THREE PROCESSES IN NATURAL LANGUAGE INTERPRETATION

TIM FERNANDO

Abstract. To address complications involving ambiguity, presupposition and implicature, three processes underlying natural language interpretation are isolated: translation, entailment and attunement. A meta-language integrating these processes is outlined, elaborating on a proof-theoretic approach to presupposition.

§1. Introduction. However outrageous Montague's slogan "English as a formal language" [24] may sound, the pressure to push the claim as far as it can go is, for many, irresistible. Basic to Montague's understanding of a formal language is the possibility of a model-theoretic interpretation — of obvious interest in various applications (e.g. databases) that employ models. But formulas of predicate logic (first-order or higher-order, modal or otherwise) differ significantly from English sentences marked with ambiguity and presupposition. Consider, for instance,

(s) If Sylvester gets holds of a canary, <u>the bird</u> will perish.

A very rough translation of (s) that falls short of predicate logic is

$$(\exists y)(\mathit{canary}(y) \wedge \mathit{SylvesterGets}(y)) \ \supset\ \mathit{willPerish}(\underline{\textsf{the bird}})\ ,$$

from which we might expect to extract various readings of (s), depending on how we treat the ill-formed expression <u>the bird</u>. For now, let us focus on the reading that construes <u>the bird</u> anaphorically as any canary Sylvester gets hold of. The obvious rendering in predicate logic,

$$(\forall y)\ (\mathit{canary}(y) \wedge \mathit{SylvesterGets}(y)) \supset \mathit{willPerish}(y)\ ,$$

moves $\exists y$ out of the antecedent to bind <u>the bird</u>, turning $\exists$ into $\forall$. An alternative leaving $\exists y$ intact is provided, under a "propositions-as-types" interpretation of existential quantification $(\exists x \in A)B$ as the dependent sum/product

$$(\textstyle\sum x : A)B \ = \ \{\langle a, b\rangle \mid a \text{ in } A \text{ and } b \text{ in } B[x \mapsto a]\}$$

As wide ranging as Solomon Feferman's work on logic is, natural language is not among his driving concerns. Nevertheless, I would like to think that studying papers such as Feferman [8, 9, 10] has helped me draw certain conceptual distinctions useful in natural language interpretation. And, as a grateful student of his, I want to try to say why. I am indebted to two anonymous referees for helpful comments, and to the Feferfest committee for the opportunity of contributing to this volume. How I regret that I could not properly thank Jon Barwise.

and implication $A \supset B$ as the dependent function space

$$(\prod x : A)B \ = \ \{\text{functions mapping } a \text{ in } A \text{ to some } b \text{ in } B[x \mapsto a]\} \ ,$$

by translating (s) to

$$(1) \qquad (\prod x : (\sum y : \mathit{canary}) \ \mathit{SylvesterGets}(y)) \ \mathit{willPerish}(lx)$$

where lx is the left projection of *any* proof x of $(\sum y : \mathit{canary}) \ \mathit{SylvesterGets}(y)$. Details are supplied in §2 below, following for the most part Martin-Löf type theory [22], where such translations have been explored extensively (Ranta [25]; see also Sundholm [27]), and where (1) is well-formed relative to a context Γ such that

$$\Gamma, \ x : (\sum y : \mathit{canary}) \ \mathit{SylvesterGets}(y) \ \Rightarrow \ \mathit{willPerish}(lx) \ \mathsf{type}$$

(with $\Rightarrow$ separating the context to its left from the assertion to its right, and A type saying 'A is well-formed'). To appreciate the novelty in using a (parametric) proof to construct (1), observe that in predicate logic (be it classical or intuitionistic), the notion of a well-formed formula does not depend on proofs or truth (at least not directly on that which the logic *subsequently* formalizes). By contrast, underlying the toy translation (1) of (s) is the claim that

(†) well-formedness may depend on parametric proofs — or, in symbols,

$$x_1 : A_1, \ \ldots \ , \ x_n : A_n \ \Rightarrow \ A \ \mathsf{type} \ .$$

(†) acknowledges the need for entailments to resolve problematic expressions such as the bird — the step from (s) to (1) relying (also) on all canaries being birds. The present paper elaborates on applications of (†) to natural language, addressing three questions:

(Q1) How can we keep the task of constructing logical forms managable, when proofs (implicated in that task by (†)) can get arbitrarily complicated?
(Q2) How can we interpret the parametricity in (†) model-theoretically?
(Q3) How can we bring natural language into the picture so as to constrain choices of A in (†)? (Just where is the English?)

1.1. Outline. Questions (Q1)-(Q3) are analyzed below by construing (†) as an attack on *presupposition*, and factoring that analysis into three steps specifying respectively: what presuppositions are, how they can be satisfied, and what to do if they can't. Towards that end, §2 formalizes a fragment of type theory given roughly by first-order formulas over a relational signature, with terms assembled from proofs supplied by context. §2 concludes with an answer to (Q2), on which §3 builds a reply to (Q1). Complications to the idealization assumed in §3 are exposed in §4, where (Q3) is taken up.

The attention paid to type theory is largely for the sake of concreteness. §4 applies also to other approaches (dealing say, with (s)), and can, up to a point, be read independently of the two preceding sections. Interest in the general line investigated in §§2 and 3 does, however, seem to be growing — witness, for instance, Ahn and Kolb [2], Kamareddine and Klein [15], Fox [13], Krause [21], Borghuis [6], Krahmer and Piwek [20], Cooper [7], Ranta [26] — and it is natural to relate this to *explicit mathematics* (Feferman [8]). Explicit mathematics provides in §2 the basis for an answer to (Q2) as well as a transparent treatment of subtypes, without any need for type coercion.[1]

1.2. From entailment to translation and attunement. Turning to the main part of the paper, an approach to natural language semantics that motivates a good deal of §§3 and 4 is *Discourse Representation Theory* (DRT, Kamp and Reyle [16]), addressing, as it does, the nagging question behind (Q3) of just how logical forms (or discourse representations) are to be derived from natural language text. The elusive answer is referred to in DRT as the *construction algorithm*. In an early paper, Hans Kamp writes

> it seems to me that the rules for the construction of discourse representations have at least as good a claim to being constitutive of meaning as the clauses that make up the definition of truth. ([17], page 409)

About a decade later, he reflects

> The principal challenge to DRT, experience indicates, is the exact formulation of the construction algorithm. ([18], page 37)

To meet this challenge, various proposals have been put forward, including appeals to rhetorical relations and defeasible entailments (e.g. Asher and Lascarides [3]). Abstracting over these proposals, the present paper investigates the construction algorithm in terms of a process of *translation*, conditioned by processes of *entailment* and *attunement*. Very briefly, the thrust of §§3 and 4 is to factor into the entailments mentioned in (†) the roles played by translation and attunement in natural language interpretation.

§2. An elementary system of dependent types.

Let L be a (many-sorted, relational) *signature* that specifies *sorts* $(U, \dots)$ and *relation symbols* $(R, \dots)$ with associated arities. Let us agree to write $U \in L_0$ to mean U is a sort in L, and $R \in L(U_1 \cdots U_n)$ to mean that R is a relation symbol in L with n arguments of sorts U_1 to U_n. This section formalizes an elementary subsystem corresponding *roughly* to first-order L-formulas φ generated according to

$$\varphi \ ::= \ \bot \mid R(t, \dots, t) \mid \varphi \wedge \varphi \mid \varphi \supset \varphi \mid (\exists x \in U)\varphi \mid (\forall x \in U)\varphi$$

where $U \in L_0$ and $R \in L(U_1 \cdots U_n)$ for some $U_1, \dots, U_n \in L_0$. What is "rough" here is that some care must be taken in specifying how $R(t_1, \dots, t_n)$ is formed contextually from proof terms t_i.

[1]This much is implicit in the reference in Fox [13] to *Frege structures* (Aczel [1]). I hasten to add that Feferman's influence on the present work reaches beyond these applications, and into §4 (as I explain in §5.2).

2.1. Context and formation rules. As usual, contexts are finite sequences of variable typings, built from the empty sequence ϵ, the typings in which seep through $\Rightarrow$.

$$\frac{}{\epsilon\ \mathsf{context}} \qquad \frac{\Gamma \Rightarrow A\ \mathsf{type}}{\Gamma, x\!:\!A\ \mathsf{context}}\ x \not\in \mathsf{Var}(\Gamma) \qquad \frac{\Gamma\ \mathsf{context}}{\Gamma \Rightarrow x\!:\!A}\ \text{`}x\!:\!A\text{' in } \Gamma\ .$$

Suppressing the subscript L on $\Rightarrow_L$ for simplicity, let us pick out a subcollection wff of type corresponding to the aforementioned L-formulas φ

$$\frac{\Gamma\ \mathsf{context}}{\Gamma \Rightarrow U\ \mathsf{type}}\ U \in L_0 \qquad \frac{\Gamma \Rightarrow A\ \mathsf{wff}}{\Gamma \Rightarrow A\ \mathsf{type}}$$

with $\bot$ and atomic L-formulas from

$$\frac{\Gamma\ \mathsf{context}}{\Gamma \Rightarrow \bot\ \mathsf{wff}} \qquad \frac{\Gamma \Rightarrow t_1\!:\!U_1 \quad \cdots \quad \Gamma \Rightarrow t_n\!:\!U_n}{\Gamma \Rightarrow R(t_1,\ldots,t_n)\ \mathsf{wff}}\ R \in L(U_1 \cdots U_n)$$

$\wedge$ and $\supset$ from

$$(\wedge)\ \frac{\Gamma \Rightarrow A\ \mathsf{wff} \qquad \Gamma, x\!:\!A \Rightarrow B\ \mathsf{wff}}{\Gamma \Rightarrow (\sum x\!:\!A)B\ \mathsf{wff}} \qquad (\supset)\ \frac{\Gamma \Rightarrow A\ \mathsf{wff} \qquad \Gamma, x\!:\!A \Rightarrow B\ \mathsf{wff}}{\Gamma \Rightarrow (\prod x\!:\!A)B\ \mathsf{wff}}$$

and $\exists x \in U$ and $\forall x \in U$ from

$$\frac{\Gamma, x\!:\!U \Rightarrow A\ \mathsf{wff}}{\Gamma \Rightarrow (\sum x\!:\!U)A\ \mathsf{wff}}\ U \in L_0 \qquad \frac{\Gamma, x\!:\!U \Rightarrow A\ \mathsf{wff}}{\Gamma \Rightarrow (\prod x\!:\!U)A\ \mathsf{wff}}\ U \in L_0\ .$$

Conventions. For expressions obtained from the rules ($\wedge$) and ($\supset$), where the variable x bound does not occur in B, let us write $A \wedge B$ and $A \supset B$, respectively. Furthermore, for $R \in L(U)$, let us write $(\sum x : R)A$ instead of $(\sum x\!:\!U)(R(x) \wedge A)$, and $(\prod x\!:\!R)A$ instead of $(\prod x\!:\!U)(R(x) \supset A)$.

Example. For the translation (1) of (s) in §1, let *bird*, *willPerish*, *canary* $\in L(U)$ for some $U \in L_0$, so that $(\sum y : \mathit{canary})A$ in (1) abbreviates $(\sum y : U)(\mathit{canary}(y) \wedge A)$. To form the term lx, we need further rules for judgments $t : A$. These are, in contrast to well-formedness judgments A wff, naturally construed as semantic, although they feed into the well-formedness rule above with side condition $R \in L(U_1 \cdots U_n)$, whence the novelty noted in the introduction.

2.2. Further semantic rules. The semantic rules for $\sum$ and $\prod$ are the familiar elimination rules

$$\frac{\Gamma \Rightarrow t\!:\!(\sum x\!:\!A)B}{\Gamma \Rightarrow lt\!:\!A} \qquad \frac{\Gamma \Rightarrow t\!:\!(\sum x\!:\!A)B}{\Gamma \Rightarrow rt\!:\!B[x \mapsto lt]} \qquad \frac{\Gamma \Rightarrow t\!:\!(\prod x\!:\!A)B \qquad \Gamma \Rightarrow t'\!:\!A}{\Gamma \Rightarrow ap(t,t')\!:\!B[x \mapsto t']}$$

(with left and right projections l and r respectively, and application ap), and introduction rules

$$\frac{\Gamma \Rightarrow t\!:\!A \qquad \Gamma \Rightarrow t'\!:\!B[x \mapsto t] \qquad \Gamma, x\!:\!A \Rightarrow B\ \mathsf{type}}{\Gamma \Rightarrow \langle t, t'\rangle\!:\!(\sum x\!:\!A)B}$$

$$\frac{\Gamma, x\!:\!A \Rightarrow t\!:\!B}{\Gamma \Rightarrow \lambda x.t\!:\!(\prod x\!:\!A)B}$$

(with pairing $\langle\cdot,\cdot\rangle$ and λ-abstraction). The force of $\bot$ can be captured by

$$\frac{\Gamma \Rightarrow A \ \mathsf{wff}}{\Gamma \Rightarrow 0\colon \bot \supset A}$$

(saying $\bot$ entails every well-formed formula) with double negation through

$$\frac{\Gamma \Rightarrow A \ \mathsf{wff}}{\Gamma \Rightarrow \delta_A\colon \neg\neg A \supset A}$$

where, as usual, $\neg A$ is $A \supset \bot$. While type subscripts on $l, r, \langle\cdot,\cdot\rangle$, ap, λ and 0 can be suppressed (in accordance with a type-free formulation), double negation δ_A lacks the uniformity for such a treatment (and is, of course, intuitionistically suspect).

2.3. Formal derivations versus interpretations. Given some set $\mathcal{D}$ of rules, as in §§2.1 and 2.2, a *$\mathcal{D}$-derivation* is a derivation (in the usual sense) from the rules in $\mathcal{D}$. Let us write '$\Gamma \Rightarrow^{\mathcal{D}} \Theta$' to mean that there is a $\mathcal{D}$-derivation of '$\Gamma \Rightarrow \Theta$', and set

$$\mathsf{context}^{\mathcal{D}} \;=\; \{\Gamma \mid \text{there is a } \mathcal{D}\text{-derivation of '}\Gamma\ \mathsf{context}\text{'}\}\ ,$$

referring to its elements as *$\mathcal{D}$-contexts*. As $\mathcal{D}$ becomes larger, so does the collection of pairs Γ, A such that

$$\text{for some term } t,\ \ \text{'}\Gamma \ \Rightarrow^{\mathcal{D}}\ t\colon A\text{'}\ ,$$

an alternative approach to which (exploiting the layer of language represented by the signature L) we pursue next.

An *L-(proof-)interpretation* $[\![\cdot]\!]$ is a function mapping every $U \in L_0$ to a set $[\![U]\!]$ and every $R \in L(U_1 \cdots U_n)$ and $u_1 \in [\![U_1]\!], \ldots, u_n \in [\![U_n]\!]$ to a set $[\![R, u_1 \cdots u_n]\!]$. An L-interpretation $[\![\cdot]\!]$ induces the many-sorted L-model M that interprets $U \in L_0$ as $U_M = [\![U]\!]$ and $R \in L(U_1 \cdots U_n)$ as

$$R_M \;=\; \{u_1 \cdots u_n \in [\![U_1]\!] \times \cdots \times [\![U_n]\!] \mid [\![R, u_1 \cdots u_n]\!] \neq \emptyset\}\ ,$$

the intuition being that $[\![R, u_1 \cdots u_n]\!]$ consists of proofs of $R(u_1, \ldots, u_n)$. Combining formal derivation in $\mathcal{D}$ with an L-interpretation $[\![\cdot]\!]$, we can define for every $\mathcal{D}$-context Γ,

(i) its set $[\![\Gamma]\!]^{\mathcal{D}}$ of *$\mathcal{D}$-instantiations* ρ interpreting Γ's variables x as $\rho(x)$

simultaneously (by induction on the length of $\mathcal{D}$-derivations) with

(ii) interpretations $[\![A]\!]^{\mathcal{D}}_{\Gamma,\rho}$ of A such that '$\Gamma \Rightarrow^{\mathcal{D}} A$ type' (for $\rho \in [\![\Gamma]\!]^{\mathcal{D}}$)

and (again for $\rho \in [\![\Gamma]\!]^{\mathcal{D}}$)

(iii) interpretations $[\![t]\!]^{\mathcal{D}}_{\Gamma,\rho}$ of t such that for some A, '$\Gamma \Rightarrow^{\mathcal{D}} t\colon A$'.

Within (i), we set $[\![\epsilon]\!]^{\mathcal{D}}_{\Gamma,\rho} = \{\emptyset\}$ and

$$[\![\Gamma, x\colon A]\!]^{\mathcal{D}}_{\Gamma,\rho} \;=\; \{\rho^x_a \mid \rho \in [\![\Gamma]\!]^{\mathcal{D}} \text{ and } a \in [\![A]\!]^{\mathcal{D}}_{\Gamma,\rho}\}$$

where ρ^x_a is $\rho \cup \{(x,a)\}$, so that, as the base clause of (iii), $[\![x]\!]^{\mathcal{D}}_{\Gamma,\rho} = \rho(x)$. Within (ii), $[\![U]\!]^{\mathcal{D}}_{\Gamma,\rho} = [\![U]\!]$, $[\![\bot]\!]^{\mathcal{D}}_{\Gamma,\rho} = \emptyset$,

$$\begin{aligned} [\![R(t_1,\dots,t_n)]\!]^{\mathcal{D}}_{\Gamma,\rho} &= [\![R, [\![t_1]\!]^{\mathcal{D}}_{\Gamma,\rho}, \dots, [\![t_n]\!]^{\mathcal{D}}_{\Gamma,\rho}]\!] \\ [\![(Qx\!:\!T)A]\!]^{\mathcal{D}}_{\Gamma,\rho} &= (Qt \in [\![T]\!]^{\mathcal{D}}_{\Gamma,\rho})[\![A]\!]^{\mathcal{D}}_{\Gamma,x:A,\rho^x_t} \quad \text{for } Q = \sum, \prod \,, \end{aligned}$$

adopting $\sum, \prod$ notation at the meta-level with : replaced by $\in$. But what meta-theory? We can adopt the framework of Feferman [10] to interpret terms t built from l, r, $\langle\cdot,\cdot\rangle$, ap, λ and 0 in a model of the type-free lambda-calculus (with products), assigning types relative to contexts such that

$(*)$ whenever $\rho \in [\![\Gamma]\!]^{\mathcal{D}}$ and '$\Gamma \Rightarrow^{\mathcal{D}} t\!:\!A$', $[\![t]\!]^{\mathcal{D}}_{\Gamma,\rho} \in [\![A]\!]^{\mathcal{D}}_{\Gamma,\rho}$.

As for double negation δ_A, let us proceed more generally and suppose we are given a set $\mathbf{Ax}$ of axioms A, with proof terms p_A introduced by

$$\frac{\Gamma \Rightarrow A \ \mathsf{wff}}{\Gamma \Rightarrow p_A\!:\!A} \ A \in \mathbf{Ax}$$

(whence δ_A amounts to $p_{\neg\neg A \supset A}$). If $[\![A]\!]^{\mathcal{D}}_{\Gamma,\rho} \neq \emptyset$, say $a \in [\![A]\!]^{\mathcal{D}}_{\Gamma,\rho}$, then we can set $[\![p_A]\!]^{\mathcal{D}}_{\Gamma,\rho} = a$. Otherwise, we might set $[\![p_A]\!]^{\mathcal{D}}_{\Gamma,\rho}$ to some arbitrary value, and classify interpretations as good or bad according to $(*)$ above. That is, let us call an L-interpretation $[\![\cdot]\!]$ *sound for* $\mathbf{Ax}$ if for every $A \in \mathbf{Ax}$,

$$[\![p_A]\!]^{\mathcal{D}}_{\Gamma,\rho} \in [\![A]\!]^{\mathcal{D}}_{\Gamma,\rho}, \ \text{whenever '}\Gamma \Rightarrow^{\mathcal{D}} A \ \mathsf{wff}\text{' and } \rho \in [\![\Gamma]\!]^{\mathcal{D}}.$$

For a sound interpretation of double negation δ_A (i.e. $p_{\neg\neg A \supset A}$) relative to a $\mathcal{D}$-instantiation ρ of a $\mathcal{D}$-context Γ such that '$\Gamma \Rightarrow^{\mathcal{D}} A$ wff', it is useful to assume the law of excluded middle for $[\![A]\!]^{\mathcal{D}}_{\Gamma,\rho}$ — if $[\![A]\!]^{\mathcal{D}}_{\Gamma,\rho} \neq \emptyset$, say $a \in [\![A]\!]^{\mathcal{D}}_{\Gamma,\rho} \neq \emptyset$, then let $[\![\delta_A]\!]^{\mathcal{D}}_{\Gamma,\rho} = \lambda x.a$; otherwise, interpret δ_A exactly as the proof 0 of $\bot \supset A$.

Returning now to the question (Q2) from §1 of interpreting proofs model-theoretically, we can turn an L-model M into an L-interpretation $[\![\cdot]\!]_M$ by setting $[\![U]\!]_M = U_M$ and

$$[\![R, u_1 \cdots u_n]\!]_M = \begin{cases} \{0\} & \text{if } R_M(u_1,\cdots,u_n) \\ \emptyset & \text{otherwise.} \end{cases}$$

While 0 can hardly be described as a *canonical* proof object of every true atomic L-sentence, it will do as *a* proof object for our purposes. Indeed, for the sequel, it will be convenient to (i) quantify away proofs t, defining

$$\Gamma \models^{\mathcal{D}}_{[\![\cdot]\!]} A \quad \text{iff} \quad \text{'}\Gamma \Rightarrow^{\mathcal{D}} A \ \mathsf{type}\text{' and } (\forall \rho \in [\![\Gamma]\!]^{\mathcal{D}}) \ [\![A]\!]^{\mathcal{D}}_{\Gamma,\rho} \neq \emptyset \,,$$

and (ii) quantify away L-interpretations $[\![\cdot]\!]$,[2] defining

$$\Gamma \models^{\mathcal{D}} A \quad \text{iff} \quad \text{for every } \mathcal{D}\text{-sound } [\![\cdot]\!], \ \Gamma \models^{\mathcal{D}}_{[\![\cdot]\!]} A$$

[2] I drop L-interpretations $[\![\cdot]\!]$ from §3 on, in no small measure because the subscript $[\![\cdot]\!]$ on $\models^{\mathcal{D}}_{[\![\cdot]\!]}$ gets very cumbersome. Re-introducing $[\![\cdot]\!]$ would lead to different notions of presupposition and entailment in §§3 and 4 that, I suspect, could be of interest, but at the expense of notational clutter. Be that as it may, I would like to draw the reader's attention to Fernando [12], where L-interpretations $[\![\cdot]\!]$ are pushed further, for a link-up with DRT.

where $[\![\cdot]\!]$ is understood to be *$\mathcal{D}$-sound* for a rule set $\mathcal{D}$ induced by **Ax** if $[\![\cdot]\!]$ is sound for **Ax**. (**Ax** may be empty, or may contain some/all instances of double negation, as well as facts such as: canaries are birds.)

§3. Translations and presuppositions. Fix a signature L and a countable set Var of variables $(x, \dots)$. Decontextualizing the types and terms from §2, let us re-christen them *pre*types and *pre*terms, under context-free presentations. That is, let pTy be the set of *pretypes* A

$$A \ ::= \ U \mid \bot \mid R(t, \dots, t) \mid (\textstyle\sum x\!:\!A)A \mid (\prod x\!:\!A)A$$

generated from the set pTm of *preterms* t

$$t \ ::= \ x \mid lt \mid rt \mid \langle t, t\rangle \mid ap(t,t) \mid \lambda x.t \mid 0$$

(leaving δ_A/p_A out for simplicity). Throughout this section, let $\mathcal{D}$ be a fixed set of rules that includes all rules in §2.1 (and possibly more). Our goal is to describe how example (s) from §1 obtains the reading (1), assuming a formalization of the clause <u>the bird</u> will perish as the "extended pretype"

$$\{bird(z)\}\ willPerish(z) \tag{2}$$

where $bird(z)$ is a "restriction" on the variable z (meant in the spirit, if not letter, of situation theory; Barwise and Cooper [4], Cooper [7]). Reconceptualized as requirements on contexts Γ, restrictions are linked in §3.1 (below) to a theory of presuppositions due to Karttunen [19], with $\{\cdot\}\cdot$ construed as a binary presupposition connective (see, for instance, §4.1 of Beaver [5]). A mechanism is then provided in §3.2 that, for the example above, binds z in (2) to the preterm lx, relative to a $\mathcal{D}$-context $\Gamma, x\!:\!(\sum y\!:\!canary)\ SylvesterGets(y)$ such that

$$\Gamma \models^{\mathcal{D}} (\textstyle\prod y\!:\!canary)\ bird(y)$$

(recalling that, under the conventions mentioned in §2.1, the preceding pretype unwinds to $(\prod y\!:\!U)(canary(y) \supset\ bird(y))$).

3.1. Extended pretypes and full $\mathcal{D}$-presuppositions. Let us generate the set epTy of *extended pretypes* e by adding an inductive clause for restrictions to those giving pTy

$$e \ ::= \ U \mid \bot \mid R(t, \dots, t) \mid (\textstyle\sum x\!:\!e)e \mid (\prod x\!:\!e)e \mid \{e\}e.$$

Now, the idea is to extend the notion '$\Gamma \Rightarrow^{\mathcal{D}} A$ type' from pretypes A to extended pretypes e. Following Karttunen [19], let us define a $\mathcal{D}$-context Γ to *$\mathcal{D}$-admit* e — in symbols, $\Gamma \rhd^{\mathcal{D}} e$ — by induction on e

$$\begin{array}{rll}
\Gamma \rhd^{\mathcal{D}} A & \text{iff} & \text{`}\Gamma \Rightarrow^{\mathcal{D}} A\ \mathsf{type}\text{'} \quad \text{for } A = U, \bot, R(t_1, \dots, t_n) \\
\Gamma \rhd^{\mathcal{D}} (Qx\!:\!e)e' & \text{iff} & \Gamma \rhd^{\mathcal{D}} e \text{ and } \Gamma, x\!:\!e^- \rhd^{\mathcal{D}} e' \quad \text{for } Q = \sum, \prod \\
\Gamma \rhd^{\mathcal{D}} \{e\}e' & \text{iff} & \Gamma \models^{\mathcal{D}} e^\wedge \text{ and } \Gamma \rhd^{\mathcal{D}} e'
\end{array}$$

where e^- and $e^\wedge$ are pretypes given by functions $\cdot^-$ and $\cdot^\wedge$ from epTy to pTy defined as follows. For $A \in \mathsf{pTy}$, $A^- = A^\wedge = A$, and for $Q \in \{\sum, \prod\}$,

$$((Qx\!:\!e_1)e_2)^- = ((Qx\!:\!e_1{}^-)e_2{}^-) \qquad ((Qx\!:\!e_1)e_2)^\wedge = ((Qx\!:\!e_1{}^\wedge)e_2{}^\wedge)\ .$$

But while $\cdot^-$ erases restrictions, $\cdot^\wedge$ incorporates them

$$(\{e_1\}e_2)^- = e_2{}^- \qquad\qquad (\{e_1\}e_2)^\wedge = e_1{}^\wedge \wedge e_2{}^\wedge$$

where, as in §2.1, $A \wedge B$ abbreviates $(\sum x\!:\!A)B$ for some fresh/dummy variable x. Clearly, for every pretype A, $\Gamma \rhd^{\mathcal{D}} A$ iff '$\Gamma \Rightarrow^{\mathcal{D}} A$ type'. Moreover, equating $\neg e$ with $(\prod x\!:\!e)\bot$ for some x not in e,

$$\Gamma \rhd^{\mathcal{D}} \neg e \ \text{ iff } \ \Gamma \rhd^{\mathcal{D}} e\ .$$

Now, e is said to *$\mathcal{D}$-presuppose* A if $\Gamma \models^{\mathcal{D}} A$ for every Γ that $\mathcal{D}$-admits e.

Example. Assuming $\Gamma \models^{\mathcal{D}} (\prod y\!:\!\mathit{canary})\ \mathit{bird}(y)$, the $\mathcal{D}$-context

$$\Gamma, \quad x\!: (\sum y\!:\!\mathit{canary})\ \mathit{SylvesterGets}(y)$$

$\mathcal{D}$-admits the extended pretype

(3) $$\{\mathit{bird}(lx)\}\ \mathit{willPerish}(lx)$$

which, in turn, $\mathcal{D}$-presupposes $\mathit{bird}(lx)$. Dropping the restriction $\{\mathit{bird}(lx)\}$ from (3), notice that if a $\mathcal{D}$-context Γ $\mathcal{D}$-admits the pretype $\mathit{willPerish}(lx)$, then '$\Gamma \Rightarrow^{\mathcal{D}} lx\!:\!U$'.

Accordingly, let us say that e *$\mathcal{D}$-presupposes* $t\!:\!A$ if '$\Gamma \Rightarrow^{\mathcal{D}} t\!:\!A$' for every Γ such that $\Gamma \rhd^{\mathcal{D}} e$. A *full $\mathcal{D}$-presupposition of* an extended pretype e is a pair $\mathcal{A}, \mathcal{T}$ of finite sets $\mathcal{A}$ and $\mathcal{T}$ of pretypes A and typings $t\!:\!B$ respectively, such that for every $\mathcal{D}$-context Γ,

$$\Gamma \rhd^{\mathcal{D}} e \quad \text{iff} \quad (\forall A \in \mathcal{A})\ \Gamma \models^{\mathcal{D}} A \ \text{ and } \ (\forall \text{`}t\!:\!B\text{'} \in \mathcal{T})\ \text{`}\Gamma \Rightarrow^{\mathcal{D}} t\!:\!B\text{'}\ .$$

Let us write this in symbols as $e \downarrow \mathcal{A}, \mathcal{T}$.

Theorem. *Suppose that $\mathcal{D}$ contains not only the rules in* §2.1 *but also the introduction rule for $\prod$ in* §2.2*, along with the usual formation rules*

$$\frac{\Gamma \Rightarrow A\ \mathsf{type} \qquad \Gamma, x\!:\!A \Rightarrow B\ \mathsf{type}}{\Gamma \Rightarrow (\sum x\!:\!A)B\ \mathsf{type}} \qquad \frac{\Gamma \Rightarrow A\ \mathsf{type} \qquad \Gamma, x\!:\!A \Rightarrow B\ \mathsf{type}}{\Gamma \Rightarrow (\prod x\!:\!A)B\ \mathsf{type}}\ .$$

Then there is an algorithm that given $e \in \mathsf{epTy}$ returns a full $\mathcal{D}$-presupposition of e.

The algorithm is based on the calculations

$$e \ \downarrow\ \emptyset, \emptyset \quad \text{for } e = \bot \text{ or some } U \in L_0$$
$$R(t_1, \ldots, t_n) \ \downarrow\ \emptyset, \{t_1\!:\!U_1, \ldots, t_n\!:\!U_n\} \quad \text{for } R \in L(U_1 \cdots U_n)$$

and if $e \downarrow \mathcal{A}, \mathcal{T}$ and $e' \downarrow \mathcal{A}', \mathcal{T}'$ then both $(\sum x : e)e'$ and $(\prod x : e)e'$ have full $\mathcal{D}$-presupposition

$$\mathcal{A} \cup \{(\prod x\!:\!e^-)A' \mid A' \in \mathcal{A}'\}\ ,\ \mathcal{T} \cup \{\lambda x.t'\!:\!(\prod x\!:\!e^-)B' \mid \text{`}t'\!:\!B'\text{'} \in \mathcal{T}'\}$$

and

$$\{e\}e' \;\downarrow\; \{e^\wedge\} \cup \mathcal{A}', \mathcal{T}'.$$

Examples. The extended pretype (3) has full $\mathcal{D}$-presupposition $\{bird(lx)\}$, $\{lx\!:\!U\}$ which can, in fact, be reduced to $\{bird(lx)\}$, $\emptyset$ since

$$\text{whenever } \Gamma \models^{\mathcal{D}} bird(lx)\ ,\ \text{`}\Gamma \Rightarrow^{\mathcal{D}} lx\!:\!U\text{'}\ .$$

An extended pretype e and its negation $\neg e$ (i.e. $(\prod x\!:\!e)\bot$) have the same full $\mathcal{D}$-presuppositions.

3.2. Presupposition binding. Missing from our account of example (s) is how to go from (2) to (3), which we presently take up, turning $z \mapsto lx$ into a substitution. Given a function θ from a subset X of Var to pTm, define the function $\hat{\theta} : (\mathsf{pTm} \cup \mathsf{epTy}) \to (\mathsf{pTm} \cup \mathsf{epTy})$ substituting $x \in X$ by $\theta(x)$

$$\begin{array}{rclrcl}
\hat{\theta}(x) & = & \theta(x) \quad \text{for } x \in X & \hat{\theta}(ft) & = & f\hat{\theta}(t) \quad \text{for } f = l, r \\
\hat{\theta}(x) & = & x \quad \text{for } x \notin X & \hat{\theta}(ap(t,t')) & = & ap(\hat{\theta}(t), \hat{\theta}(t')) \\
\hat{\theta}(A) & = & A \quad \text{for } A = U, \bot & \hat{\theta}(\langle t,t'\rangle) & = & \langle \hat{\theta}(t), \hat{\theta}(t')\rangle \\
\hat{\theta}(R(t_1,\ldots,t_n)) & = & R(\hat{\theta}(t_1),\ldots,\hat{\theta}(t_n)) & \hat{\theta}(\{e\}e') & = & \{\hat{\theta}(e)\}\hat{\theta}(e')
\end{array}$$

except where the variable occurs bound

$$\begin{array}{rcl}
\hat{\theta}(\lambda x.t) & = & \lambda x.\hat{\theta_x}(t) \\
\hat{\theta}((\sum x\!:\!e)e') & = & (\sum x\!:\!\hat{\theta}(e))\hat{\theta_x}(e') \\
\hat{\theta}((\prod x\!:\!e)e') & = & (\prod x\!:\!\hat{\theta}(e))\hat{\theta_x}(e')
\end{array}$$

with $\theta_x = \theta - \{(x, \theta(x))\}$ (thereby confining the substitution to free occurrences). Next, fix an extended pretype e and a set $X \subset \mathsf{Var}$ of variables that intuitively are crying out to be bound. That is, let the pair (e, X) represent an "ambiguous" extended pretype which a function θ with domain X disambiguates as $\hat{\theta}(e)$. Finding a suitable θ such that a given context Γ $\mathcal{D}$-admits $\hat{\theta}(e)$ can be non-trivial. Part of the problem is $\models^{\mathcal{D}}$, which we might calculate by extending $\mathcal{D}$ to some $\mathcal{D}' \supseteq \mathcal{D}$ such that for every A,

$$\Gamma \models^{\mathcal{D}} A \ \text{ iff for some } t,\ \text{`}\Gamma \Rightarrow^{\mathcal{D}'} t\!:\!A\text{'}.$$

The search for $\mathcal{D}'$ and t can, however, be open-ended — far harder than the calculation of full presuppositions.

Notice that we have relativized condition (†) from §1 to $\mathcal{D}$,

$$\text{`}\Gamma \ \Rightarrow^{\mathcal{D}} \ A \ \mathsf{type}\text{'},$$

and broken that into two steps. First, calculate a full $\mathcal{D}$-presupposition of A; second, show that Γ satisfies that presupposition. The first step can be carried out without Γ; the second can do without A. The approach applies to extended pretypes e (not just A), on top of which some indeterminacy can be introduced through different bindings on a set X of variables. In reply then to question (Q1) from §1, we have distinguished two tractable tasks,

the calculation of full presuppositions, and checking $\Gamma \Rightarrow t\!:\!A$, from two other problems,

determining $\models^{\mathcal{D}}$, and resolving X through suitable bindings θ.

A handle on the indeterminacy from X is provided in the next section through a general framework for translating expressions such as (e, X) to logical forms such as pretypes, on which $\models^{\mathcal{D}}$ induces a notion of entailment.

§4. Non-determinism in translation and attunement. Translating English into a formal language is non-trivial insofar as the contextual character of ordinary discourse is taken into account. To capture discourse effects, let us apply a translation process to sequences of sentences, encoding the input/output behavior of the process by a set

$$St \subseteq \bigcup_{n \geq 0} (E^n \times \Phi^n)$$

of pairs $(\vec{\mathsf{e}}, \vec{\varphi})$ of sequences $\vec{\mathsf{e}} = \mathsf{e}_1 \cdots \mathsf{e}_n$ and $\vec{\varphi} = \varphi_1 \cdots \varphi_n$ from a set E of *expressions* e_i and a set Φ of *logical forms* φ_i such that the input $\vec{\mathsf{e}}$ may translate to (the output) $\vec{\varphi}$. The input/output relation St is *not* assumed functional, allowing an input $\vec{\mathsf{e}}$ to be ambiguous because there are distinct sequences $\vec{\varphi}$ and $\vec{\psi}$ such that both $(\vec{\mathsf{e}}, \vec{\varphi})$ and $(\vec{\mathsf{e}}, \vec{\psi})$ belong to St. Of course, the difference between $\vec{\varphi}$ and $\vec{\psi}$ may well be superficial — a measure of which is provided by what these sequences entail. That is, there is presumably some notion $\vdash \subseteq \Phi^* \times \Phi$ of entailment on Φ that tests just how different two outputs $\vec{\varphi}$ and $\vec{\psi}$ are. Indeed, we may well expect $\vdash$ to be the reason for mapping E to Φ.

Example (from §3, given a a fixed set $\mathcal{D}$ of rules). Let $E = \mathsf{epTy} \times Pow(\mathsf{Var})$, $\Phi = \mathsf{pTy}$, $\vdash^{\mathcal{D}} \subseteq \mathsf{pTy}^* \times \mathsf{pTy}$ be given by

$$A_1 \cdots A_n \vdash^{\mathcal{D}} A \text{ iff } x_1 : A_1, \ldots, x_n : A_n \models^{\mathcal{D}} A$$

and $St_{\mathcal{D}}$ consist, for every $n \geq 0$, of all pairs

$$((e_1, X_1)(e_2, X_2) \cdots (e_n, X_n)\ ,\ A_1 A_2 \cdots A_n) \in E^n \times \Phi^n$$

such that for some binding $\theta : (\bigcup_{i=1}^n X_i) \to \mathsf{pTm}$, whenever $1 \leq i \leq n$,

$$\mathsf{Var}(e_i) \cap dom(\theta) \subseteq X_i, \quad A_i \text{ is } \hat{\theta}(e_i)^- \text{ and } x_1 : A_1, \ldots, x_{i-1} : A_{i-1} \ \rhd^{\mathcal{D}}\ \hat{\theta}(e_i)$$

where $\mathsf{Var}(e_i)$ is the set of all variables with free occurrences in e_i. Notice that the same binding θ disambiguates each of $e_1, \ldots, e_n$.

4.1. The language $\mathcal{L}_\circ$. To lift $\vdash$-entailments of $\vec{\varphi}$ up to $\vec{\mathsf{e}}$ along the translation set St, it is useful to extend the set Φ of logical forms φ to a set of $\mathcal{L}_\circ(E, \Phi)$-*formulas* A generated according to

$$\mathsf{A} ::= \varphi \mid \langle \mathsf{e} \rangle \mathsf{A} \mid \tilde{\neg} \mathsf{A} \mid \mathsf{A} \tilde{\wedge} \mathsf{A}$$

with different modalities $\langle \mathsf{e} \rangle$ for each expression $\mathsf{e} \in E$, plus negation $\tilde{\neg}$ and conjunction $\tilde{\wedge}$ (not to be confused with any connectives in Φ or E). Let us shorten $\mathcal{L}_\circ(E, \Phi)$ to $\mathcal{L}_\circ$ and identify $\mathcal{L}_\circ$ with the set of $\mathcal{L}_\circ$-formulas. The semantics of $\mathcal{L}_\circ$ is specified relative to an $\mathcal{L}_\circ$-*model* $(St, \vdash)$ by a "forcing" relation $|\!\vdash_\circ \subseteq St \times \mathcal{L}_\circ$ as follows, where $(\vec{\mathsf{e}}, \vec{\varphi}) \in St$:

(i) a logical form φ is forced if $\vdash$-entailed by the output

$$(\vec{\mathsf{e}}, \vec{\varphi}) \;|\!\vdash_\circ \varphi \quad \text{iff} \quad \vec{\varphi} \vdash \varphi$$

(ii) $\langle \mathsf{e} \rangle \mathsf{A}$ is forced by $(\vec{\mathsf{e}}, \vec{\varphi})$ if some extension of $(\vec{\mathsf{e}}, \vec{\varphi})$ by e forces A

$$(\vec{\mathsf{e}}, \vec{\varphi}) \;|\!\vdash_\circ \langle \mathsf{e} \rangle \mathsf{A} \quad \text{iff} \quad \text{for some } \varphi,\ (\vec{\mathsf{e}}\mathsf{e}, \vec{\varphi}\varphi) \;|\!\vdash_\circ \mathsf{A}$$

(where it is understood that $(\vec{\mathsf{e}}\mathsf{e}, \vec{\varphi}\varphi)$ must also belong to St)

(iii) $\tilde{\neg}$ and $\tilde{\wedge}$ are treated classically

$$(\vec{\mathsf{e}}, \vec{\varphi}) \;|\!\vdash_\circ \tilde{\neg}\mathsf{A} \quad \text{iff} \quad \text{not } (\vec{\mathsf{e}}, \vec{\varphi}) \;|\!\vdash_\circ \mathsf{A}$$

$$(\vec{\mathsf{e}}, \vec{\varphi}) \;|\!\vdash_\circ \mathsf{A} \,\tilde{\wedge}\, \mathsf{B} \quad \text{iff} \quad (\vec{\mathsf{e}}, \vec{\varphi}) \;|\!\vdash_\circ \mathsf{A} \text{ and } (\vec{\mathsf{e}}, \vec{\varphi}) \;|\!\vdash_\circ \mathsf{B}\ .$$

Observe that $|\!\vdash_\circ$ is just Kripke semantics, where the accessibility relation $\xrightarrow{\mathsf{e}}$ underlying $\langle \mathsf{e} \rangle$ is the binary relation on St such that

$$(\vec{\mathsf{e}}, \vec{\varphi}) \xrightarrow{\mathsf{e}} (\vec{\mathsf{e}'}, \vec{\varphi'}) \quad \text{iff} \quad \vec{\mathsf{e}'} = \vec{\mathsf{e}}\mathsf{e} \text{ and } \vec{\varphi'} = \vec{\varphi}\varphi \text{ for some } \varphi$$

for all $(\vec{\mathsf{e}}, \vec{\varphi}), (\vec{\mathsf{e}'}, \vec{\varphi'}) \in St$. The persistence of the outputs $\vec{\varphi}$ under $\xrightarrow{\mathsf{e}}$ reflects a processing of the inputs which is

(i) *incremental*, translating a sequence $\mathsf{e}_1 \cdots \mathsf{e}_n$ one sentence at a time

$$(\epsilon, \epsilon) \xrightarrow{\mathsf{e}_1} (\mathsf{e}_1, \varphi_1) \xrightarrow{\mathsf{e}_2} \cdots \xrightarrow{\mathsf{e}_n} (\mathsf{e}_1 \cdots \mathsf{e}_n, \vec{\varphi})$$

(whence for all $(\vec{\mathsf{e}}, \vec{\varphi}) \in St$, the length of $\vec{\varphi}$ is the same as that of $\vec{\mathsf{e}}$)

and

(ii) *inertial* insofar as the incremental results are preserved: whenever $(\vec{\mathsf{e}}, \vec{\varphi}) \xrightarrow{\mathsf{e}} (\vec{\mathsf{e}'}, \vec{\psi})$, $\vec{\varphi}$ is a prefix of $\vec{\psi}$ (not to mention $\vec{\mathsf{e}}$ is a prefix of $\vec{\mathsf{e}'}$).

For a translation process that is incremental and inertial (as in DRT), we can regard St not only as its input/output relation but also as its set of *stages*. This perspective is developed in Fernando [11]. The additional point of the present work is twofold: to tie $\mathcal{L}_\circ$ in with §§2 and 3 above, and to expand $\mathcal{L}_\circ$ so as to bring in a third process underlying interpretation.

But first, a bit more background. Ambiguity in an expression e at a stage $(\vec{\mathsf{e}}, \vec{\varphi})$ is discerned in $\mathcal{L}_\circ$ by $\mathcal{L}_\circ$-formulas A, if any, such that

$$(\vec{\mathsf{e}}, \vec{\varphi}) \;|\!\vdash_\circ\; \langle \mathsf{e} \rangle \mathsf{A} \;\tilde{\wedge}\; \langle \mathsf{e} \rangle \tilde{\neg} \mathsf{A}\ .$$

Quantifying over all stages, let us call an $\mathcal{L}_\circ$-formula A *valid in* $(St, \vdash)$ if for every $(\vec{\mathsf{e}}, \vec{\varphi}) \in St$, $(\vec{\mathsf{e}}, \vec{\varphi}) \;|\!\vdash_\circ \mathsf{A}$. For example, suppose E includes the following English sentences e_1 and e_2.

e_1	Doris informed Chloe <u>she</u> aced the exam.
e_2	In fact, <u>she</u> was the only Texan who passed.

Ambiguous though e_1 may be, to say that the she's in $\mathsf{e}_1\mathsf{e}_2$ co-vary in the $\mathcal{L}_\circ$-model $(St, \vdash)$ is to require that the $\mathcal{L}_\circ$-formula

$$[\mathsf{e}_1][\mathsf{e}_2]\ (\forall x \in \mathit{Texan})(\mathit{passed}(x) \supset \mathit{aced}(x))$$

be valid in $(St, \vdash)$, where $[\mathsf{e}]\mathsf{A}$ abbreviates $\tilde{\neg}\langle\mathsf{e}\rangle\tilde{\neg}\,\mathsf{A}$.

Before worrying about multiple outputs, we might check if an input e at stage $(\vec{\mathsf{e}}, \vec{\varphi})$ has any output at all, which can be expressed as

$$(\vec{\mathsf{e}}, \vec{\varphi}) \;\; |\!\vdash_\circ \;\; \langle\mathsf{e}\rangle\tilde{\top} \qquad [\text{ iff } \;(\vec{\mathsf{e}}, \vec{\varphi}) \in \mathit{dom}(\xrightarrow{\mathsf{e}}) \;]$$

where $\tilde{\top}$ is an $\mathcal{L}_\circ$-tautology (valid in every $\mathcal{L}_\circ$-model). Underlying §3.1 is the identification of the presupposition of e with its precondition for translation, $\langle\mathsf{e}\rangle\tilde{\top}$. Under this conception, the *negation test* for presupposition, which states that the presupposition of the negation $\neg\mathsf{e}$ of an expression e is no different from the presupposition of e, becomes the $\mathcal{L}_\circ$-formula

$$\langle\neg\mathsf{e}\rangle\tilde{\top} \;\tilde{\equiv}\; \langle\mathsf{e}\rangle\tilde{\top}$$

(where $\tilde{\equiv}$ is the Boolean $\mathcal{L}_\circ$-biconditional defined from $\tilde{\neg}$ and $\tilde{\wedge}$), saying $\neg\mathsf{e}$ is translatable iff e is. As for a binary connective $\star$ on E (such as conjunction $\wedge$), let us agree to call $\star$ *St-sequential* if St translates the arguments of $\star$ sequentially — i.e., for all $(\vec{\mathsf{e}}, \vec{\varphi}) \in St$ and $\mathsf{e}, \mathsf{e}' \in E$,

$$\begin{aligned}(\vec{\mathsf{e}}(\mathsf{e}\star\mathsf{e}'), \vec{\varphi}\psi) \in St \;\text{ iff }\; & \psi = \varphi\cdot\varphi' \text{ for some } \varphi \text{ and } \varphi' \text{ such that} \\ & (\vec{\mathsf{e}}\mathsf{e}, \vec{\varphi}\varphi) \in St \text{ and } (\vec{\mathsf{e}}\mathsf{e}\mathsf{e}', \vec{\varphi}\varphi\varphi') \in St\end{aligned}$$

where $\cdot$ is some binary connective on Φ (possibly $\star$). If $\star$ is St-sequential then it immediately follows that the $\mathcal{L}_\circ$-scheme

$$\langle\mathsf{e}\star\mathsf{e}'\rangle\tilde{\top} \;\tilde{\equiv}\; \langle\mathsf{e}\rangle\langle\mathsf{e}'\rangle\tilde{\top}$$

is valid in $(St, \vdash)$, for any $\vdash$. The dependent type interpretations of conjunction $\wedge$ and implication $\supset$ in §§2 and 3 above are both $St_{\mathcal{D}}$-sequential (under the definition of $St_{\mathcal{D}}$ given right before this subsection). Moreover, the negation test, $\langle\neg\mathsf{e}\rangle\tilde{\top} \tilde{\equiv} \langle\mathsf{e}\rangle\tilde{\top}$, becomes a consequence of defining $\neg\mathsf{e}$ as $\mathsf{e}\supset\bot$, provided $\langle\bot\rangle\tilde{\top}$, as is the case with our dependent type example $St_{\mathcal{D}}$.[3]

Going well beyond $\langle\bot\rangle\tilde{\top}$, an assumption about St is made in Fernando [11], motivated by the example of translating English sentences (E) into ordinary predicate logic formulas (Φ). The assumption, called ($\mathrm{a}_\circ$), embeds the set Φ of logical forms in the set E of inputs to translation, with the understanding that St keeps the logical forms fixed, and contains the pair (ϵ, ϵ) — i.e.,

$$\begin{aligned}(\mathrm{a}_\circ) \qquad & \Phi \subset E, \;\; (\epsilon,\epsilon) \in St \;\text{ and for every } (\mathsf{e}_1\cdots\mathsf{e}_n, \varphi_1\cdots\varphi_n) \in St, \\ & (\forall i \in \{1,\ldots,n\}) \;\text{ if } \mathsf{e}_i \in \Phi \text{ then } \mathsf{e}_i = \varphi_i \\ & \qquad (\forall\varphi\in\Phi) \;\; (\mathsf{e}_1\cdots\mathsf{e}_n\varphi, \varphi_1\cdots\varphi_n\varphi) \in St\;.\end{aligned}$$

[3]To be more precise, we have, over $St_{\mathcal{D}}$, $\langle(\bot,\emptyset)\rangle\tilde{\top}$ rather than $\langle\bot\rangle\tilde{\top}$. We can, however, systematically confuse $(A,\emptyset)$ with A (as we see shortly). But we should not confuse presupposition satisfaction with consistency any more than translation with entailment

$$\frac{\text{translation}}{\text{presupposition satisfaction}} \approx \frac{\text{entailment}}{\text{consistency}}.$$

Inter-related though entailment and translation are, the processes are distinct, there being no shortage of English sentences that translate into what are, under ordinary notions of entailment, outright contradictions.

Under ($a_\circ$), the logical forms in Φ are extreme cases of unambiguous expressions with vacuous presuppositions. This assumption does not hold for the example of $St_{\mathcal{D}}$ above (with $\Phi = \mathsf{pTy}$), as an arbitrary context need not satisfy the presuppositions of an arbitrary pretype. Otherwise, however, notice that $(\epsilon, \epsilon) \in St_{\mathcal{D}}$ and that a copy of pTy in $E = \mathsf{epTy} \times Pow(\mathsf{Var})$ is given by mapping $A \in \mathsf{pTy}$ to $(A, \emptyset) \in \mathsf{epTy} \times Pow(\mathsf{Var})$. (The choice of $\emptyset$ ensures that $(A, \emptyset)$ can only be translated to A in $St_{\mathcal{D}}$.) Turning to $\vdash$, let us call $\vdash \subseteq \Phi^* \times \Phi$ $\mathcal{L}_\circ$*-transparent* if for all $\varphi_1, \ldots, \varphi_n, \varphi \in \Phi$,

$$\varphi_1 \cdots \varphi_n \vdash \varphi \quad \text{iff} \quad (\varphi_1 \tilde{\wedge} \cdots \tilde{\wedge} \varphi_n) \tilde{\supset} \varphi \text{ is } (St, \vdash)\text{-valid for any } St$$

(where $\tilde{\supset}$ is, like $\tilde{\equiv}$, a Boolean $\mathcal{L}_\circ$-connective defined from $\tilde{\neg}$ and $\tilde{\wedge}$).

Proposition. $\vdash$ *is* $\mathcal{L}_\circ$*-transparent iff* $\vdash$ *verifies the schemes*

$$\textit{(Weak)}\ \frac{\vec{\varphi} \vdash \varphi}{\psi\vec{\varphi} \vdash \varphi} \quad \frac{\vec{\varphi} \vdash \varphi}{\vec{\varphi}\psi \vdash \varphi} \qquad \textit{(Ref)}\ \frac{}{\varphi \vdash \varphi} \qquad \textit{(Cut)}\ \frac{\vec{\varphi} \vdash \psi \qquad \psi\vec{\varphi} \vdash \varphi}{\vec{\varphi} \vdash \varphi} \ .$$

Just as ($a_\circ$) does not quite hold for $St_{\mathcal{D}}$, neither does $\mathcal{L}_\circ$-transparency for $\vdash^{\mathcal{D}}$. The assumptions are, however, met by a special fragment of pTy, including all first-order L-formulas, and are of interest inasmuch as that fragment is. So continuing with our review of Fernando [11], let us add to $\mathcal{L}_\circ$-transparency the assumption that Φ has conjunction and implication (distinct from $\mathcal{L}_\circ$-connectives), recording these in

(a$\vdash$) (Weak), (Ref) and (Cut) hold in $\vdash$, and there are binary connectives $\wedge$ and $\supset$ in Φ such that for all $\varphi, \psi, \varphi' \in \Phi$ and $\vec{\varphi} \in \Phi^*$,

$$(\varphi \wedge \psi)\vec{\varphi} \vdash \varphi' \ \text{ iff } \ \varphi\psi\vec{\varphi} \vdash \varphi'$$
$$\vec{\varphi}\varphi \vdash \psi \ \text{ iff } \ \vec{\varphi} \vdash \varphi \supset \psi \ .$$

Using $\wedge$ to combine any finite number of logical forms entailed at a certain stage into one, and $\supset$ to anticipate logical forms entailed at a next stage, we can establish

Theorem. *The set of* $\mathcal{L}_\circ$*-formulas valid in all* $\mathcal{L}_\circ$*-models* $(St, \vdash)$ *satisfying* ($a_\circ$) *and* (a$\vdash$) *is decidable, and consists of the theorems of normal modal logic, supplemented with the* $\mathcal{L}_\circ$*-axiom schemes*

(A1)	$\varphi \supset \varphi$	*(A4)*	$\varphi \tilde{\supset} [\mathsf{e}]\varphi$
(A2)	$\varphi \wedge \psi \ \tilde{\equiv} \ \varphi \tilde{\wedge} \psi$	*(A5)*	$\langle\varphi\rangle\psi \ \tilde{\equiv} \ \varphi \supset \psi$
(A3)	$(\varphi \supset \psi) \ \tilde{\supset} \ (\varphi \tilde{\supset} \psi)$	*(A6)*	$\langle\varphi\rangle\mathsf{A} \ \tilde{\supset} \ [\varphi]\mathsf{A}$

Moreover, for a fixed $\vdash$ *satisfying* (a$\vdash$)*, it suffices to strenghten (A1)-(A3) to the* $\mathcal{L}_\circ$*-scheme*

$$\varphi_1 \tilde{\wedge} \cdots \tilde{\wedge} \varphi_n \ \tilde{\supset} \ \varphi$$

for all $\varphi_1, \ldots, \varphi_n, \varphi$ *such that* $\varphi_1 \cdots \varphi_n \vdash \varphi$.

The obvious question now is how to modify the axioms in the theorem above, when stepping beyond elementary predicate logic and into the rest of pTy.

4.2. Presuppositions and leaps between $\mathcal{L}_\circ$-models. The preceding analysis of presupposition $\langle \mathsf{e} \rangle \tilde{\top}$ in terms of *St*-sequentiality may have left the impression that presupposition is all a matter of translation *St*. This is misleading insofar as it suggests that entailment $\vdash$ is irrelevant to presupposition — which is hardly the case with our dependent type example; $St_{\mathcal{D}}$ is tightly coupled to $\vdash^{\mathcal{D}}$. More concretely, returning to example (s), let Γ_1 be the $\mathcal{D}$-context

$$x_1 \ : \ (\sum y\!: canary)\ \mathit{SylvesterGets}(y)$$

and abstracting out ambiguity for the time being, let us translate

$$\underline{\textsf{the bird}}\ \textsf{will perish}$$

as $(\hat{e}, \emptyset)$ where $\hat{e}$ is the extended pretype $\{\mathit{bird}(lx_1)\}\ \mathit{willPerish}(lx_1)$. Assuming (as we have) that *bird*, *willPerish*$\in L(U)$ for the same $U \in L_0$, it follows that

$$\langle (\hat{e}, \emptyset) \rangle \tilde{\top} \ \ \tilde{\equiv} \ \ \mathit{bird}(lx_1)$$

is $(St_{\mathcal{D}}, \vdash^{\mathcal{D}})$-valid, and that if $\models^{\mathcal{D}} (\prod y\!: canary)\mathit{bird}(y)$, Γ_1 $\mathcal{D}$-admits $\hat{e}$. Observe that dropping the restriction $\{\mathit{bird}(lx_1)\}$ on $\hat{e}$ complicates matters, as reducing the presupposition $\langle (\mathit{willPerish}(lx_1), \emptyset) \rangle \tilde{\top}$ to some pretype is problematic. This difficulty is not surprising, in view of well-known limitations on internal definability in explicit mathematics and in Frege structures (pp 41,42, Aczel [1]). Put positively, imposing suitable restrictions on an extended pretype e allows us to assume that e has a full $\mathcal{D}$-presupposition $\mathcal{A}, \emptyset$ where $\mathcal{A}$ consists of a single pretype A and where moreover we might arrange '$\Gamma \Rightarrow^{\mathcal{D}} A$ wff' for every $\mathcal{D}$-context Γ that $\mathcal{D}$-admits e. Re-introducing ambiguity, let us replace $(\hat{e}, \emptyset)$ by $(e', \{z\})$ where e' is the extended pretype $\{\mathit{bird}(z)\}$ $\mathit{willPerish}(z)$. In this case, our reduction of presupposition stops at

$$\langle (e', \{z\}) \rangle \tilde{\top} \ \ \tilde{\equiv} \ \ \langle (\{\mathit{bird}(z)\}\bot, \{z\}) \rangle \ \tilde{\top} \ .$$

Identifying the presupposition of an expression e with $\langle \mathsf{e} \rangle \tilde{\top}$ raises the question: what if $\langle \mathsf{e} \rangle \tilde{\top}$ fails? Suppose, for instance, (s) were altered ever so slightly to

(s') If Sylvester gets holds of a canary, $\underline{\textsf{the poor bird}}$ will perish.

Intuitively (s') is as interpretable as (s). But whereas the rule set $\mathcal{D}$ chosen at the outset may well contain the information that canaries are birds, it is another matter to anticipate (s') by arranging that

$$\epsilon \ \models^{\mathcal{D}} \ (\prod y\!: canary)\ (\mathit{SylvesterGets}(y) \supset \mathit{poorBird}(y)) \ . \tag{4}$$

If (4) fails, then we have presupposition failure, and interpretation of (s') requires leaping out of the $\mathcal{L}_\circ$-model $(St_{\mathcal{D}}, \vdash^{\mathcal{D}})$ and into some other $\mathcal{L}_\circ$-model where the translation of (s') can proceed. Such leaps, repairing defective contexts, are called *accommodation* in the presupposition literature (surveyed in Beaver [5]). They are not to be undertaken mindlessly. Consider

If Sylvester gets holds of a canary, $\underline{\textsf{Sylvester's son}}$ will get dinner.

Assuming we were willing to accommodate, just what assumption should we take on board: that Sylvester has a son, *Shas*, or that he does if he captures a canary

$$(\prod y : canary)\ (SylvesterGets(y) \supset Shas)\ ?$$

When and how to accommodate are notoriously tricky questions involving judgments of plausibility that may or may not be amenable to formalization. In particular, while the presuppositions calculated in §3.1 may be useful for flagging presupposition failure, they need not themselves be the appropriate assumptions to incorporate into defective contexts. And once recovery from some cases of presupposition failure is permitted, it is far from clear what intuitions about presuppositions to test. For instance, is interpreting (s') really a matter of accommodation or not? Our somewhat slippery reply above is: that depends on whether or not interpretation begins at a rule set $\mathcal{D}$ verifying (4). The skeptical reader may inquire instead if

$$(\prod y : canary)\ (SylvesterGets(y) \supset poorBird(y))$$

is part of the presupposition, assertion or implicature of (s')? The term presupposition has been applied as a label on so many things that perhaps any account of it is bound to be contentious.[4] What is at stake in the present work is the claim that failures in translation are interesting, as are the adjustments (to $\mathcal{L}_\circ$-models) made before, during or after a particular instance of translation.

[4]A prima facie different approach pursued in (for instance) Gazdar [14] is to cancel presuppositions (initially adopted in toto, unaltered from the sentence's simple constituents) which conflict with certain implicatures derived from Grice's maxim of quantity. (As this cancellation is to be carried out after logical forms have been constructed, we may expect that some stages entail the contradiction $\bot$ — as exemplified by the stage $(\bot, \bot)$ that $(\mathrm{a}_\circ)$ provides, embodying the saying: "garbage in, garbage out." But we can, in $\mathcal{L}_\circ$, discriminate between stages that force $\bot$ and those that force $\tilde{\neg}\,\bot$, leaping between $\mathcal{L}_\circ$-models on that basis.)

Instead of cancellation, the approach due to Karttunen [19] which we have been considering above takes a sentence "to be an increment to a context that satisfies its presuppositions" (page 191). It is not altogether evident, however, that increments are any simpler than decrements, as accommodation complicates the sense in which increments might be described as minimal. The effect of accommodation is to satisfy presuppositions on the basis not so much of entailment but of *compatibility*, insofar as any $\mathcal{L}_\circ$-model on which we land after a leap must (in some plausible sense) be compatible with the old $\mathcal{L}_\circ$-model (from which we take off). Compatibility here may explain why there is nothing at all odd about (s') relative to (s). (That said, we may prefer to satisfy presuppositions without changing $\mathcal{L}_\circ$-models, resorting to accommodation only if we have to, and only if we could — difficult it may be to spell out a general rule stating when and how we could.) A wider ranging account of the coherence of (s') might look at the interplay between (hearer) interpretation and (speaker) generation, and, with that, intentions (purposes) and planning. (How to bring all this within the present framework I can only hint at, in §5.1 below.)

Whatever picture of presupposition one adopts, there are many reasons (including Gricean implicatures) for considering background information α alongside the explicit discourse context $(\vec{\mathsf{e}}, \vec{\varphi})$. The generality of logical forms (e.g. McCarthy and Buvač [23]) is brought out by keeping α separate from the translation $\vec{\varphi}$ of an input $\vec{\mathsf{e}}$ (rather than packing it into $\overrightarrow{\varphi^{\alpha}}$) and then applying $\vec{\varphi}$ to different α's. The process of choosing α I propose to label *attunement*, and relate to the two other processes of entailment and translation by stipulating that α determines an $\mathcal{L}_\circ$-model $(St_\alpha, \vdash_\alpha)$. Collecting α's in a set I that indexes a family $\{(St_\alpha, \vdash_\alpha)\}_{\alpha \in I}$ of $\mathcal{L}_\circ$-models, an expansion $\mathcal{L}$ of $\mathcal{L}_\circ$ is outlined in the next section, relative to which such a family of $\mathcal{L}_\circ$-models constitutes a model. Without settling what are legitimate cases of accommodation or implicature, we can nonetheless survey the possibilities in $\mathcal{L}$, passing from $\mathcal{L}_\circ$-stages $(\vec{\mathsf{e}}, \vec{\varphi})$ to $\mathcal{L}$-stages $(\vec{\mathsf{e}}, \vec{\varphi})_\alpha$, where α keeps track of the $\mathcal{L}_\circ$-model to which $(\vec{\mathsf{e}}, \vec{\varphi})$ is understood to belong. The set I of indices opens up a dimension in interpretation, beyond translation and entailment, both of which take on a monotonic character in §4.1. The expansion from $\mathcal{L}_\circ$ to $\mathcal{L}$ allows non-monotonicity to enter through leaps $(\vec{\mathsf{e}}, \vec{\varphi})_\alpha \xrightarrow{-} (\vec{\mathsf{e}}, \vec{\varphi})_\beta$ that change α — although, in fact, already in $\mathcal{L}_\circ$, transitions $(\vec{\mathsf{e}}, \vec{\varphi}) \xrightarrow{\diamond} (\vec{\mathsf{e}}, \vec{\psi})$ exploiting ambiguity within a fixed $\mathcal{L}_\circ$-model provide a measure of non-monotonicity. Non-monotonicity aside, the mickey-mouse impression given by translations such as *SylvesterGets*(y) points to the possibility of refining the analysis, reifying Sylvester, and beyond that, the possible events of Sylvester getting hold of a canary. Such refinements amount to cases of re-attunement, if we encode into α not only the rule set $\mathcal{D}$ but also the signature L underlying $\mathcal{D}$, allowing the notion Φ of logical form to vary with α. Having embedded Φ in E under (a$_\circ$), it is only fair to obliterate restrictions keeping expressions in E from being returned as outputs.

4.3. Reifying $\mathcal{L}_\circ$-contexts and explicit translations. Blurring the line between E and Φ, let us define an *$\mathcal{L}$-model* to be a pair $(ST, \{\vdash_\alpha\}_{\alpha \in I})$ such that $ST \subseteq \bigcup_{n \geq 0}(E^n \times E^n \times I)$ and for each $\alpha \in I$, $\vdash_\alpha \subseteq E^* \times E$. For $\alpha \in I$, let us set

$$ST_\alpha = \{(\vec{\mathsf{e}}, \vec{\varphi}) \mid (\vec{\mathsf{e}}, \vec{\varphi})_\alpha \in ST\} ,$$

writing $(\vec{\mathsf{e}}, \vec{\varphi})_\alpha$ for $(\vec{\mathsf{e}}, \vec{\varphi}, \alpha)$, and using $\vec{\varphi}$ and φ for expressions occurring as translation outputs. $\mathcal{L}$-formulas include all $\mathcal{L}_\circ(E,E)$-formulas, and are interpreted relative to an $\mathcal{L}$-model $(ST, \{\vdash_\alpha\}_{\alpha \in I})$ as in $\mathcal{L}_\circ$

$$\begin{array}{rcl}
(\vec{\mathsf{e}}, \vec{\varphi})_\alpha \mid\!\vdash \varphi & \text{iff} & \vec{\varphi} \vdash_\alpha \varphi \\
(\vec{\mathsf{e}}, \vec{\varphi})_\alpha \mid\!\vdash \langle \mathsf{e} \rangle \mathsf{A} & \text{iff} & (\vec{\mathsf{e}}\mathsf{e}, \vec{\varphi}\varphi)_\alpha \mid\!\vdash \mathsf{A} \text{ for some } \varphi \\
(\vec{\mathsf{e}}, \vec{\varphi})_\alpha \mid\!\vdash \tilde{\neg} \mathsf{A} & \text{iff} & \text{not } (\vec{\mathsf{e}}, \vec{\varphi})_\alpha \mid\!\vdash \mathsf{A} \\
(\vec{\mathsf{e}}, \vec{\varphi})_\alpha \mid\!\vdash \mathsf{A} \mathbin{\tilde{\wedge}} \mathsf{B} & \text{iff} & (\vec{\mathsf{e}}, \vec{\varphi})_\alpha \mid\!\vdash \mathsf{A} \text{ and } (\vec{\mathsf{e}}, \vec{\varphi})_\alpha \mid\!\vdash \mathsf{B}
\end{array}$$

for all $(\vec{\mathsf{e}}, \vec{\varphi})_\alpha \in ST$ and $\varphi, \mathsf{e} \in E$. It follows that for every $\alpha \in I$ and all $\mathcal{L}_\circ(E,E)$-formulas A, $(\vec{\mathsf{e}}, \vec{\varphi})_\alpha \mid\!\vdash \mathsf{A}$ iff $(\vec{\mathsf{e}}, \vec{\varphi}) \mid\!\vdash_\circ \mathsf{A}$ relative to $(ST_\alpha, \vdash_\alpha)$.

Stepping outside of $\mathcal{L}_\circ$, the point in expanding $\mathcal{L}_\circ$ to $\mathcal{L}$ is to allow changes in $\alpha \in I$, the simplest of which is $\overset{-}{\rightarrow} \subseteq ST \times ST$ defined by

$$(\vec{\mathsf{e}}, \vec{\varphi})_\alpha \overset{-}{\rightarrow} (\vec{\mathsf{e}'}, \vec{\varphi'})_{\alpha'} \quad \text{iff} \quad \vec{\mathsf{e}'} = \vec{\mathsf{e}} \ \text{ and } \ \vec{\varphi'} = \vec{\varphi} \ ,$$

keeping $\vec{e}$ and $\vec{\varphi}$ fixed, with modal operator $\langle - \rangle$

$$(\vec{\mathsf{e}}, \vec{\varphi})_\alpha \Vdash \langle - \rangle \mathsf{A} \quad \text{iff} \quad (\vec{\mathsf{e}}, \vec{\varphi})_\beta \Vdash \mathsf{A} \text{ for some } \beta \ .$$

For example, that "e's presuppositions fail at $(\vec{\mathsf{e}}, \vec{\varphi})_\alpha$ but can be accommodated consistently" can be expressed as

$$(\vec{\mathsf{e}}, \vec{\varphi})_\alpha \ \Vdash \ \tilde{\neg} \langle \mathsf{e} \rangle \tilde{\top} \ \tilde{\wedge} \ \langle - \rangle (\tilde{\neg} \bot \,\tilde{\wedge}\, \langle \mathsf{e} \rangle \tilde{\top}) \ .$$

This leaves the previous translations $\vec{\varphi}$ intact. Varying $\vec{\varphi}$ rather than α,

$$(\vec{\mathsf{e}}, \vec{\varphi})_\alpha \Vdash \Diamond \mathsf{A} \quad \text{iff} \quad (\vec{\mathsf{e}}, \vec{\psi})_\alpha \Vdash \mathsf{A} \text{ for some } \vec{\psi} \ ,$$

with accessibility relation $\overset{\Diamond}{\rightarrow} \subseteq \ ST \times ST$ confining the revisions to $\vec{\varphi}$

$$(\vec{\mathsf{e}}, \vec{\varphi})_\alpha \ \overset{\Diamond}{\rightarrow} \ (\vec{\mathsf{e}'}, \vec{\varphi'})_{\alpha'} \quad \text{iff} \quad \vec{\mathsf{e}} = \vec{\mathsf{e}'} \text{ and } \alpha = \alpha' \ .$$

$\langle - \rangle$ can be distinguished from $\Diamond$ by introducing for every $\varphi \in E$ the atomic $\mathcal{L}$-formula $\underline{\varphi}$ saying φ is explicit:

$$(\vec{\mathsf{e}}, \vec{\varphi})_\alpha \Vdash \underline{\varphi} \quad \text{iff} \quad \varphi \text{ is listed among } \vec{\varphi} \ .$$

(In fact, $\underline{\varphi}$ can be formulated in terms of a $\overset{-}{\rightarrow}$-leap into an $\mathcal{L}_\circ$-model where entailment amounts to look-up in the sense given by the preceding clause; but let us keep matters simple and stick to $\underline{\varphi}$.) Now, beyond the $\mathcal{L}$-schemes

$$\begin{array}{l} \underline{\varphi} \ \tilde{\supset} \ [\mathsf{e}]\underline{\varphi} \\ \tilde{\neg}\, \underline{\varphi} \,\tilde{\wedge}\, \tilde{\neg}\, \underline{\psi} \ \tilde{\supset} \ [\mathsf{e}] \tilde{\neg} \,(\underline{\varphi} \,\tilde{\wedge}\, \underline{\psi}) \quad \text{for distinct } \varphi \text{ and } \psi \ , \end{array}$$

we can distinguish $\langle - \rangle$ from $\Diamond$ through

$$\underline{\varphi} \,\tilde{\supset}\, [-] \underline{\varphi} \qquad\qquad\qquad \tilde{\neg}\, \underline{\varphi} \,\tilde{\supset}\, [-] \tilde{\neg}\, \underline{\varphi}$$

(where $[-]$ is $\tilde{\neg} \langle - \rangle \tilde{\neg}$), saying previous outputs are untouched by $\overset{-}{\rightarrow}$. With $\mathcal{L}$-formulas $\underline{\varphi}$, we can use

$$\langle e \rangle \underline{e} \ \tilde{\wedge} \ [e] \underline{e} \tag{5}$$

to pick out expressions e from E that approximate the logical forms in an $\mathcal{L}_\circ$-model satisfying ($\mathrm{a}_\circ$). Let us henceforth write $\mathsf{LF}(e)$ to abbreviate (5). If $\wedge$ and $\supset$ are sequential, then

$$\begin{array}{rcl} \mathsf{LF}(\varphi) \,\tilde{\wedge}\, \langle \varphi \rangle \mathsf{LF}(\psi) & \tilde{\equiv} & \mathsf{LF}(\varphi \wedge \psi) \\ \mathsf{LF}(\varphi) \,\tilde{\wedge}\, \langle \varphi \rangle \mathsf{LF}(\psi) & \tilde{\equiv} & \mathsf{LF}(\varphi \supset \psi). \end{array}$$

For the remainder of this section, let $E = \mathsf{epTy} \times Pow(\mathsf{Var})$, I be some set of rule sets $\mathcal{D}$, and, construing the pair $(A, \emptyset) \in E$ as a copy of the pretype A,

$$\begin{array}{rcl} ST_{\mathcal{D}} & = & \{(\vec{\mathsf{e}}, (A_1, \emptyset) \cdots (A_n, \emptyset)) \mid (\vec{\mathsf{e}}, A_1 \cdots A_n) \in St_{\mathcal{D}}\} \\ \vdash_{\mathcal{D}} & = & \{(\vec{\mathsf{e}}, (A, \emptyset)) \mid (\forall (\vec{\mathsf{e}}, \vec{A}) \in St_{\mathcal{D}}) \ \vec{A} \vdash^{\mathcal{D}} A\} \ . \end{array}$$

The $\mathcal{L}_\circ$-schemes (A1)-(A6) from the end of §4.1 above relativize straightforwardly to the $(ST, \{\vdash_\mathcal{D}\}_{\mathcal{D}\in I})$-valid $\mathcal{L}$-schemes

$$\begin{array}{rcl@{\qquad}rcl}
\mathsf{LF}(\varphi) & \tilde{\supset} & \varphi \supset \varphi & \varphi & \tilde{\supset} & [\mathsf{e}]\varphi \\
\mathsf{LF}(\varphi) \,\tilde{\wedge}\, \mathsf{LF}(\psi) & \tilde{\supset} & \varphi \,\tilde{\wedge}\, \psi \;\tilde{\equiv}\; \varphi \wedge \psi & \mathsf{LF}(\varphi) \,\tilde{\wedge}\, \langle\varphi\rangle \mathsf{LF}(\psi) & \tilde{\supset} & \langle\varphi\rangle\psi \;\tilde{\equiv}\, \varphi \supset \psi \\
\mathsf{LF}(\psi) \,\tilde{\wedge}\, (\varphi \supset \psi) & \tilde{\supset} & \varphi \,\tilde{\supset}\, \psi & \mathsf{LF}(\varphi) \,\tilde{\wedge}\, \langle\varphi\rangle \mathsf{A} & \tilde{\supset} & [\varphi]\mathsf{A} \ .
\end{array}$$

Additional $(ST, \{\vdash_\mathcal{D}\}_{\mathcal{D}\in I})$-valid $\mathcal{L}$-schemes include $\underline{\varphi} \,\tilde{\supset}\, \varphi$,

$$\mathsf{LF}(\varphi) \;\tilde{\supset}\; [\mathsf{e}]\mathsf{LF}(\varphi)$$

$$\tilde{\neg}\, \underline{\varphi} \,\tilde{\wedge}\, \langle\mathsf{e}\rangle(\underline{\varphi} \,\tilde{\wedge}\, \mathsf{LF}(\psi) \,\tilde{\wedge}\, \psi) \;\tilde{\supset}\; (\varphi \supset \psi)$$

but, given presuppositional pretypes, we should resist the $\mathcal{L}$-scheme

$$\langle\mathsf{e}\rangle\mathsf{LF}(\varphi) \;\tilde{\supset}\; \mathsf{LF}(\varphi) \ .$$

Then there are the S5 axioms for $\Diamond$ and $\langle - \rangle$, and how the different modalities interact. I hope to report on these matters elsewhere.

§5. Conclusion.

First, a quick summary. With an eye to interpreting a set E of expressions, we formalize in §2 an elementary system of dependent types around a signature L and rule set $\mathcal{D}$. In §3, we decontextualize the types (passing from type to pTy), introduce restrictions along with presuppositions (extending pTy to epTy), and admit ambiguity (multiplying epTy by $Pow(\mathsf{Var})$). These notions are analyzed in §4 through a language $\mathcal{L}_\circ$ that integrates translation, St, and entailment, $\vdash$. $\mathcal{L}_\circ$ is expanded to a language $\mathcal{L}$ for multiple translation-entailment pairs $\{(St_\alpha, \vdash_\alpha)\}_{\alpha\in I}$, carving context up between an explicit (or external) part (in $\mathcal{L}_\circ$), and an *i*mplicit (or *i*nternal) part I that incorporates the choice α of a signature L and rule set $\mathcal{D}$. $\mathcal{L}$ provides a handle on the non-determinism in natural language interpretation abstracted away by the homomorphisms invoked for "English as a formal language."

5.1. From types to processes.

$\mathcal{L}$ sharpens the type-theoretic claim (†) formulated in the introduction to

(‡) whenever $(\mathsf{e}_1 \cdots \mathsf{e}_n \mathsf{e}, \varphi_1 \cdots \varphi_n \varphi) \in ST_\alpha$,

$$x_1 : \varphi_1, \ \ldots \ , \ x_n : \varphi_n \ \Rightarrow_\alpha \ \varphi \ \mathsf{wff} \ .$$

(‡) recognizes a place for English $\vec{e}$ and for an indexical element α determining a notion of inference $\Rightarrow_\alpha$ that conditions not only a process ST_α of translation from e's to φ's, but also the entailments $\vdash_\alpha$ between φ's. Although not explicitly linked above to indexicals and deixis, the indices α might be characterized as particular settings of deictic parameters (building on, for example, Kamp [18] and Cooper [7] for a finer analysis of attunement).

5.2. A multi-faceted approach.

Rather than reducing natural language interpretation to proof theory, the present work distinguishes entailment from two other processes: translation and attunement. While the formal distinctions between the processes drawn above are perhaps crisp enough, there are intuitive differences that are somewhat harder to pin down — for instance, the differing computational pressures on the processes. Translation has a computational urgency that the open-ended process of entailment does not, as hinted in §1 by (Q1). Although translations may depend on entailments, the entailments considered after logical forms have been translated may, in practice, be expected to be more complex than those involved in translating them. Otherwise, the price of decoding the English discourse is hardly worth the bit of information buried in it. An obvious requirement on "computationally significant" notions $St, \vdash$ of translation and entailment over a collection E of expressions is that St be decidable and $\vdash$ be semi-decidable. Or within the program of Feferman [8], we might fix some class $\mathcal{I}$ of indices and two operations $\hat{t}, \hat{p}$ (of translation and proof) such that

$$\begin{aligned} \hat{t} &: (\mathcal{I} \times \bigcup_{n \geq 0} E^n \times E^n) \rightarrow \{0,1\} \\ \hat{p} &: (\mathcal{I} \times D \times E^* \times E) \rightarrow \{0,1\} \end{aligned}$$

for some class D of candidate derivations. An index $\alpha \in \mathcal{I}$ would then determine the $\mathcal{L}_\circ$-model $(ST_\alpha, \vdash_\alpha)$ given by

$$\begin{aligned} ST_\alpha &= \bigcup_{n \geq 0} \{\langle \vec{e}, \vec{\varphi} \rangle \in E^n \times E^n \mid ap(\hat{t}, \langle \alpha, \vec{e}, \vec{\varphi} \rangle) \simeq 0\} \\ \vdash_\alpha &= \{\langle \vec{e}, e \rangle \in E^* \times E \mid (\exists d \in D)\ ap(\hat{p}, \langle \alpha, d, \vec{e}, e \rangle) \simeq 0\} . \end{aligned}$$

Failure to translate an input $\vec{e}$ may arise from incompleteness in the entailment relation $\vdash_\alpha$, inviting a leap into a different index β, as in Turing's ordinal logics (Feferman [9]). How the indices α and β are chosen is the deep mystery of attunement, which we might explore within an expansion $\mathcal{L}$ of $\mathcal{L}_\circ$. It is a considerable stretch from $\mathcal{I}$ to Church-Kleene ordinal notations $\mathcal{O}$, but just as Feferman has asked "what is a natural well-ordering?", we may ask "what is a natural subclass $I \subseteq \mathcal{I}$ for $\mathcal{L}$?" The latter question may not belong to logic (let alone proof theory), but a sense for logic (and formal systems) must surely throw some light on what it could mean.

REFERENCES

[1] Peter Aczel, *Frege structures and the notions of proposition, truth and set*, ***The Kleene symposium*** (J. Barwise, H.J. Keisler, and K. Kunen, editors), North-Holland, Amsterdam, 1980.

[2] R. Ahn and H-P. Kolb, *Discourse representation meets constructive mathematics*, ***Papers from the second symposium on logic and language*** (L. Kálmán and L. Pólos, editors), Akadémiai Kiadó, Budapest, 1990.

[3] N. Asher and A. Lascarides, *Bridging*, ***Journal of Semantics***, vol. 15 (1998), pp. 83–113.

[4] J. Barwise and R. Cooper, *Extended Kamp notation: a graphical notation for situation theory*, ***Situation theory and its applications*** (P. Aczel, D. Israel, Y. Katagiri, and S. Peters, editors), vol. 3, CSLI Lecture Notes Number 37, Stanford, 1993.

[5] David Ian Beaver, *Presupposition*, ***Handbook of logic and language*** (J. van Benthem and A. ter Meulen, editors), Elsevier, 1997.

[6] Tijn Borghuis, *Modal pure type systems*, ***Journal of Logic, Language and Information***, vol. 7 (1998), pp. 265–296.

[7] Robin Cooper, *Mixing situation theory and type theory to formalize information states in dialogue exchanges*, ***Formal semantics and pragmatics of dialogue***, Twente, the Netherlands, 1998.

[8] Solomon Feferman, *A language and axioms for explicit mathematics*, ***Algebra and logic*** (J.N. Crossley, editor), LNM 450, Springer-Verlag, Berlin, 1975.

[9] ———, *Turing in the land of 0(z)*, ***The universal Turing machine*** (R. Herken, editor), Oxford University Press, Oxford, 1988.

[10] ———, *Polymorphic typed lambda-calculi in a type-free axiomatic framework*, ***Logic and computation***, Contemporary Mathematics, vol. 106, A.M.S., Providence, 1990.

[11] Tim Fernando, *A modal logic for non-deterministic discourse processing*, ***Journal of Logic, Language and Information***, vol. 8 (1999), pp. 445–468, **Corrigendum**: the axiom scheme $(\varphi \supset \psi) \equiv (\varphi > \psi)$ in §6 (p.465) should be weakened to $(\varphi > \psi) \supset (\varphi \supset \psi)$.

[12] ———, *A type reduction from proof-conditional to dynamic semantics*, Submitted, 2000.

[13] Chris Fox, *Discourse representation, type theory and property theory*, ***Proc. international workshop on computational semantics*** (H. Bunt, R. Muskens, and G. Rentier, editors), ITK, Tilburg, 1994.

[14] Gerald Gazdar, ***Pragmatics: Implicature, presupposition and logical form***, Academic Press, New York, 1979.

[15] F. Kamareddine and E. Klein, *Nominalization, predication and type containment*, ***Journal of Logic, Language and Information***, vol. 2 (1993), pp. 171–215.

[16] H. Kamp and U. Reyle, ***From discourse to logic***, Kluwer Academic Publishers, Dordrecht, 1993.

[17] Hans Kamp, *Events, instants and temporal reference*, ***Semantics from different points of view*** (R. Bäuerle, U. Egli, and A. von Stechow, editors), Springer, Berlin, 1979.

[18] ———, *Prolegomena to a structural account of belief and other attitudes*, ***Propositional attitudes*** (C.A. Anderson and J. Owens, editors), CSLI Lecture Notes Number 20, Stanford, 1990.

[19] Lauri Karttunen, *Presupposition and linguistic context*, ***Theoretical Linguistics***, (1974), pp. 181–194.

[20] E. Krahmer and P. Piwek, *Presupposition projection as proof construction*, ***Computing meaning***, Kluwer Academic Publishers, Dordrecht, 1999.

[21] Peter Krause, *Presupposition and abduction in type theory*, ***Computational logic and natural language processing***, South Queensferry, Scotland, 1995.

[22] Per Martin-Löf, ***Intuitionistic type theory***, Bibliopolis, Napoli, 1984, Notes by Giovanni Sambin of a series of lectures given in Padua, June 1980.

[23] J. McCarthy and S. Buvač, *Formalizing context (expanded notes)*, ***Computing natural language*** (A. Aliseda, R. van Glabbeek, and D. Westerståhl, editors), CSLI Lecture Notes Number 81, Stanford, 1997.

[24] Richard Montague, *English as a formal language*, ***Formal philosophy*** (R. Thomason, editor), Yale University Press, New Haven, 1974.

[25] Aarne Ranta, ***Type-theoretical grammar***, Oxford University Press, Oxford, 1994.

[26] ———, *Grammatical framework tutorial*, Available at `www.xrce.xerox.com/research/mltt/gf/home.html`, 1999.

[27] Göran Sundholm, *Proof theory and meaning*, ***Handbook of philosophical logic*** (D. Gabbay and F. Guenthner, editors), vol. 3, Reidel, Dordrecht, 1986.

COMPUTER SCIENCE
TRINITY COLLEGE
DUBLIN 2, IRELAND
E-mail: Tim.Fernando@tcd.ie