

A Note on Interpretations for Federated Languages and the Use of Disquotation

Steven O. Kimbrough
University of Pennsylvania
3730 Walnut Street, Suite 500
Philadelphia, PA 19104 USA
kimbrough@wharton.upenn.edu

ABSTRACT

Quotation and subsequent disquotation of propositional content sentences has been proposed for, and has some attractive features for, representing content sentences embedded in complex sentences by subordinating clauses. The meaning of and proper semantics for such representations, however, remain obscure, amid lurking worries that any such language will be inconsistent. This paper sketches an account of a semantics for quotation and disquotation. The key technical devices employed are (1) composing languages by federating two or more distinctly defined languages and (2) treating quotation as an invertible enumeration device, employed in the conventions governing use of the federated language.

1. BACKGROUND

Linguists and grammarians have usefully distinguished three main types of sentences (e.g., [11, pages 142–4]). First, a *simple sentence* consists of a single clause (verb plus noun phrase(s)) that “stands alone as a sentence.” Roughly, a simple sentence corresponds to a predicate in first order logic with its arguments properly filled in. Second, a *coordinate sentence* is composed from simple sentences by such coordinating operators as *and*, *but*, and *or*. Roughly, in first order logic coordinate sentences are composed using the logical constants on predicates.

Third, a *complex sentence* is one in which “a clause can be incorporated *into* another clause” [11, page 143]. Davidson’s famous example, “Galileo said that the earth moves” [10], serves to illustrate important distinctions for dealing with complex sentences. “the earth moves” is a clause (roughly, verb+noun phrase properly combined) that is here “incorporated” into the clause “Galileo said”. The use of “that” is diagnostic of complex sentences; it typically serves to indicate the *subordinator* (subordinating clause, here “Galileo said”) and the *subordinated* (or embedded) clause (here “the earth moves”). Adding to the relevant terminology, philosophers speak of an embedded clause (or the proposition it

represents) as the *propositional content* of the embedding (complex) sentence. Also, often in practice “that” as a subordinate clause indicator is dropped, because no ambiguity results, as in “Galileo said the earth moves”.

Linguistic complexity of this kind—sentences with embedded clauses, utterances that refer to other utterances—is quite common, and as noted is often signaled by the use of “that”. Here are two examples, drawn by convenience from fiction but quite representative of actual dialog.

“There are too many other things that could have happened,” I said. “That she did go away with Lavery and they split up. That she went away with some other man and the wire is a gag. That she went away alone or with a woman. That she drank herself over the edge and is holed up in some private sanatorium taking a cure. That she got into some jam we have no idea of. That she met with foul play.”

“Good God, don’t say that,” Kingsley exclaimed.
– *The Lady in the Lake*, chapter 2, Raymond Chandler

“The gun, you know, is kind of queer.”

“But I told you I found it lying on the stairs,” she said angrily. “I don’t know anything else about it. I don’t know anything about guns at all. I—I never shot one in my life.” She opened a large blue bag and pulled a handkerchief out of it and sniffled.

“That’s your story,” I said. “I don’t have to get stuck with it.”

– *The Lady in the Lake*, chapter 15, Raymond Chandler

Talk about talk is complex in our present context (sentences with embedded clauses). It is ubiquitous and, in particular, it suffuses legal discourse (see [12] for many examples). In addition, there is a well-recognized catalog of subordinating expression types (see [23] for a discussion), including those normally associated with these verbs: *believes* (as in *X believes that P*), *desires*, *intends*, and other mentalistic concepts; *asserts*, *promises*, *declares*, *commands*, and other speech acts; *it is necessary that*, *it is possible that*, *it is obligatory that*, *it is permitted that*, and other broadly modal concepts; as well as such notions as *seeing to it that*, *feeling that*, *seeing that*, *hearing that*, *requiring that*, and *needing that*.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICAIL ’05, June 6–11, 2005, Bologna, Italy.

Copyright 2005 ACM 1-59593-081-7/05/0006 ...\$5.00.

Complex sentences, utterances with embedded propositional content, pose considerable challenges to formalization in logic. First order logic does not permit, at least in any straightforward way, representation of embedded clauses. Moreover, first order logic is extensional and most complex sentences are intensional to some degree. Simplifying, we might express the generic complex sentence as having the form P that Q , where P is the subordinator, the embedding clause, and Q the subordinated, or embedded, clause (think: $P = \textit{Galileo said}$; $Q = \textit{the earth moves}$). In many cases, it may be true that P that Q and that $Q \leftrightarrow R$, but not that P that R . For example, let Q be $2+2=4$ and R be $123-121=8-6$. Not only is it the case that $Q \leftrightarrow R$, but it is necessarily the case, yet it may well be true that Galileo said that Q and did not ever say that R . Similarly, it may be true that P that $Q(a)$, that $a = b$, and not true that P that $Q(b)$. Standard examples are created from $a = \textit{the morning star}$ and $b = \textit{the evening star}$, and from $a = \textit{Cicero}$ and $b = \textit{Tully}$. The morning star is, at it happens, one and the same as the evening star. *Cicero* and *Tully* are two names for the same historical individual. Also, we have Superman and Clark Kent, and Batman and Bruce Wayne, among others. Cases are unexotically generated with definite descriptions, even if one’s favorite example is exotic: $a = \textit{Jocasta}$, $b = \textit{the mother of } \textit{Edipus}$, and $c = \textit{the wife of } \textit{Edipus}$. (Mackie [27] contains a commendably clear and accessible treatment of these issues.)

Logicians have developed a rich collection of extensions of propositional and first order logic for modeling and exploring particular classes of subordinators. Thus, for example, modal logics have been developed to formalize and explore a related group of subordinators including *It is necessary*, and *It is possible* (cf., [8, 13, 14]). Similarly, deontic logics have been developed for formalizing and exploring such subordinators as *It is obligatory*, *It is permitted*, and *It is forbidden* (cf., [8, 18]). In these logics, sentence (formula) operators are introduced to model the various subordinators under study. Thus, we have the familiar symbols \Box and \Diamond of modal logic, meant to represent *It is necessary that* and *It is possible that*, and $\Box P$ and $\Diamond Q$ become well-formed formulas. This *operator strategy for formalizing subordination* has been used successfully with a growing range of subordinators, including believing, intending, desiring, acting (X sees to it that), speech acts (cf., [16, 17, 18, 20]), and institutional power (see [21] for exploration of the diadic subordinator *counts as*). The operator strategy has been productive, has yielded useful logics rigorously specified, and continues to enjoy much attention and growth.

An alternative complements the operator strategy. It is called the *disquotational strategy for formalizing subordination* and it has begun to be explored (cf., [23, 24, 19, 26]). The key move in the disquotational approach is to treat a so-called quoted formula (is it ‘quoted’?) as a first order term and to provide some form of meaning postulates that unpack—disquote—the embedded formula in order to associate it truth functionally with other formulas. The simplest of examples will suffice for present purposes. Let the intended interpretation of $\textit{Promise}(x, y, z)$ be *The event x is a promising by y of z* . For the particular case of s promising via event e that P we write

$$\textit{Promise}(e, s, [P]) \tag{1}$$

Associated with this is the meaning postulate

$$\textit{Promise}(e, s, [P]) \rightarrow (\textit{Kept}(e) \leftrightarrow P) \tag{2}$$

which may be read as *If e is the event of s promising that P , then the promise named by e is kept if and only if P* .¹

There is much to be said in principle in favor of a disquotation approach to representing embedded content. The papers cited above present evidence in this regard and prototype implementations have yielded positive results (e.g., [1, 2, 3, 29]). From a theoretical perspective, note that intensionality comes in degrees. The disquotation approach defaults to the strongest level of intensionality (substitution of terms of identical meaning is not guaranteed to preserve truth) and allows, via meaning postulates, arbitrary relaxation of intensionality. This affords the modeler great flexibility. From a practical perspective, note that finding and integrating the appropriate operator-based intensional logics may often be challenging—and the required logics may not exist. The disquotation approach affords an accessible, incremental approach to modeling and implementation. It is possible to identify some of the logical structure required and to implement it with the disquotation approach, leaving open the possibility of integrating its representations with operator-oriented logics. See [25] and [19] for exercises of this sort. The disquotation approach is properly seen as a complement to the operator approach.

The disquotation approach is, however, not without its difficulties. Principal among them is the question of providing a semantics. Expressions (1) and (2) are not wffs (well-formed formulas) in first order logic. In what logic are they wffs? Is this logic complete? Is it consistent?

The aim of this paper is to address these questions. To this end I develop and explore the concept of what may be called a *federated language*. A federated language is a language comprising two or more distinct languages. I shall refer to these latter languages as *federating languages*. Our interest here is mainly in federated languages whose federating (sub)-languages are languages of formal logic, in particular sentence logic and first order logic.² The goal of the present paper is to present and develop the concept, and then to apply it to the problem of making semantic sense of disquotation as used to represent embedded content.

2. PRELIMINARIES

Recall that a *formal language* (of logic), \mathcal{L} , is specified by five elements:

1. Symbols. It will be useful to distinguish three kinds of symbols:

- (a) Logical constants.

E.g., $\neg, \wedge, \rightarrow$

¹Very many details are being suppressed, or elided, in the interests of focusing on the matters to hand. For example, a valid promise is arguably to the effect that one promises that *one sees to it that P* . Also, as emphasized by Searle [32, 33], promises are arguably forward-looking; what it is one promises must be something that occurs in the future, certainly not the past. I believe that the elisions committed are without prejudice to a complete and workable theory.

²The term *federated language* is a neologism so far as I know, motivated by analogy with *federated database*. Bowen and Kowalski [7] introduced the term *amalgamated language* for a distinct but related concept.

- (b) Punctuation.
E.g. (,)
- (c) Logical terms.
E.g.,
 - i. Statement letters: P, Q
 - ii. Variables: x, y
 - iii. Names: a, b
 - iv. Functions: f, g

I shall make the usual (but usually unstated) assumption that the symbols of \mathcal{L} are partitioned by the logical constants, the punctuation, and the logical terms. That is, every symbol of \mathcal{L} is in exactly one of these three sets.

2. Well-formed terms (wfts).

These are normally specified by *rules of formation* on the set of symbols, e.g.,

- (a) If ϕ is variable, then ϕ is wft.
- (b) If ϕ is name, then ϕ is wft.

Denote the set of wfts of \mathcal{L} by $wft(\mathcal{L})$.

3. Well-formed formulas (wffs).

These are normally specified by *rules of formation* on the set of symbols, e.g.,

- (a) If ϕ is a statement letter, then ϕ is a wff.
- (b) If ϕ is a statement letter, then $\neg\phi$ is a wff.

Denote the set of wffs of \mathcal{L} by $wff(\mathcal{L})$.

4. Axioms.

These are often given as axiom schemas, e.g., for all ϕ and $\psi \in \mathcal{L}$, $(\phi \rightarrow (\psi \rightarrow \phi))$.

5. Rules of derivation.

E.g., *modus ponens* or *detachment*: Given ϕ and $(\phi \rightarrow \psi)$, then ψ may be directly inferred.

Our attention here will be focused on federating languages of quite standard logic: the propositional calculus and first order logic. It matters for present purposes not at all which logical systems are used. For the sake of definiteness and accessibility, however, I shall have in mind the systems described in Richard Jeffrey's textbook [15], which use truth tree methods. These systems have no axioms at all and rules of derivation (including tree construction rules) that are especially convenient and transparent.

The axioms and rules of derivation for a logical language might be called its *logical basis*. It will simplify matters, without cost of generality, to stick to a single logical basis throughout our discussion. For the same reason, it will be useful to restrict our attention to federating languages with identical sets of logical constants and punctuation marks. With these assumptions to hand, the following definitions are in order.

DEFINITION 1 (LANGUAGE TYPE). A language type is a specified logical language, which specification is unique up to its logical terms. Uniformly and unambiguously renaming (some or all of) its logical terms (e.g., renaming P as Q) does not alter the type of the language.

Note: For the sake of precluding ambiguity, new names may not duplicate old names.

DEFINITION 2 (LANGUAGE TOKEN). A language token is an instance of a given language type and is identified by its logical terms. Uniformly and unambiguously renaming (some or all of) its logical terms (e.g., renaming P as Q) does not alter the type of the language, but does change one token of the language into another.

DEFINITION 3 (ATOMICALLY DISJOINT). Two sets of expressions, Δ^1 and Δ^2 , are said to be atomically disjoint if they have no logical terms in common. Two language tokens, \mathcal{L}^1 and \mathcal{L}^2 , are said to be atomically disjoint if they have no logical terms in common (since their respective sets of wffs are atomically disjoint).

Comments:

1. In the case of two language tokens of type propositional logic, the languages are atomically disjoint if and only if they have syntactically different sentence letters. In the case of first order logic, predicates, variables, names, and functions will have to be syntactically distinct if two language tokens are to be atomically disjoint.
2. If two language tokens of the same type are atomically disjoint, then any well-formed expression in either language can unambiguously be assigned to its proper language; no expression—whether term or formula—is well formed in more than one of the two language tokens.

3. FEDERATING LANGUAGES OF PROPOSITIONAL LOGIC

Let \mathcal{L}^1 be a language (token) of propositional logic, specified as in figure 1.

Let \mathcal{L}^2 be language token of the language type propositional logic and be atomically disjoint from \mathcal{L}^1 . Let $S^2 = \{P_i^2 | i = 1, 2, \dots\}$. (By atomic disjointness, $S^1 \cap S^2 = \emptyset$.) The specifications of \mathcal{L}^1 and \mathcal{L}^2 are identical, except for their atomic sentence letters, S^1 and S^2 . Thus, the specification of \mathcal{L}^2 is as in figure 1, with S^2 substituted for S^1 .

Given \mathcal{L}^1 and \mathcal{L}^2 specified as indicated, the language token \mathcal{L}^3 is federated from (the atomically disjoint federating languages) \mathcal{L}^1 and \mathcal{L}^2 (we shall say $\mathcal{L}^3 = \mathcal{L}^1 \circ \mathcal{L}^2$) as follows: \mathcal{L}^3 is specified as in figure 1, with $S^3 = S^1 \cup S^2$ uniformly substituted for S^1 . Line 1(c)i of figure 1 becomes

- Statement letters: Every element of $S^3 = S^1 \cup S^2$.

Line 3a of figure 1 becomes

- If $\phi \in S^3$, then ϕ is a wff.

Given that a proper semantics for \mathcal{L}^1 (a standard propositional logic) is given via the valuation function V^1 and that a proper semantics for \mathcal{L}^2 is similarly given via V^2 , it is straightforward to give a proper valuation function, V^3 , for \mathcal{L}^3 . Figure 2 provides the definition. A word on what it is to be a proper semantics. Even for an utterly standard propositional logic, \mathcal{L}^1 for example, not every valuation function is proper. In particular, valuation functions must be complete

1. Symbols.
 - (a) Logical constants: $\neg, \wedge, \rightarrow, \vee, \leftrightarrow$
 - (b) Punctuation: $(,)$
 - (c) Logical terms.
 - i. Statement letters: Every element of $S^1 = \{P_i^1 | i = 1, 2, \dots\}$.
2. Well-formed terms (wfts): none
3. Well-formed formulas (wffs):
 - (a) If $\phi \in S^1$, then ϕ is a wff.
 - (b) If ϕ is a wff, then $\neg(\phi)$ is a wff.
 - (c) If ϕ is a wff and ψ is a wff, then $(\phi \wedge \psi)$ is a wff.
 - (d) If ϕ is a wff and ψ is a wff, then $(\phi \vee \psi)$ is a wff.
 - (e) If ϕ is a wff and ψ is a wff, then $(\phi \rightarrow \psi)$ is a wff.
 - (f) If ϕ is a wff and ψ is a wff, then $(\phi \leftrightarrow \psi)$ is a wff.
 - (g) Nothing else is a wff.
4. Axioms: \emptyset , as in [15].
5. Rules of derivation: truth tree method, as in [15].

Figure 1: Specification of \mathcal{L}^1 , a token of type propositional logic

and determinate in the sense that they provide a unique valuation for each and every wff in the language.³ V^1 is not proper, then, if the following is among its rules:

- $V^1(P_1^1) = \top$ if this sentence is false; otherwise $V^1(P_1^1) = \perp$.

We can get into similar difficulties if we allow, say

- $V^1(P_1^1) = \top$ if $V^2(P_2^2) = \perp$; otherwise $V^1(P_1^1) = \perp$.
- $V^2(P_2^2) = \top$ if $V^1(P_1^1) = \top$; otherwise $V^2(P_2^2) = \perp$.

In consequence we prohibit V^1 from having functional dependence on \mathcal{L}^2 ; similarly, V^2 cannot be functionally dependent on \mathcal{L}^1 in any way. In short, we require that both V^1 and V^2 be proper valuation functions for their respective languages.

OBSERVATION 1. *The language token \mathcal{L}^3 composed (by federation) in this manner is of type propositional logic. This may be seen by syntactic transformation of \mathcal{L}^3 into, e.g., \mathcal{L}^4 as follows: $P_i^1 \rightsquigarrow P_{2i-1}^4$ and $P_i^2 \rightsquigarrow P_{2i}^4$. Then \mathcal{L}^4 and \mathcal{L}^1 are syntactic variants: uniformly substituting P_i^1 for P_i^4 transforms \mathcal{L}^4 into \mathcal{L}^1 .*

OBSERVATION 2. *Federation does not generate inconsistency.*

This may be stated more precisely and proved with the following lemma.

LEMMA 1. *Let Γ^1 be a consistent set of wffs in \mathcal{L}^1 and Γ^2 a consistent set of wffs in \mathcal{L}^2 . Let $\Gamma^3 = \Gamma^1 \cup \Gamma^2$ be a set of wffs in $\mathcal{L}^3 = \mathcal{L}^1 \circ \mathcal{L}^2$. Then Γ^3 is consistent.*

³I note, but elide for the sake of simplicity, the issue of evaluating first order formulas with free (unbound) variables.

1. If $\phi \in S^3$ then $V^3(\phi) = V^1(\phi)$ if $\phi \in S^1$, and $V^3(\phi) = V^2(\phi)$ if $\phi \in S^2$.
Comment: Since $S^1 \cap S^2 = \emptyset$ and since V^1 is a proper interpretation of \mathcal{L}^1 and V^2 is a proper interpretation of \mathcal{L}^2 , it follows that V^3 is fully and uniquely defined for $\phi \in S^3$.
2. $V^3(\neg\phi) = \top$ if $V^3(\phi) = \perp$; otherwise $V^3(\neg\phi) = \perp$
3. $V^3(\phi \wedge \psi) = \top$ if $V^3(\phi) = \top$ and $V^3(\psi) = \top$; otherwise $V^3(\phi \wedge \psi) = \perp$.
4. $V^3(\phi \vee \psi) = \top$ if $V^3(\phi) = \top$ or (inclusively) $V^3(\psi) = \top$; otherwise $V^3(\phi \vee \psi) = \perp$.
5. $V^3(\phi \rightarrow \psi) = \top$ if $V^3(\phi) = \perp$ or $V^3(\psi) = \top$; otherwise $V^3(\phi \rightarrow \psi) = \perp$
6. $V^3(\phi \leftrightarrow \psi) = \top$ if $V^3(\phi) = V^3(\psi)$; otherwise $V^3(\phi \leftrightarrow \psi) = \perp$.

Figure 2: A Federated Semantics for the Federated Propositional Language \mathcal{L}^3

Since proofs serve the dual goals of establishing results and promoting insight, we offer two proofs of the lemma. In the semantic sense, consistency of a set of wffs means that there is some interpretation under which each member of the set is true. In the proof-theoretic sense, consistency of a set of wffs means that there is no formula for which both it and its denial may be proved.

PROOF : SEMANTIC VERSION. Since Γ^1 is consistent, there is an interpretation (or valuation) on S^1 (an assignment of truth values to each sentence letter in S^1), call it $V^1(S^1)$, under which every wff in Γ^1 is true. Similarly, let $V^2(S^2)$ be an interpretation which makes every wff in Γ^2 true. Let $V^i(P_j)$ be the value (here truth value) assigned by V^i on S^i to $P_j \in S^i$. Then, since $S^1 \cap S^2 = \emptyset$,

$$V^3(P_j) = \begin{cases} V^1(P_j) & \text{if } P_j \in S^1 \\ V^2(P_j) & \text{if } P_j \in S^2 \end{cases} \quad (3)$$

is well-defined and constitutes an interpretation, $V^3(S^3)$ (recall $S^3 = S^1 \cup S^2$ and $S^1 \cap S^2 = \emptyset$), in which every wff in $\Gamma^3 = \Gamma^1 \cup \Gamma^2$ is true. \square

PROOF : PROOF-THEORETIC VERSION. By definition, Γ^3 is consistent if and only if it is not possible in \mathcal{L}^3 to derive both ϕ and $\neg\phi$. Consider two cases: (i) $\phi \in S^3 = S^1 \cup S^2$ (so ϕ is an atomic sentence), and (ii) $\phi \notin S^3 = S^1 \cup S^2$ (so ϕ is not an atomic sentence). We need consider only case (i), since if we can derive $(\phi \wedge \neg\phi)$ for some non-atomic wff ϕ , it will be trivial in the logic to hand to derive $(\psi \wedge \neg\psi)$ for any atomic wff ψ .

If no atomic wff ϕ is derivable from Γ^3 , then Γ^3 is consistent. Alternatively, let $\phi \in S^3$ be derivable from Γ^3 . Since Γ^1 and Γ^2 are each consistent and $S^1 \cap S^2 = \emptyset$, if $\phi \in S^1$ then Γ^2 is irrelevant to any derivation of ϕ or $\neg\phi$. Similarly, if $\phi \in S^2$ then Γ^1 is irrelevant to any derivation of ϕ or $\neg\phi$. Because Γ^1 and Γ^2 are atomically disjoint, no wff in Γ^1 or any wff produced by applications of the rules of inference on wffs in Γ^1 can directly contradict any wff in Γ^2 , or any wff produced by applications of the rules of inference on wffs in Γ^2 . Thus, if $\Gamma^3 \vdash \phi$, then either $\phi \in S^1$ and $\Gamma^1 \vdash \phi$ or

$\phi \in S^2$ and $\Gamma^2 \vdash \phi$.

Assume without loss of generality that $\phi \in S^1$. By consistency of Γ^1 , $\Gamma^1 \not\vdash \neg\phi$. By consistency of Γ^2 and $S^1 \cap S^2 = \emptyset$, $\Gamma^2 \not\vdash \neg\phi$. Thus, Γ^3 is consistent, since for no ϕ is it true that $\Gamma^3 \vdash \phi$ and $\Gamma^3 \vdash \neg\phi$. \square

OBSERVATION 3. *It is entirely possible to have intended interpretations of \mathcal{L}^1 and \mathcal{L}^2 that produce nominal but not formal contradictions in \mathcal{L}^3 .*

An interpretation in sentence logic is an assignment of sentence letters to $\{\top, \perp\}$. An *intended interpretation* is an assignment of sentence letters to states of affairs in the world, e.g., $P_1^1 = \text{More noncombatants have been killed during the American occupation of Iraq than people were killed by the Boxing Day tsunami of 2004}$. While the intended interpretation serves to motivate one's use of a language, it has no role in the semantics or proof theory for the language. For any interpretation, more than one intended interpretation is possible.

By way of illustrating this last observation, let us have the following intended interpretations:

P_1^1 It is Tuesday.

P_2^1 It is raining.

P_1^2 It is Tuesday.

P_2^2 It is raining.

The the following sequent is valid in \mathcal{L}^3 :

$$P_1^1, (P_1^1 \rightarrow P_2^1), P_2^2, (P_2^2 \rightarrow \neg P_2^2) \models (P_2^1 \wedge \neg P_2^2) \quad (4)$$

While this seeming failure to detect contraction may seem odd, it is not an anomaly. \mathcal{L}^3 is simply an instance of sentence logic. The behavior on display here is quite available in any instance of sentence logic, as may be seen by applying our previous transformation:

P_1^4 It is Tuesday.

P_3^4 It is raining.

P_2^4 It is Tuesday.

P_4^4 It is raining.

which leads to

$$P_1^4, (P_1^4 \rightarrow P_3^4), P_2^4, (P_2^4 \rightarrow \neg P_4^4) \models (P_3^4 \wedge \neg P_4^4) \quad (5)$$

If a composed (i.e., federated) language such as \mathcal{L}^3 is simply a token of the type associated with its federating languages, what advantage could it possibly offer over another token, e.g., over one of its components? Any advantage would have to lie in syntax, rather than in proof theory or model theory. But one syntax may well afford us advantages not provided by another. Certainly it is possible to conceive of a hideously cumbersome syntax. On the positive side we might think of \mathcal{L}^1 and \mathcal{L}^2 as offering different perspectives on a given topic. Certain questions are more conveniently asked given a felicitous syntax. For example, given the intended interpretations

P_2^1 It is raining.

P_2^2 It is raining.

we can easily ask if, under present assumptions, the two perspectives agree on whether it is raining. It is simply convenient for this task that P_2^i “means” in language i that it is raining. Syntactic arrangements of this sort are replaceable and augmentable by, for example, maintaining tables of correspondence between formulas in \mathcal{L}^1 and \mathcal{L}^2 . Such tables are, of course, outside the languages in question, yet they may be useful in working with and within the languages. As we shall see, more exciting possibilities attend a composed first order language.

1. Symbols.

- (a) Logical constants: $\neg, \wedge, \rightarrow, \vee, \leftrightarrow$
- (b) Punctuation: $(,)$
- (c) Logical expressions.
 - i. Predicate letters (without regard to arity): Every element of $S^1 = \{P_i^1 | i = 1, 2, \dots\}$.
 - ii. Functions (without regard to arity): Every element of $F^1 = \{f_i^1 | i = 1, 2, \dots\}$.
 - iii. Name constants: $N^1 = \{a_i^1 | i = 1, 2, \dots\}$.
 - iv. Variables: $X^1 = \{x_i^1 | i = 1, 2, \dots\}$.

2. Well-formed terms (wfts):

- (a) If $\psi \in N^1$, then ψ is a wft.
- (b) If $\psi \in X^1$, then ψ is a wft.
- (c) If $\psi \in F^1$ and has arity n , and $\chi_1, \chi_2, \dots, \chi_n$ are each wfts, then $\psi(\chi_1, \chi_2, \dots, \chi_n)$ is a wft.

3. Well-formed formulas (wffs):

- (a) If $\phi \in S^1$ with arity n , and $\psi_1, \psi_2, \dots, \psi_n$ are each wfts, then $\phi(\psi_1, \psi_2, \dots, \psi_n)$ is a wff.
- (b) If ϕ is a wff, then $\neg(\phi)$ is a wff.
- (c) If ϕ is a wff and ψ is a wff, then $(\phi \wedge \psi)$ is a wff.
- (d) If ϕ is a wff and ψ is a wff, then $(\phi \vee \psi)$ is a wff.
- (e) If ϕ is a wff and ψ is a wff, then $(\phi \rightarrow \psi)$ is a wff.
- (f) If ϕ is a wff and ψ is a wff, then $(\phi \leftrightarrow \psi)$ is a wff.
- (g) If ϕ is a wff and $\psi \in X^1$, then $\exists\psi(\phi)$ is a wff.
- (h) If ϕ is a wff and $\psi \in X^1$, then $\forall\psi(\phi)$ is a wff.
- (i) If χ_1 is a wft and χ_2 is a wft, then $(\chi_1 = \chi_2)$ is a wff.
- (j) Nothing else is a wff.

4. Axioms: \emptyset , as in [15].

5. Rules of derivation: truth tree method, as in [15].

Figure 3: Specification of \mathcal{L}^1 , a token of type first order logic

4. FEDERATING FIRST ORDER LANGUAGES

Here we discuss \mathcal{L}^3 as composed (federated) from language tokens of first order logic, \mathcal{L}^1 and \mathcal{L}^2 . Predicates of \mathcal{L}^i ($i \in \{1, 2\}$), regardless of arity will be denoted P_1^i, P_2^i, \dots . Similarly,

1. Symbols.
 - (a) Logical constants: $\neg, \wedge, \rightarrow, \vee, \leftrightarrow$
 - (b) Punctuation: $(,)$
 - (c) Logical expressions.
 - i. Predicate letters (without regard to arity): Every element of $S^3 = S^1 \cup S^2$.
 - ii. Functions (without regard to arity): Every element of $F^3 = F^1 \cup F^2$.
 - iii. Name constants: $N^3 = N^1 \cup N^2$.
 - iv. Variables: $X^3 = X^1 \cup X^2$.
2. Well-formed terms (wfts):
 - (a) If $\psi \in \text{wft}(\mathcal{L}^1) \cup \text{wft}(\mathcal{L}^2)$, then ψ is a wft.
3. Well-formed formulas (wffs):
 - (a) If $\phi \in \text{wff}(\mathcal{L}^1) \cup \text{wff}(\mathcal{L}^2)$ then ϕ is a wff.
 - (b) If ϕ is a wff, then $\neg(\phi)$ is a wff.
 - (c) If ϕ is a wff and ψ is a wff, then $(\phi \wedge \psi)$ is a wff.
 - (d) If ϕ is a wff and ψ is a wff, then $(\phi \vee \psi)$ is a wff.
 - (e) If ϕ is a wff and ψ is a wff, then $(\phi \rightarrow \psi)$ is a wff.
 - (f) If ϕ is a wff and ψ is a wff, then $(\phi \leftrightarrow \psi)$ is a wff.
 - (g) If ϕ is a wff and $\psi \in X^3$, then $\exists\psi(\phi)$ is a wff.
 - (h) If ϕ is a wff and $\psi \in X^3$, then $\forall\psi(\phi)$ is a wff.
 - (i) Nothing else is a wff.
4. Axioms: \emptyset , as in [15].
5. Rules of derivation: truth tree method, as in [15], but universal instantiation and existential instantiation are modified to block introduction of non-wffs.

Figure 4: Specification of a first order $\mathcal{L}^3 = \mathcal{L}^1 \circ \mathcal{L}^2$

1. functions: f_1^i, f_2^i, \dots
2. constants: a_1^i, a_2^i, \dots
3. variables: x_1^i, x_2^i, \dots

We may now form $\mathcal{L}^3 = \mathcal{L}^1 \circ \mathcal{L}^2$, proceeding much as we did in the case of federating propositional logic tokens. Let \mathcal{L}^1 be specified as in figure 3, the first order logic analog to figure 1. We specify \mathcal{L}^2 by substituting in figure 3: S^2 for S^1 , F^2 for F^1 , N^2 for N^1 , and X^2 for X^1 . We compose the federated language $\mathcal{L}^3 = \mathcal{L}^1 \circ \mathcal{L}^2$ by substituting in figure 3: $S^3 = (S^1 \cup S^2)$ for S^1 , $F^3 = (F^1 \cup F^2)$ for F^1 , $N^3 = (N^1 \cup N^2)$ for N^1 , and $X^3 = (X^1 \cup X^2)$ for X^1 , and in addition we use the wff-formation rules of propositional logic. Details are given in figures 4 and 5.

Comments:

1. In figure 4, we have dropped the wff-formation rule:
 - If χ_1 is a wft and χ_2 is a wft, then $(\chi_1 = \chi_2)$ is a wff.

Note that specification in figure 4 does not allow as well formed such sub-wff mixed expressions as $P_1^1(a_2^2)$

1. If $\phi \in S^3$ then $V^3(\phi) = V^1(\phi)$ if $\phi \in S^1$, and $V^3(\phi) = V^2(\phi)$ if $\phi \in S^2$.
Comment: Since $S^1 \cap S^2 = \emptyset$ and since V^1 is a proper interpretation of \mathcal{L}^1 and V^2 is a proper interpretation of \mathcal{L}^2 , it follows that V^3 is fully and uniquely defined for $\phi \in S^3$.
2. $V^3(\neg\phi) = \top$ if $V^3(\phi) = \perp$; otherwise $V^3(\neg\phi) = \perp$
3. $V^3(\phi \wedge \psi) = \top$ if $V^3(\phi) = \top$ and $V^3(\psi) = \top$; otherwise $V^3(\phi \wedge \psi) = \perp$.
4. $V^3(\phi \vee \psi) = \top$ if $V^3(\phi) = \top$ or (inclusively) $V^3(\psi) = \top$; otherwise $V^3(\phi \vee \psi) = \perp$.
5. $V^3(\phi \rightarrow \psi) = \top$ if $V^3(\phi) = \perp$ or $V^3(\psi) = \top$; otherwise $V^3(\phi \rightarrow \psi) = \perp$
6. $V^3(\phi \leftrightarrow \psi) = \top$ if $V^3(\phi) = V^3(\psi)$; otherwise $V^3(\phi \leftrightarrow \psi) = \perp$.
7. $V^3(\forall\psi^i(\phi)) = V^3(\phi)$ if $\psi^i \in X^i$ does not occur freely in ϕ . If $\psi^i \in X^i$ does occur freely in ϕ , $V^3(\forall\psi^i(\phi)) = \top$ if for every $n \in N^i$, $V^3(\phi[x^i/n]) = \top$; otherwise $V^3(\forall\psi^i(\phi)) = \perp$.
Comment: $\phi[x^i/n]$ is the wff obtained from ϕ by substituting n for each freely occurring instance of x^i . Assumed is that $i \in \{1, 2\}$.
8. $V^3(\exists\psi^i(\phi)) = V^3(\phi)$ if $\psi^i \in X^i$ does not occur freely in ϕ . If $\psi^i \in X^i$ does occur freely in ϕ , $V^3(\exists\psi^i(\phi)) = \top$ if for some $n \in N^i$, $V^3(\phi[x^i/n]) = \top$; otherwise $V^3(\exists\psi^i(\phi)) = \perp$.

Figure 5: A Federated Semantics for the Federated First Order Language \mathcal{L}^3

(although vacuous quantification is permitted, for example, $\exists x_3^2(P_2^1(a_5^1))$). The effect of this is to separate the domains of the two sublanguages, \mathcal{L}^1 and \mathcal{L}^2 , as in fact they were separated in the case of sentence logic, above.

2. Note in particular that the composed language has no variables of its own, thus quantification only ranges over a single domain, the domain of one of the sublanguages. It is perhaps best to think of \mathcal{L}^3 as a typed first order language, with the typing emerging from the composition. Put differently: there is no identity between individuals in different domains and predicates group only individuals in a single domain. Given this restriction, creating an instance of a first order language from \mathcal{L}^3 (so composed) is straightforward. Thus, analogs of the observations and the lemma regarding propositional logic, in §3, obtain in the case of first order logic. In the interests of avoiding unnecessary pedantry, I leave these as an exercise for the reader.

We turn now to an application for federated languages.

5. REPRESENTATION OF PROPOSITIONAL CONTENT

Organizations conducting business transactions over time will design relational database schemas that facilitate the

addition and deletion of records about transactions as they occur. For this purpose, the underlying schemas need to be robust in a least two senses: expressive power (they need to be able to accommodate the transaction information) and consistency (they need to remain coherent as records are added and deleted). While the relational model is strong on consistency, it is generally acceded to be not well suited for representation of complex sentences and propositional content, at least in any straightforward way. The aim for the present section is to sketch an account both of (1) how we can represent complex sentences and propositional content in a semantically proper way, and (2) how such a representation is suitable for recording business transactions, and hence for serving as a proper basis for practical inference.

The exposition proceeds first by example and demonstration, then by generalization. Starting with a concrete example, recall expressions (1), a record of the utterance of a promise, and (2), a meaning postulate pertaining to the promise. We'll proceed in steps to make sense of these expressions.

For the sake of simplicity, let \mathcal{L}^1 be the \mathcal{L}^1 of §3, a standard propositional logic language. Let \mathcal{L}^2 be the \mathcal{L}^2 of §4, a standard first order logic, but with the following minor modifications. (\mathcal{L}^2 will remain a first order logic.) Let the set of name constants in \mathcal{L}^2 , N^2 , have its elements relabeled and partitioned into three distinct sets: $\mathcal{E}^2 = \{e_1^2, e_2^2, \dots\}$ (for naming events); $\mathcal{O}^2 = \{o_1^2, o_2^2, \dots\}$ (mnemonic: other); and $\mathcal{A}^2 = \{a_1^2, a_2^2, \dots\}$. With $N^2 = \mathcal{E}^2 \cup \mathcal{O}^2 \cup \mathcal{A}^2$, we then create $\mathcal{L}^3 = \mathcal{L}^1 \circ \mathcal{L}^2$ in the usual way, by federation, as specified in figure 4.

We, the users of \mathcal{L}^3 , now build a *correspondence convention*. Under this convention, which is outside of \mathcal{L}^3 , we say $x \Rightarrow y$ if we want x to correspond to y . Our first convention (there will be others) is that $o_1^2, o_2^2, \dots \Rightarrow P_1^1, P_2^1, \dots$, or in short $o_i^2 \Rightarrow P_i^1$. Formally and explicitly (and for future reference):

CORRESPONDENCE CONVENTION 1. $o_i^2 \Rightarrow P_i^1$

We require of all correspondence conventions what is evident here: that they specify a 1-1 relationship between name constants in \mathcal{L}^2 and wffs in \mathcal{L}^1 . Corresponding to the first correspondence convention we have the first *augmentation convention*.

AUGMENTATION CONVENTION 1. *Given a collection of sentences in Γ^3 , a user, s , of \mathcal{L}^3 may add to Γ^3 an expression of the form $Promise(e, s, o)$, provided $e \in \mathcal{E}^2$, $o \in \mathcal{O}^2$, $Promise \in S^2$, and $s \in \mathcal{A}^2$, and provided that the name e does not occur in Γ^3 .*

Further, if an expression of the form $Promise(e, s, o)$ is added, so must an expression of the form $Promise(e, s, o) \rightarrow (Kept(e) \leftrightarrow P)$, where $o \Rightarrow P$ is in Correspondence Convention 1, and $Kept \in S^2$.

Because we require that e be a heretofore unused name, every added expression of the form $Promise(e, s, o)$ will be unique. In a relational database context, e could serve as the key for the corresponding record and thereby guarantee the continued consistency of the database. Here, we need a stronger condition: That $\Gamma^3 \not\vdash \neg Promise(e, s, o)$. Of course, for any Γ and ϕ , if $\Gamma \not\vdash \neg \phi$ then $\Gamma \cup \{\phi\}$ is consistent. We note, however, that $Promise(e, s, o)$ is consistent, so $\not\vdash \neg Promise(e, s, o)$. Further, since e is not present in Γ^3 , so long as Γ^3 is itself consistent and there is no

occurrence of $Promise(x, s, o)$ in Γ^3 where x is governed by a universal quantifier, then $\Gamma^3 \not\vdash \neg Promise(e, s, o)$ and $\Gamma^3 \cup \{Promise(e, s, o)\}$ will be consistent. It is entirely possible to manage the use of the language so that this obtains.

Finally, our first augmentation convention says that if we indeed add an expression of the form $Promise(e, s, o)$ to Γ^3 , then we must also add a meaning postulate of the form

$$Promise(e, s, o) \rightarrow (Kept(e) \leftrightarrow P) \quad (6)$$

where $o \Rightarrow P$ from our first correspondence convention, and $Kept \in S^2$. Note that adding expression (6) to a consistent $\Gamma^3 \cup \{Assert(e, s, o)\}$ will produce a consistent set, if it is not the case that $Kept(x)$ occurs in Γ^3 where x is governed by a universal quantifier. Note as well that adding additional promises in this fashion will not generate an inconsistency. (If one addition doesn't because these general conditions apply, neither will any finite number of such additions.)

With these conventions to hand, we can coherently express in \mathcal{L}^3 promises whose propositional contents are atomic sentences in \mathcal{L}^1 . Extension to other speech acts and indeed to other subordinators is more or less straightforward. Thus, for s to assert that P we would add expressions of the form

$$Assert(e, s, o) \quad (7)$$

and

$$Assert(e, s, o) \rightarrow (Veridical(e) \leftrightarrow P) \quad (8)$$

where $o \Rightarrow P$ from our first correspondence convention, and $Veridical \in S^2$. Given that Γ^3 is managed so that this addition process maintains consistency, we need not actually add the rule expressions (as in expressions (6) and (8)) to the database. Instead, we may abstract and use them as axiom schemas, e.g.,

$$Promise(e, s, o_i) \rightarrow (Kept(e) \leftrightarrow P_i) \quad (9)$$

subject to the usual qualifications ($P_i \in S^1$, etc.). Having taken this step it is useful to note that our (first) correspondence convention is a map from an arbitrary enumeration of certain names in \mathcal{O}^2 of \mathcal{L}^2 to an arbitrary enumeration of sentence letters in S^1 of \mathcal{L}^1 . Which enumerations we use are entirely immaterial, so we may as well use the expressions as their own enumerations. Thus, we can express the first correspondence convention as follows: $[\phi] \Rightarrow \phi$, for $\phi \in wff(\mathcal{L}^1)$ and $[\phi] \in \mathcal{O}^2$ (i.e., replace o_1, o_2, \dots in \mathcal{O}^2 with $[P_1^1], [P_2^1], \dots$ for all wffs in \mathcal{L}^1). Since the wffs of \mathcal{L}^1 are denumerable, we can replace the first with the second correspondence convention:

CORRESPONDENCE CONVENTION 2. $[\phi] \Rightarrow \phi$, for $[\phi] \in \mathcal{O}^2$ and $\phi \in wff(\mathcal{L}^1)$.

Our augmentation convention is revised to reflect the new notation:

AUGMENTATION CONVENTION 2. *Given a collection of sentences in Γ^3 , a user, s , of \mathcal{L}^3 may add to Γ^3 an expression of the form $Promise(e, s, [\phi])$, provided $e \in \mathcal{E}^2$, $[\phi] \in \mathcal{O}^2$, $Promise \in S^2$, and $s \in \mathcal{A}^2$, and provided that the name e does not occur in Γ^3 .*

Further, if an expression of the form $Promise(e, s, [\phi])$ is added, so must an expression of the form $Promise(e, s, [\phi]) \rightarrow (Kept(e) \leftrightarrow \phi)$, where $[\phi] \Rightarrow \phi$ is in Correspondence Convention 2, and $Kept \in S^2$.

Finally, given the apparatus above, there is no need to limit \mathcal{L}^1 to sentence logic. So long as \mathcal{L}^1 and \mathcal{L}^2 are atomically disjoint we may create and operate a federated language, as sketched in this section. Nor is there need to limit the modeling to promises and assertions. But these are details best left for subsequent papers.

6. SUMMARY & DISCUSSION

In a federated language, \mathcal{L}^3 , every wff is either a wff of \mathcal{L}^1 , a wff of \mathcal{L}^2 , or a boolean combination of such wffs (possibly with some vacuous quantification mixed in). Because the federating languages are atomically disjoint, if Γ^1 is a consistent set of \mathcal{L}^1 formulas and Γ^2 is a consistent set of \mathcal{L}^2 formulas, then $\Gamma^3 = \Gamma^1 \cup \Gamma^2$ is also consistent, as shown above. In the context of a federated language in which at least \mathcal{L}^2 includes first order logic, or at least certain fragments of it, quotation and disquotation in the present sense is semantically unexceptionable and indeed unexotic.⁴ Quotation and disquotation are merely syntactic devices, to be used with governing conventions, for mapping an enumeration of names in \mathcal{L}^2 to formulas in \mathcal{L}^1 . Quotation—in the sense developed here—is a function—think of a collision-free hash function—that returns the index value of an (implicit) enumeration on the domain of the function, here, wffs in a federating language. Disquotation—again in the present sense—is simply the inverse of quotation; it returns the domain item that has been quoted. Under the regime sketched here the quotation function is maintained as a convention governing the use of the federated language. This and sensible management of Γ^3 will permit logically accessible expression of complex sentences while maintaining consistency.

It is interesting to compare the analysis offered here with the at most semi-formal account presented by the philosopher Donald Davidson in his perceptive essay, “On Saying That” [10]. Davidson’s views are concisely captured in the following passages, in which he focuses on “Galileo said that the earth moves” as a prototype for complex sentences (or as he calls it, indirect discourse).

The form ‘ $(\exists x)(\text{Galileo’s utterance } x \text{ and my utterance } y \text{ makes us samesayers})$ ’ is thus a way of attributing any saying I please to Galileo provided I find a way of replacing ‘ y ’ by a word or phrase that refers to an appropriate utterance of mine. And surely there is a way I can do this: I need only produce the required utterance and replace ‘ y ’ by a reference to it. Here goes:

The earth moves.
 $(\exists x)(\text{Galileo’s utterance } x \text{ and my last utterance makes us samesayers})$

Definitional abbreviation is all that is needed to bring this little skit down to:

The earth moves.
 Galileo said that.

Here the ‘that’ is a demonstrative singular term referring to an utterance (not a sentence).

⁴How tiny the fragments can be may or may not be interesting. What is interesting is that Horn clause logic, as in Prolog for example, is amply rich for these purposes.

From this point, Davidson finds it an innocuous step to reverse the order.

But in the present case nothing stands in the way of reversing the order of things, thus:

Galileo said that.
 The earth moves.

And finally

It is now safe to allow a tiny orthographic change, a change without semantic significance, but suggesting to the eye the relation of introducer and introduced: we may suppress the stop after ‘that’ and the consequent capitalization:

Galileo said that the earth moves.

Davidson summarizes as follows:

The proposal then is this: sentences in indirect discourse [what we have called complex sentences], as it happens, wear their logical form on their sleeves (except for one small point). They consist of an expression referring to a speaker, the two-place predicate ‘said’, and a demonstrative referring to an utterance. Period.

The suggestion here is that if you are attracted to Davidson’s account and you want a full formalization (Davidson did not offer one), then the disquotation approach, as outlined in this essay, is one you should find congenial. The fundamental challenge to formalizing any account such as Davidson’s is the challenge of formalizing “a demonstrative referring to an utterance” (and Davidson does not address the challenge). This is achieved, in the present account, by naming utterances (or rather their contents) and mapping the names to indices which afford automated recovery of the utterances. That, as noted, is but a suggestion. Unpacking it in detail is beyond the scope of this paper.

As advertised, this note is only a sketch of a concept and its development (the concept being the use of quotation as a device for encoding propositional content into a first order language). As always, and especially here, much remains to be done. I’ll conclude, then, with remarks on selected opportunities for further investigation.

Stronger federation. What has been presented and explored in this note is a minimal form of federation. Stronger forms may be possible and these may offer greater expressive power. For example, it is a requirement of our semantic valuation functions for each federating language—the V^i s—that they be definable without reference to the other federating language. This may not be necessary. A move analogous to that made in Russell’s theory of types may work here: language \mathcal{L}^i may make reference to language \mathcal{L}^j for $i > j$. For example, we might permit $V^2(P_1^2) = V^1(P_3^1)$ as part of an interpretation of \mathcal{L}^2 , but not as part of the interpretation of \mathcal{L}^1 , without risking a fall into inconsistency. Kaplan and Montague [22] make a similar suggestion, although they do not develop it, in their article demonstrating that certain very strong quotational languages fall into inconsistency. This theme—inconsistency of quotational languages—is explored further in [28]. The point of the present paper is to demonstrate that weaker uses of quotation can be interesting without entailing inconsistency. How much these weaker

uses can be strengthened without risking inconsistency remains an open question, a subject for future research.

Iterated subordinators. One may safely say that all linguists and grammarians agree that subordinators may be iterated. The federation examples described here do not support iterated subordination. There is no way to express, for example, that Bob said that Carol promised that Ted requested that Alice be home by midnight. This is a limitation that needs to be addressed both for theoretical and for practical purposes. If we limit ourselves to a declared maximum depth of iteration, D , then it would appear that $\mathcal{L}^{D+1} = \mathcal{L}^1 \circ \dots \circ \mathcal{L}^D$, federating D sub-languages, is a workable move. This would likely suffice for most computerized applications, provided the governing conventions can be worked out satisfactorily. Perhaps the depth of iteration need not be specified, provided the users of a system of languages obey a well-crafted convention, which would allow introduction of higher-order federation whenever required to express a deeper level of iteration.

Applications and reduction to practice. There are many topics of interest here. I note just two that particularly merit exploration. First, a federated language is a variation on the theme of combining multiple languages in a common application. This theme has a venerable history, notably in the design of meta-interpreters (often in a logic programming context). See [7, 31, 34]. It has also appeared in the model management literature (in management science) in the guise of “embedded languages” for describing and reasoning about mathematical models. See [4, 5, 6]. The second theme is the mixing the disquotation and operator approaches in a common environment. Jones and Kimbrough [19] have recently explored the syntactic integration of a first order disquotation language with an operator-based language for stating the conventions underlying various speech acts. The exploration has not been tested against representative real world problems and semantic issues remain unaddressed.

Paraconsistency. In classical logics it is possible to prove anything from a contradiction. Letting \models indicate a logical consequence relation, either semantic or proof-theoretic, it is said that \models is *explosive* (for a given logic) if and only if for every wff ϕ and ψ ,

$$\{\phi, \neg\phi\} \models \psi \quad (10)$$

Classical logics are thus explosive. Logics with non-explosive logical consequence relations are said to be *paraconsistent*. A number of such logics have been developed (see [30] for a review) and they constitute an active area of research. The trick in designing such logics, of course, is to find a principled way to contain an inconsistency. The problem is to allow certain inconsistencies without the entire system being explosive and collapsing into the pure calculus of irrelevance.

The federated languages discussed above are not paraconsistent. Recall, however, the discussion in §3 which pointed out that the following sequent is valid in \mathcal{L}^3 .

$$P_1^1, (P_1^1 \rightarrow P_2^1), P_1^2, (P_1^2 \rightarrow \neg P_2^2) \models (P_2^1 \wedge \neg P_2^2) \quad (4)$$

Because P_2^1 and P_2^2 may have the same intended meaning (viz., that it is raining) there is no impediment in principle to deriving a ‘contradiction’ of this sort. Might a federated language with disquotation be used to gain the benefit—or at least the effect—of paraconsistency without itself being inconsistent? Consider the following example. Let:

$$\Gamma^1 = \{B^1, B^1 \rightarrow F^1\} \quad (11)$$

$$\Gamma^2 = \{B^2, (B^2 \wedge P^2) \rightarrow \neg F^2\} \quad (12)$$

$$\Gamma^3 = \Gamma^1 \cup \Gamma^2 \quad (13)$$

We set up \mathcal{L}^1 and \mathcal{L}^2 so that syntactically matched sentence letters have identical intended interpretations, e.g., B^1 and B^2 each are intended to mean that Tweety is a bird. Notice that $\Gamma^3 \models F^1$ and $\Gamma^3 \not\models F^2$. Now add to Γ^3 both P^1 and P^2 . Then,

$$\Gamma^3 \cup \{P^1, P^2\} \models (F^1 \wedge \neg F^2) \quad (14)$$

This is the familiar Tweety example of defeasible reasoning and nonmonotonic logic, with F^i for Tweety flies and P^i for Tweety is a penguin.

What sense might we make of $(F^1 \wedge \neg F^2)$ and how might we use it? Recall the comment in §3 that we might think of \mathcal{L}^1 and \mathcal{L}^2 as different perspectives on a common subject. Consider by way of analogy a doctor with two different tests for diagnosing a disease. Test 1 will be positive with good probability if the patient has the disease. If the test is negative, then the patient may or may not have the disease. Test 2 is complementary. It falls positive with good probability if the patient does not have the disease, and if it is negative the patient may or may not have the disease. Neither test is entirely reliable. If they agree, are consistent in a broad sense, then very likely they give an accurate diagnosis. If test 1 is positive and test 2 negative, the patient very probably has the disease. If test 1 is negative and test 2 is positive, the patient probably does not have the disease. If both tests are negative no conclusion can be drawn by way of diagnosis. Finally, if both are positive we have the analog of expression (14). How we choose to *act* on $(F^1 \wedge \neg F^2)$ is a matter of policy. In the case of the Tweety example, a natural policy is to act on the assumption of F^2 , since \mathcal{L}^2 is more specific than \mathcal{L}^1 . In the case of a disease test, policy just might be based on evidence.

7. REFERENCES

- [1] A. Abrahams. *Developing and Executing Electronic Commerce Applications with Occurrences*. PhD thesis, University of Cambridge Computer Laboratory, 2002.
- [2] A. S. Abrahams and J. M. Bacon. A software implementation of Kimbrough’s disquotation theory for representing and enforcing electronic commerce contracts. *Group Decision and Negotiations Journal*, 11(6):1–38, November 2002.
- [3] A. S. Abrahams, D. M. Eyers, and J. M. Bacon. Practical contract storage, checking, and enforcement for business process automation. In S. O. Kimbrough and D. J. Wu, editors, *Formal Modelling for Electronic Commerce*, International Handbooks on Information Systems, pages 33–77. Springer, Berlin, Germany, 2004.
- [4] H. K. Bhargava. *A Logic Model for Model Management: An Embedded Languages Approach*. PhD thesis, University of Pennsylvania, Philadelphia, PA, 1990. Available as a technical report from the Department of Operations and Information Management.
- [5] H. K. Bhargava and S. O. Kimbrough. Model management: An embedded languages approach. *Decision Support Systems*, 10:277–299, 1993. File: BharKimDSSDolk-final-fini.

- [6] H. K. Bhargava and S. O. Kimbrough. On embedded languages, meta-level reasoning and computer-aided modeling. In S. G. Nash and A. Sofer, editors, *The Impact of Emerging Technologies on Computer Science and Operations Research*, pages 27–44. Kluwer Academic Publishers, Boston, MA, 1995. ISBN 0-7923-9542-5. File: csts-94-meta-sok-hkb.
- [7] K. Bowen and R. Kowalski. Amalgamating language and meta-language in logic programming. In K. Clark and S.-A. Tarnlund, editors, *Logic Programming*, pages 153–172. Academic Press, 1982.
- [8] B. F. Chellas. *Modal Logic: An Introduction*. Cambridge University Press, New York, NY, 1980.
- [9] D. Davidson. On saying that. *Synthèse*, 19:130–46, 1968–9.
- [10] D. Davidson. *Inquiries into Truth and Interpretation*, chapter On Saying That, pages 93–108. Clarendon Press, Oxford University Press, Walton Street, Oxford OX2 6DP, United Kingdom, 1984. Originally published as [9].
- [11] E. Finegan. *Language: Its Structure and Use*. Harcourt Brace College Publishers, Fort Worth, TX, third edition, 1999.
- [12] A. v. Gardner. *An Artificial Intelligence Approach to Legal Reasoning*. Artificial Intelligence and Legal Reasoning. MIT Press, Cambridge, MA, 1987.
- [13] G. E. Hughes and M. J. Cresswell. *An Introduction to Modal Logic*. Methuen, London, UK, 1968.
- [14] G. E. Hughes and M. J. Cresswell. *A Companion to Modal Logic*. Methuen, London, UK, 1984.
- [15] R. C. Jeffrey. *Formal Logic: Its Scope and Limits*. McGraw-Hill, New York, New York, third edition, 1991. ISBN: 0-07-032357-7.
- [16] A. Jones. On the concept of trust. *Decision Support Systems*, 33(3):225–232, 2002.
- [17] A. J. Jones. Toward a formal theory of communication and speech acts. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Communication*, System Development Foundation Benchmark, chapter 8, pages 141–160. The MIT Press, Cambridge, Massachusetts, 1990.
- [18] A. J. Jones. A logical framework. In J. Pitt, editor, *Open Agent Societies: Normative Specifications in Multi-Agent Systems*, chapter 3. John Wiley & Sons, Chichester, UK, 30 December 2004. ISBN: 047148668X.
- [19] A. J. Jones and S. O. Kimbrough. A note on modelling speech acts as signalling conventions. In S. O. Kimbrough and D. J. Wu, editors, *Formal Modelling in Electronic Commerce*, International Handbooks on Information Systems, pages 325–342. Springer, Berlin, Germany, 2004. ISBN 3-540-21431-3.
- [20] A. J. Jones and X. Parent. Conventional signalling acts and conversation. In F. Dignum, editor, *Advances in Agent Communication*, Lecture Notes in Artificial Intelligence, Volume 2922, pages 1–17. Springer-Verlag, Berlin, Heidelberg, New York, 2004.
- [21] A. J. Jones and M. J. Sergot. A formal characterisation of institutionalised power. *Journal of the Interest Group in Pure and Applied Logic (IGPL)*, 4(3):427–443, 1996. Reprinted in [36, pages 349–367].
- [22] D. Kaplan and R. Montague. A paradox regained. *Notre Dame Journal of Formal Logic*, 1:79–90, 1960. Reprinted in [35].
- [23] S. O. Kimbrough. Reasoning about the objects of attitudes and operators: Towards a disquotation theory for representation of propositional content. In *Proceedings of ICAIL '01, International Conference on Artificial Intelligence and Law*, 2001.
- [24] S. O. Kimbrough. A note on the Good Samaritan paradox and the disquotation theory of propositional content. In J. Horty and A. J. Jones, editors, *Proceedings of ΔEON'02, Sixth International Workshop on Deontic Logic in Computer Science*, pages 139–148, May 2002.
- [25] S. O. Kimbrough and Y.-H. Tan. On lean messaging with unfolding and unwrapping for electronic commerce. *International Journal of Electronic Commerce*, 5(1):83–108, 2000.
- [26] S. O. Kimbrough and Y. Yang. On representing special languages with flbc: Message markers and reference fixing in seaspeak. In S. O. Kimbrough and D. J. Wu, editors, *Formal Modelling in Electronic Commerce*, International Handbooks on Information Systems, pages 297–324. Springer, Berlin, Germany, 2004. ISBN 3-540-21431-3.
- [27] J. L. Mackie. Problems of intentionality. In Joan and P. Mackie, editors, *Logic and Knowledge: Selected Papers, Volume I*, pages 102–116. Oxford University Press, Oxford, England, 1985. ISBN: 0 19 824679 X.
- [28] R. Montague. Syntactical treatments of modality, with corollaries on reflexion principles and finite axiomatizability. *Acta Philosophica Fennica*, 16:153–67, 1963. Reprinted in [35].
- [29] S. A. Moore. *Saying and Doing: Uses of Formal Languages in the Conduct of Business*. PhD thesis, University of Pennsylvania, The Wharton School, Philadelphia, PA, 19104, USA, December 1993.
- [30] G. Priest and K. Tanaka. Paraconsistent logic. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. <http://plato.stanford.edu/archives/win2004/entries/logic-paraconsistent/>, Winter 2004.
- [31] S. Schocken and T. Finin. Meta-interpreters for rule-based inference under uncertainty. *Decision Support Systems*, 6(2):165–181, 1990.
- [32] J. R. Searle. *Speech Acts*. Cambridge University Press, Cambridge, England, 1969.
- [33] J. R. Searle and D. Vanderveken. *Foundations of Illocutionary Logic*. Cambridge University Press, Cambridge, England, 1985.
- [34] L. Sterling and E. Shapiro. *The Art of Prolog: Advanced Programming Techniques*. MIT Press, Cambridge, MA, second edition, 1994.
- [35] R. H. Thomason, editor. *Formal Philosophy: Selected Papers of Richard Montague*. Yale University Press, New Haven, CT, 1974.
- [36] E. G. Valdés et al., editors. *Normative Systems in Legal and Moral Theory – Festschrift for Carlos E. Alchourrón and Eugenio Bulygin*. Duncker & Humblot, Berlin, Germany, 1997.