rather than:

$$\neg \mathit{Intro}(f) \vee \neg \mathit{Intro}(g) \vee (f{:}x \wedge g{:}\not\approx x) \vee \psi$$

as at least one intuitive reading of the original implication would suggest. This latter equivalence is achieved using existential quantification:

$$\exists x.[f{:}x \wedge g{:}x] \rightarrow \psi$$

In theory, the antecedent language could include variable-free path inequations as well. In practice, an implementation of this extension, while possible, would be far less elegant as inequations are typically represented as failure blocked on an equality constraint.

It will be assumed here that all constraints are implicational in form:

Definition 6. *Given a signature, $\Sigma = \langle \mathcal{S}, \mathcal{F}, \mathcal{A} \rangle$, a type basis for Σ, T_Σ, and a countable set of variables, V, the* implicational language for Σ and T_Σ*, is the set:*

$$C_{\Sigma,T} = \{\phi \rightarrow \psi \mid \phi \in N_{\Sigma,T}, \psi \in D_{\Sigma,T}\}.$$

Of course, non-implicational constraints, ψ, can be expressed as $\bot \rightarrow \psi$. In practice, constraints on $\bot$ and even the presence of features appropriate to $\bot$ are disallowed in order to induce term encodings more easily (Penn 2000).

Because we are not assuming a logic with subsumption (types here can simply be taken as disjunctions of the species they subsume), general negation can be interpreted as set complement, and implication can be interpreted using negation of the antecedent in the usual way. The interpretation of general inequations given here follows the interpretation assumed in ALE's implementation of them. The reason for including fully general inequations in ALE, namely the existence of distinguished *extensional* types, does not apply here, however, although these can easily be added. When the inequated argument is restricted to be a variable, these reduce to path inequations.

DEFINITION 7. *Given a signature, $\Sigma = \langle \mathcal{S}, \mathcal{F}, \mathcal{A} \rangle$, a type basis for Σ, T_Σ, a countable set of variables, V, an interpretation of Σ, $I = \langle U, S, F \rangle$, and a partial variable assignment function, $\alpha : V \rightharpoonup U$, the* description denotation function *is the total function, $[\cdot]_I^\alpha : (D_{\Sigma,T} \cup N_{\Sigma,T}) \to P(U)$ defined such that:*

- $[x]_I^\alpha = \{\alpha(x)\}$,
- $[t]_I^\alpha = \{u \mid t \sqsubseteq S(u)\}$, *for all* $t \in T_\Sigma$,
- $[f{:}\phi]_I^\alpha = \{u \mid F(f)(u)\downarrow$ *and* $F(f)(u) \in [\phi]_I^\alpha\}$,
- $[\exists x.\phi]_I^\alpha = \bigcup_{u \in U} [\phi]_I^{\alpha[x \mapsto u]}$,
- $[\phi_1 \wedge \phi_2]_I^\alpha = [\phi_1]_I^\alpha \cap [\phi_2]_I^\alpha$,
- $[\phi_1 \vee \phi_2]_I^\alpha = [\phi_1]_I^\alpha \cup [\phi_2]_I^\alpha$, *and*
- $[\not\approx\phi]_I^\alpha = \{u \mid \exists v \in [\phi]_I^\alpha . v \neq u\}$.

We can use one function for both antecedent and consequent denotations because they agree in value on their intersection.

DEFINITION 8. *Given a signature, $\Sigma = \langle \mathcal{S}, \mathcal{F}, \mathcal{A} \rangle$, a type basis for Σ, T_Σ, a countable set of variables, V, and an interpretation of Σ, $I = \langle U, S, F \rangle$, the* constraint denotation function *is the total function, $[\cdot]_I^\alpha : C_{\Sigma,T} \to P(U)$ defined such that:*

$$[\phi \to \psi]_I = \bigcup_{\alpha : V \to U} ((U \backslash [\phi]_I^\alpha) \cup [\psi]_I^\alpha).$$

We are now in a position to state what our grammar signatures, effectively King signatures plus a collection of implicational constraints, mean and what we are interested in preserving in the transformation presented below. These collections of implicational constraints are interpreted conjunctively and universally as in SRL:

DEFINITION 9. *A* grammar signature *is a triple, $\Gamma = \langle \Sigma, T, \theta \rangle$, where Σ is a King signature, T is a type basis for Σ, and $\theta \subseteq C_{\Sigma,T}$ is a* constraint system.

DEFINITION 10. *Given a grammar signature, $\Gamma = \langle \Sigma, T, \theta \rangle$, a countable set of variables, V, and an interpretation of Σ, $I = \langle U, S, F \rangle$, the* constraint system denotation function *is the total function, $\Theta_{\Sigma,T} : P(C_{\Sigma,T}) \to$*

$P(U)$ such that:

$$\Theta_{\Sigma,T}(\theta) = \bigcap_{\gamma \in \theta} [\gamma]_I$$

DEFINITION 11. *Given a grammar signature, $\Gamma = \langle \Sigma, T, \theta \rangle$, $I = \langle U, S, F \rangle$ is a* model *of Γ iff:*

- *I is an interpretation of Σ, and*
- *$\Theta_{\Sigma,T}(\theta) = U$.*

DEFINITION 12. *A grammar signature, Γ, is* modellable *iff there exists an interpretation, I, that is a model of Γ.*

This notion of modellability is taken from King 1999. What we absolutely need to preserve is the condition that a transformed grammar signature, Γ', is modellable iff the original grammar signature, Γ, is modellable. Below, a stronger condition is proven, namely that the models of Γ are precisely the models of Γ'.

3. THE TYPE-ANTECEDENT TRANSFORMATION

The transformation, represented here as $\Rightarrow$, converts any antecedent to the form $\tau|\psi$, where $\tau \in T$, and $\psi \in D_{\Sigma,T}$. Given a constraint, $\phi \rightarrow \psi$, and a witness to the transformation $\phi \Rightarrow \tau|\psi'$, it is trivial to show that:

$$[\phi \rightarrow \psi]_I = [\tau \rightarrow (\psi' \vee \psi)]_I$$

and thence that a constraint system of transformed constraints has the same interpretations as the constraint system of original constraints. All of this hinges exclusively on showing that:

PROPOSITION 1. *Given a countably infinite set of variables, V, for any King signature, Σ, with type basis, T_Σ, for any interpretation, I, of Σ and variable assignment function, $\alpha : V \rightarrow U$, and for any antecedent $\phi \in N_{\Sigma,T}$ such that $\phi \Rightarrow \tau|\psi$, where $\tau \in T$ and $\psi \in D_{\Sigma,T}$:*

$$[\phi]_I^\alpha = [\tau]_I^\alpha \cap (U \backslash [\psi]_I^\alpha)$$

In other words, $\tau|\psi'$ can be thought of as $\tau \wedge \neg\psi'$, and $\tau \rightarrow \psi'$, as being logically equivalent to $\neg\phi$.
Proof: The proof is by induction on the structure of the witness to the transformation $\phi \Rightarrow \tau|\psi$. This witness can be constructed from the calculus that defines $\Rightarrow$, presented below. Along with each inference rule, the corresponding step in the proof will be shown.

It will be assumed that every antecedent is presented in disjunctive-prenex normal form, i.e.:

$$\phi = \bigvee_{i=1}^{n} \exists x_1^i . \ldots . \exists x_{m_i}^i . \phi_i ,$$

where each ϕ_i is a disjunction-free, quantifier-free, inequation-free description. The derivation of this normal form and its preservation of $[\phi]_I^\alpha$ are omitted here. Within the quantified expressions of this normal form, we will also require a generalised form of the transformation, $\phi_i \Rightarrow_\beta^X \langle \tau_i, \psi_i, \bar{\phi}_i \rangle$, where $X \subseteq V$ is finite and $\beta : X \rightarrow P(V)$. For this transformation, the following generalisation of the proposition applies:

LEMMA 1. *Given a countably infinite set of variables, V, for any King signature, Σ, with type basis, T_Σ, for any interpretation, I, of Σ and variable assignment function, $\alpha : V \rightarrow U$, and for any antecedent $\phi \in N_{\Sigma,T}$ such that $\phi \Rightarrow_\beta^X \langle \tau, \psi, \bar{\phi} \rangle$, where $\tau \in T$ and $\psi, \bar{\phi} \in D_{\Sigma,T}$, $X \subseteq V$ is finite and $\beta : X \rightarrow P(V)$:*

$$[\exists_{x \in X} x . \phi]_I^\alpha = [\tau]_I^\alpha \cap (U \backslash [\psi]_I^\alpha) \cap \bigcap_{x \in X} \bigcap_{y,z \in \beta(x), distinct} (U \backslash [[y/\not\approx z]\bar{\phi}]_I^\alpha).$$

Given this lemma, the following two rules apply to the basic transformation itself:

$$\vee_\Rightarrow : \frac{\forall 1 \leq i \leq n . [\phi_i \Rightarrow \tau_i | \psi_i]}{\bigvee_{i=1}^n \phi_i \Rightarrow (\sqcap_{i=1}^n \tau_i) | (\bigwedge_{i=1}^n (\bigvee_{s \in S, \tau_i \not\sqsubseteq s} s \vee \psi_i)}$$

Step $\vee_\Rightarrow$: By the inductive hypothesis, for all $1 \leq i \leq n$, $[\phi_i]_I^\alpha = [\tau_i]_I^\alpha \cap (U \backslash [\psi_i]_I^\alpha)$. So:

$$\begin{aligned} [\textstyle\bigvee_{i=1}^n \phi_i]_I^\alpha &= \textstyle\bigcup_{i=1}^n ([\tau_i]_I^\alpha \cap (U \backslash [\psi_i]_I^\alpha)) \\ &= [\textstyle\sqcap_{i=1}^n \tau_i]_I^\alpha \cap \bigcup_{i=1}^n ([\tau_i]_I^\alpha \cap (U \backslash [\psi_i]_I^\alpha)), \end{aligned}$$

since $\bigcup_{i=1}^n ([\tau_i]_I^\alpha \cap (U \backslash [\psi_i]_I^\alpha)) \subseteq \bigcup_{i=1}^n [\tau_i]_I^\alpha \subseteq [\sqcap_{i=1}^n \tau_i]_I^\alpha$. Then

$$\begin{aligned} [\textstyle\bigvee_{i=1}^n \phi_i]_I^\alpha &= [\textstyle\sqcap_{i=1}^n \tau_i]_I^\alpha \cap U \backslash \bigcap_{i=1}^n (U \backslash [\tau_i]_I^\alpha \cup [\psi_i]_I^\alpha) \\ &= [\textstyle\sqcap_{i=1}^n \tau_i]_I^\alpha \cap U \backslash \bigcap_{i=1}^n (\bigcup_{s \in S, \tau_i \not\sqsubseteq s} [s]_I^\alpha \cup [\psi_i]_I^\alpha) \\ &= [\textstyle\sqcap_{i=1}^n \tau_i]_I^\alpha \cap U \backslash [\bigwedge_{i=1}^n (\bigvee_{s \in S, \tau_i \not\sqsubseteq s} s \vee \psi_i)]_I^\alpha . \end{aligned}$$

$$\exists_\Rightarrow : \frac{\phi \Rightarrow_\beta^{\{x_1, \ldots, x_n\}} \langle \tau, \psi, \bar{\phi} \rangle}{\exists x_1 . \ldots . \exists x_n . \phi \Rightarrow \tau | (\psi \vee \bigvee_{1 \leq i \leq n} \bigvee_{y,z \in \beta(x_i)} [y/\not\approx z] \phi)}$$

Step $\exists_\Rightarrow$: This is an immediate consequence of the lemma.

$$\tau_{\Rightarrow^X_\beta} : \frac{}{\tau \Rightarrow^X_{\beta_\emptyset} \langle \tau, \top, \tau \rangle}$$

where $\beta_\emptyset$ is the function defined such that for all $x \in X$, $\beta_\emptyset(x) = \emptyset$.
Step $\tau_{\Rightarrow^X_\beta}$: The lemma trivially holds.

$$\text{free}_{\Rightarrow^X_\beta} : \frac{}{x \Rightarrow^X_{\beta_\emptyset} \langle \bot, \not\approx x, x \rangle} \; x \notin X$$

Step free$_{\Rightarrow^X_\beta}$: $[\exists_{y \in X}.x]^\alpha_I = [x]^\alpha_I = \{\alpha(x)\}$ since $x \notin X$. Now, since $\beta = \beta_\emptyset$:

$$\begin{aligned} & [\tau]^\alpha_I \cap (U \backslash [\psi]^\alpha_I) \cap \bigcap_{x \in X} \bigcap_{y,z \in \beta(x),\text{distinct}} (U \backslash [[y/\not\approx z]\bar{\phi}]^\alpha_I) \\ = \; & [\tau]^\alpha_I \cap (U \backslash [\psi]^\alpha_I) \\ = \; & [\bot]^\alpha_I \cap (U \backslash [\not\approx x]^\alpha_I) \\ = \; & U \cap (U \backslash \{u \mid \exists v \in [x]^\alpha_I . v \neq u\}) \\ = \; & U \backslash \{u \mid \exists v \in \{\alpha(x)\}. v \neq u\} \\ = \; & U \backslash (U \backslash \{\alpha(x)\}) \\ = \; & \{\alpha(x)\}. \end{aligned}$$

$$\text{bound}_{\Rightarrow^X_\beta} : \frac{}{x \Rightarrow^X_{[x \mapsto \{y\}]} \langle \bot, \top, y \rangle} \; x \in X, y \in V \text{ new}$$

Step bound$_{\Rightarrow^X_\beta}$:

$$\begin{aligned} [\exists x.x]^\alpha_I & = \bigcup_{u \in U} [x]^{\alpha[x \mapsto u]}_I \\ & = U. \end{aligned}$$

Since $\beta(x)$ is a singleton set:

$$\begin{aligned} & [\tau]^\alpha_I \cap (U \backslash [\psi]^\alpha_I) \cap \bigcap_{x \in X} \bigcap_{y,z \in \beta(x),\text{distinct}} (U \backslash [[y/\not\approx z]\bar{\phi}]^\alpha_I) \\ = \; & [\tau]^\alpha_I \cap (U \backslash [\psi]^\alpha_I) \\ = \; & [\bot]^\alpha_I \cap (U \backslash [\top]^\alpha_I) \\ = \; & U. \end{aligned}$$

$$f_{\Rightarrow^X_\beta} : \frac{\phi \Rightarrow^X_\beta \langle \tau, \psi, \bar{\phi} \rangle}{f{:}\phi \Rightarrow^X_\beta \langle \mathit{Intro}(f), f{:}(\bigvee_{s \in S, \tau \not\sqsubseteq s} s \vee \psi), f{:}\bar{\phi} \rangle}$$

Step $f_{\Rightarrow^X_\beta}$: By the inductive hypothesis:

$$\begin{aligned} & [\exists x_1 . \ldots . \exists x_n . \phi]^\alpha_I = \bigcup_{u_1 \in U} \cdots \bigcup_{u_n \in U} [\phi]^{\alpha[x_1 \mapsto u_1, \ldots, x_n \mapsto u_n]}_I \\ & = [\tau]^\alpha_I \cap (U \backslash [\psi]^\alpha_I) \cap \bigcap_{x \in \{x_1, \ldots x_n\}} \bigcap_{y,z \in \beta(x),\text{distinct}} (U \backslash [[y/\not\approx z]\bar{\phi}]^\alpha_I), \end{aligned}$$

where $X = \{x_1, \ldots, x_n\}$. Now:

$$
\begin{aligned}
&[\exists x_1 \ldots . \exists x_n . f{:}\phi]^\alpha_I = \bigcup_{u_1 \in U} \cdots \bigcup_{u_n \in U} [f{:}\phi]^{\alpha[x_1 \mapsto u_1, \ldots, x_n \mapsto u_n]}_I \\
&= \bigcup_{u_1 \in U} \cdots \bigcup_{u_n \in U} \{u \mid F(f)(u){\downarrow} \text{ and } F(f)(u) \in [\phi]^{\alpha[x_1 \mapsto u_1, \ldots, x_n \mapsto u_n]}_I\} \\
&= \{u \mid F(f)(u){\downarrow} \text{ and } F(f)(u) \in \bigcup_{u_1 \in U} \cdots \bigcup_{u_n \in U} [\phi]^{\alpha[x_1 \mapsto u_1, \ldots, x_n \mapsto u_n]}_I\} \\
&= \{u \mid F(f)(u){\downarrow}\ \} \cap \{u \mid F(f)(u) \in \bigcup_{u_1 \in U} \cdots \bigcup_{u_n \in U} [\phi]^{\alpha[x_1 \mapsto u_1, \ldots, x_n \mapsto u_n]}_I\} \\
&= [\mathit{Intro}(f)]^\alpha_I \cap \\
&\quad \{u \mid F(f)(u) \in ([\tau]^\alpha_I \cap (U \backslash [\psi]^\alpha_I) \cap \bigcap_{x \in \{x_1, \ldots x_n\}} \bigcap_{y,z \in \beta(x), \text{distinct}} (U \backslash [[y/\not\approx z]\bar{\phi}]^\alpha_I))\} \\
&= [\mathit{Intro}(f)]^\alpha_I\ \cap \{u \mid F(f)(u) \in [\tau]^\alpha_I\} \cap \{u \mid F(f)(u) \in (U \backslash [\psi]^\alpha_I)\} \cap \\
&\quad \{u \mid F(f)(u) \in \bigcap_{x \in \{x_1, \ldots x_n\}} \bigcap_{y,z \in \beta(x), \text{distinct}} (U \backslash [[y/\not\approx z]\bar{\phi}]^\alpha_I)\} \\
&= [\mathit{Intro}(f)]^\alpha_I\ \cap ([\mathit{Intro}(f)]^\alpha_I \cap U \backslash \{u \mid F(f)(u) \in U \backslash [\tau]^\alpha_I\}) \cap \\
&\quad ([\mathit{Intro}(f)]^\alpha_I \cap U \backslash \{u \mid F(f)(u) \in [\psi]^\alpha_I\}) \cap \\
&\quad \bigcap_{x \in \{x_1, \ldots x_n\}} \bigcap_{y,z \in \beta(x), \text{distinct}} \{u \mid F(f)(u) \in (U \backslash [[y/\not\approx z]\bar{\phi}]^\alpha_I)\} \\
&= [\mathit{Intro}(f)]^\alpha_I\ \cap U \backslash \{u \mid F(f)(u) \in U \backslash [\tau]^\alpha_I\} \cap U \backslash \{u \mid F(f)(u) \in [\psi]^\alpha_I\} \cap \\
&\quad \bigcap_{x \in \{x_1, \ldots x_n\}} \bigcap_{y,z \in \beta(x), \text{distinct}} ([\mathit{Intro}(f)]^\alpha_I \cap U \backslash \{u \mid F(f)(u) \in [[y/\not\approx z]\bar{\phi}]^\alpha_I\}) \\
&= [\mathit{Intro}(f)]^\alpha_I\ \cap U \backslash (\{u \mid F(f)(u) \in U \backslash [\tau]^\alpha_I\} \cup \{u \mid F(f)(u) \in [\psi]^\alpha_I\}) \cap \\
&\quad \bigcap_{x \in \{x_1, \ldots x_n\}} \bigcap_{y,z \in \beta(x), \text{distinct}} U \backslash \{u \mid F(f)(u) \in [[y/\not\approx z]\bar{\phi}]^\alpha_I\} \\
&= [\mathit{Intro}(f)]^\alpha_I\ \cap U \backslash \{u \mid F(f)(u) \in (U \backslash [\tau]^\alpha_I \cup [\psi]^\alpha_I)\} \cap \\
&\quad \bigcap_{x \in \{x_1, \ldots x_n\}} \bigcap_{y,z \in \beta(x), \text{distinct}} U \backslash [[y/\not\approx z] f{:}\bar{\phi}]^\alpha_I \\
&= [\mathit{Intro}(f)]^\alpha_I\ \cap U \backslash \{u \mid F(f)(u) \in ([\bigvee_{s \in S, \tau \not\sqsubseteq s} s]^\alpha_I \cup [\psi]^\alpha_I)\} \cap \\
&\quad \bigcap_{x \in \{x_1, \ldots x_n\}} \bigcap_{y,z \in \beta(x), \text{distinct}} U \backslash [[y/\not\approx z] f{:}\bar{\phi}]^\alpha_I \\
&= [\mathit{Intro}(f)]^\alpha_I\ \cap U \backslash [f : (\bigvee_{s \in S, \tau \not\sqsubseteq s} s \vee \psi)]^\alpha_I \cap \\
&\quad \bigcap_{x \in \{x_1, \ldots x_n\}} \bigcap_{y,z \in \beta(x), \text{distinct}} U \backslash [[y/\not\approx z] f{:}\bar{\phi}]^\alpha_I .
\end{aligned}
$$

The crucial case is conjunction:

$$
\wedge_{\Rightarrow^X_\beta} :\ \frac{\phi_1 \Rightarrow^X_{\beta_1} \langle \tau_1, \psi_1, \bar{\phi}_1 \rangle \quad \phi_2 \Rightarrow^X_{\beta_2} \langle \tau_2, \psi_2, \bar{\phi}_2 \rangle}{\phi_1 \wedge \phi_2 \Rightarrow^X_{(\beta_1 \cup \beta_2)} \langle (\tau_1 \sqcup \tau_2), (\psi_1 \vee \psi_2), (\bar{\phi}_1 \wedge \bar{\phi}_2) \rangle}
$$

where $(\beta_1 \cup \beta_2)(x) = \beta_1(x) \cup \beta_2(x)$.

Step $\wedge_{\Rightarrow^X_\beta}$:

$$
\begin{aligned}
&[\exists x_1 . \ldots . \exists x_n . \phi_1 \wedge \phi_2]^\alpha_I = \bigcup_{u_1 \in U} \cdots \bigcup_{u_n \in U} [\phi_1 \wedge \phi_2]^{\alpha[x_1 \mapsto u_1, \ldots, x_n \mapsto u_n]}_I \\
&= \bigcup_{u_1 \in U} \cdots \bigcup_{u_n \in U} ([\phi_1]^{\alpha[x_1 \mapsto u_1, \ldots, x_n \mapsto u_n]}_I \cap [\phi_2]^{\alpha[x_1 \mapsto u_1, \ldots, x_n \mapsto u_n]}_I) \\
&= \bigcup_{u_1 \in U} \cdots \bigcup_{u_n \in U} [\phi_1]^{\alpha[x_1 \mapsto u_1, \ldots, x_n \mapsto u_n]}_I \cap \bigcup_{u_1 \in U} \cdots \bigcup_{u_n \in U} [\phi_2]^{\alpha[x_1 \mapsto u_1, \ldots, x_n \mapsto u_n]}_I \\
&\qquad \cap \bigcap_{x \in \{x_1, \ldots, x_n\}} \bigcap_{y \in \beta_1(x), z \in \beta_2(x)} U \backslash [[y/\not\approx z](\bar{\phi}_1 \wedge \bar{\phi}_2)]^\alpha_I),
\end{aligned}
$$

where $X = \{x_1, \ldots, x_n\}$. $\bar{\phi}_1$ and $\bar{\phi}_2$ are copies of ϕ_1 and ϕ_2 in which every instance of a bound variable has been replaced by a new free variable. β_1 and β_2 record those new variables for each bound variable. We can break apart the denotation of $\phi_1 \wedge \phi_2$ into the denotations of ϕ_1 and ϕ_2

but we will lose the information about variables that are shared between ϕ_1 and ϕ_2. For most variables, this does not matter, because the variable assignment function preserves that information — provided that we consistently use the same assignment. Existentially quantified variables, however, are special: when we commute the intersection arising from conjunction with the unions arising due to the existential quantifications, it could be that our intersection will pair off objects that were obtained with different assignment functions. We can correct for this by removing, for each pair of new variables corresponding to the same bound variable in $\bar{\phi}_1$ and $\bar{\phi}_2$, respectively, all of those objects that do not share the same values for that pair. This can be achieved using inequations without a universal quantifier in the transformed formula, and we can always use the assignment function, α, to do it because $\bar{\phi}_1$ and $\bar{\phi}_2$ have no occurences of any bound variable.

Using the inductive hypothesis:

$$\begin{aligned}
&\bigcup_{u_1 \in U} \cdots \bigcup_{u_n \in U} [\phi_1]_I^{\alpha[x_1 \mapsto u_1, \ldots, x_n \mapsto u_n]} \cap \bigcup_{u_1 \in U} \cdots \bigcup_{u_n \in U} [\phi_2]_I^{\alpha[x_1 \mapsto u_1, \ldots, x_n \mapsto u_n]} \\
&\qquad \cap \bigcap_{x \in \{x_1, \ldots, x_n\}} \bigcap_{y \in \beta_1(x), z \in \beta_2(x)} U \backslash [[y / \not\approx z](\bar{\phi}_1 \wedge \bar{\phi}_2)]_I^{\alpha}). \\
&= [\exists x_1 . \ldots \exists x_n . \phi_1]_I^{\alpha} \cap [\exists x_1 . \ldots \exists x_n . \phi_2]_I^{\alpha} \\
&\qquad \cap \bigcap_{x \in \{x_1, \ldots, x_n\}} \bigcap_{y \in \beta_1(x), z \in \beta_2(x)} U \backslash [[y / \not\approx z](\bar{\phi}_1 \wedge \bar{\phi}_2)]_I^{\alpha}) \\
&= [\tau_1]_I^{\alpha} \cap (U \backslash [\psi_1]_I^{\alpha}) \cap \bigcap_{x \in X} \bigcap_{y, z \in \beta_1(x), \mathrm{distinct}} (U \backslash [[y / \not\approx z] \bar{\phi}_1]_I^{\alpha}) \cap \\
&\quad [\tau_2]_I^{\alpha} \cap (U \backslash [\psi_2]_I^{\alpha}) \cap \bigcap_{x \in X} \bigcap_{y, z \in \beta_2(x), \mathrm{distinct}} (U \backslash [[y / \not\approx z] \bar{\phi}_2]_I^{\alpha}) \cap \\
&\qquad \bigcap_{x \in \{x_1, \ldots, x_n\}} \bigcap_{y \in \beta_1(x), z \in \beta_2(x)} U \backslash [[y / \not\approx z](\bar{\phi}_1 \wedge \bar{\phi}_2)]_I^{\alpha}) \\
&= [\tau_1]_I^{\alpha} \cap (U \backslash [\psi_1]_I^{\alpha}) \cap [\tau_2]_I^{\alpha} \cap (U \backslash [\psi_2]_I^{\alpha}) \cap \\
&\qquad \bigcap_{x \in X} \bigcap_{y, z \in (\beta_1 \cup \beta_2)(x), \mathrm{distinct}} (U \backslash [[y / \not\approx z](\bar{\phi}_1 \wedge \bar{\phi}_2]_I^{\alpha})) \\
&= [\tau_1 \sqcup \tau_2]_I^{\alpha} \cap (U \backslash [\psi_1 \vee \psi_2]_I^{\alpha}) \cap \\
&\qquad \bigcap_{x \in X} \bigcap_{y, z \in (\beta_1 \cup \beta_2)(x), \mathrm{distinct}} (U \backslash [[y / \not\approx z](\bar{\phi}_1 \wedge \bar{\phi}_2]_I^{\alpha})).
\end{aligned}$$

$\beta_1(x)$ and $\beta_2(x)$ are always distinct because step $\mathrm{free}_{\Rightarrow^X_\beta}$ uses new variables, and we can refine the contributions of the inductive step to $\phi_1 \wedge \phi_2$ since objects in $[\phi_1]_I^{\alpha}$ or $[\phi_2]_I^{\alpha}$ that are not in $[\phi_1 \wedge \phi_2]_I^{\alpha}$ have already been excluded. The inductive step takes care of the shared variables inside ϕ_1 and ϕ_2 and the current step adds the missing cases that straddle ϕ_1 and ϕ_2. □

4. Combining Inequations, Variables and General Negation

If we were to add inequations of general descriptions and general negation to the same language such that negation could take scope over inequations, then we could form descriptions such as $\neg \not\approx \phi$. Using the semantics given here for inequations, and assuming the usual set-theoretic treatment of the negation operator, we obtain:

$$\begin{aligned}[\neg\not\approx\phi]_I^\alpha &= U\backslash\{u \mid \exists v \in [\phi]_I^\alpha . v \neq u\} \\ &= \{u \mid \forall v \in [\phi]_I^\alpha . v = u\} \\ &= \begin{cases} U & \text{if } |[\phi]_I^\alpha| = 0 \\ [\phi]_I^\alpha & \text{if } |[\phi]_I^\alpha| = 1 \\ \emptyset & \text{if } |[\phi]_I^\alpha| > 1 \end{cases}\end{aligned}$$

As a result, this combination would allow us to test whether a particular consistent description denotes one or more objects in an interpretation. If ϕ is a species, for example, we could test whether there is one object of that species or more than one object. This is a significant and awkward extension of the power of the description language. If inequations are restricted to path inequations, or, equivalently, if the inequated description is constrained to be a free variable, x, then we will always know that $[\phi]_I^\alpha = [x]_I^\alpha = \{\alpha(x)\}$ is a singleton set. That removes the extra expressive power.

A similar effect emerges if we permit general inequations, or even inequations of only variables, to occur within the scope of the existential quantifier that we defined for our antecedent language. In such a language, $\exists x.\not\approx x$ is a well-formed description and:

$$\begin{aligned}[\exists x.\not\approx x]_I^\alpha &= \textstyle\bigcup_{u\in U}[\not\approx x]_I^{\alpha[x\mapsto u]} \\ &= \textstyle\bigcup_{u\in U} U\backslash\{\alpha[x \mapsto u](x)\} \\ &= \textstyle\bigcup_{u\in U} U\backslash\{u\} \\ &= \begin{cases} \emptyset & \text{if } |U| \leq 1 \\ U & \text{if } |U| > 1 \end{cases}\end{aligned}$$

Similarly, a description such as $\exists x.(f{:}\not\approx x \wedge g{:}\not\approx x)$, when negated, selects models that have at most two elements. As mentioned earlier, path inequations could be added to the antecedent language, because these lack variables that could be bound by the existential quantifier.

References

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

Carpenter, B. and G. Penn (1996). Compiling typed attribute-value logic grammars. In H. Bunt and M. Tomita (Eds.), *Recent Advances in Parsing Technologies*, pp. 145–168. Kluwer.

Kasper, R. (1987). A unification method for disjunctive feature structures. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics (ACL-87)*, pp. 235–242.

King, P. (1989). *A Logical Formalism for Head-driven Phrase Structure Grammar.* Ph. D. thesis, University of Manchester.

King, P. J. (1999). Towards Truth in Head-driven Phrase Structure Grammar. In V. Kordoni (Ed.), *Tübingen Studies in Head-Driven Phrase Structure Grammar*, Arbeitspapiere des SFB 340, Nr. 132, Volume 2, pp. 301–352. Eberhard-Karls-Universität Tübingen.

Penn, G. (2000). *The Algebraic Structure of Attributed Type Signatures.* Ph. D. thesis, Carnegie Mellon University.

Pollard, C. and I. Sag (1987). *Information-based Syntax and Semantics.* Number 13 in CSLI Lecture Notes. CSLI Publications.

Pollard, C. and I. Sag (1994). *Head-driven Phrase Structure Grammar.* Chicago.