Ambiguous discourse in a compositional context

An operational perspective

Tim Fernando*

revised: February 2000

Abstract. The processing of sequences of (English) sentences is analyzed compositionally through transitions that merge sentences, rather than *de*composing them. Transitions that are in a precise sense inertial are related to disjunctive and non-deterministic approaches to ambiguity. Modal interpretations are investigated, inducing various equivalences on sequences.

1 Introduction

Compositionality, as understood in this paper, is the principle that the meaning $[\![e]\!]$ of an expression e is determined by the meanings $[\![e_i]\!]$ of its parts e_i, and how those parts are put together within e. I will restrict myself to a set E_b of expressions freely generated from a set E through exactly one operation, a binary connective b. Compositionality of $[\![\cdot]\!]$ amounts in this context to nothing more nor less than the existence of a binary function β on meanings such that

$$[\![b(e_1, e_2)]\!] \quad = \quad \beta([\![e_1]\!], [\![e_2]\!]) \qquad (1)$$

for all e_1 and $e_2 \in E_b$. Equation (1) is routinely read reductively as a prescription for calculating the meaning $[\![b(e_1, e_2)]\!]$ of $b(e_1, e_2)$ by calculating $[\![e_1]\!]$ and $[\![e_2]\!]$, and then applying β to $[\![e_1]\!]$ and $[\![e_2]\!]$. The assumption underlying the induction is that we have our hands not only on β but also on the restriction of $[\![\cdot]\!]$ to the set E of *atomic* expressions. How plausible this assumption is depends on just what b and E are.

1.1 Processing sequences of sentences

The set E of atomic expressions I have in mind consists of English sentences, with the connective b as a means of combining sentences into discourses (the intuitively unacceptable among which, $[\![\cdot]\!]$ must somehow mark out). For instance, e_1 might be the sentence

`A friend of every student in Doris's class loves squash`

which can be combined with the sentence e_2

*Computer Science, Trinity College, Dublin 2, Ireland. Email: `Tim.Fernando@tcd.ie`

```
He is the only kid in town who eats it raw
```

to form the discourse $e_1 e_2$, encoded in E_b as $b(e_1, e_2)$. Notice that whatever interpretations we may give to these examples, it is hard to deny that e_1 is, in isolation, ambiguous in ways that it is not within the larger discourse:

```
A friend of every student in Doris's class loves squash.
He is the only kid in town who eats it raw.
```

The same can, for that matter, be said about e_2, the more general point being that one of the possible effects of combining an expression with other expressions (to the left or right) is disambiguation (be it partial or total). While disambiguation by the surrounding text is hardly surprising, it leads to the following twist in determining the meaning of an expression e via (1). Rather than taking the meaning of an atomic expression for granted and reducing the meaning of an expression $b(e_1, e_2)$ to (1)'s right hand side (under some inductive assumption that the complexity there is less), the expression e whose meaning we wish to understand might be identified in (1) with either e_1 or e_2, on the chance that more is known about the left hand side, $[\![b(e_1, e_2)]\!]$, of (1) than about the right hand side components, $\beta, [\![e_1]\!]$ and $[\![e_2]\!]$. It is this co-inductive construal of (1), according to which meaning is uncovered by merging rather than (as in the inductive reading) by breaking apart, that is explored below. The thrust is to broaden the context, not to contract it.

An operational perspective on meaning $[\![\cdot]\!]$ is adopted, revolving around machines α, to which we can input any finite number n of expressions $e_1 \cdots e_n$ from E.[1] The term "machine" may, for concreteness, be understood recursion-theoretically (perhaps relativized to certain oracles, to transcend ordinary Turing machines), the essential point for now being to think of α as a translator, which, when fed $\vec{e} = e_1 \cdots e_n$, may spit back a sequence $\vec{\varphi} = \varphi_1 \cdots \varphi_n$ of logical forms (or well-formed formulas) translating $\vec{e}$. Non-determinism is allowed in that for any input sequence $\vec{e}$, the set of output sequences $\vec{\varphi}$ that α may return could be empty, a singleton, or neither. Collecting the logical forms into the set Φ, the input/output relation of α is a set

$$\mathsf{Stage}_\alpha \quad \subseteq \quad \bigcup_{n \geq 0} (E^n \times \Phi^n) \ ,$$

with the intuition that

$$(\vec{e}, \vec{\varphi}) \in \mathsf{Stage}_\alpha \qquad \text{iff} \qquad \text{on seeing } \vec{e}, \text{ the translator } \alpha \text{ may guess } \vec{\varphi} \ .$$

When we can get away with it, we will drop the subscript α on Stage. Elements of Stage are called *stages*, among which we will assume is the pair (ϵ, ϵ), built from the empty sequence ϵ, representing the initial stage of a translation process when nothing is seen, whence nothing guessed. The process may then evolve in any number of ways, as inputs from E are fed, one at a time, to α. At

[1]The assumption that such machines exist reflects the reality of language use (human and otherwise), independent of any particular theory of linguistic meaning. Reconstructing such machines is (nonetheless) the grand dream behind the present work.

this level of abstraction, there is clearly no principled reason to insist that E be a set of English sentences or that α even be in the natural language processing business. A bit more substance is introduced below by bringing out an entailment relation $\vdash$ on the logical forms, and coupling that with Stage. But confining our attention first to Stage, we may ask: is all the structure of Stage really necessary to analyze discourse interpretation? Can we not make do with less? Indeed, must we even distinguish E from Φ?

1.2 Outline

The preceding questions are taken up in §2, where a language $\mathcal{L}_\circ(E, \Phi)$ is constructed from logical forms $\varphi \in \Phi$ with modal operators $\langle e \rangle$ labeled by expressions $e \in E$. A particular class of models for $\mathcal{L}_\circ(E, \Phi)$ are singled out in §3, where $\langle e \rangle$ is interpreted by transitions that are in a precise sense inertial. Simple answers to what $[\![\cdot]\!]$ and β could be under (1) are proposed in §4, and extensions discussed in the concluding section.

2 Between E and Φ

Given an expression e ambiguous between two formulas φ and ψ in Φ, why not reduce e to the disjunction $\varphi \dot{\vee} \psi$, where Φ is assumed to be closed under $\dot{\vee}$?[2] Although this disjunctive approach to ambiguity has been widely criticized (e.g. van Deemter [2]), it serves as an instructive point of departure.

2.1 Componentwise approximations

For every expression $e \in E$, suppose we could form in Φ the disjunction $\dot{\bigvee}\delta(e) \in \Phi$ of the set $\delta(e) \subseteq \Phi$ of possible readings (in Φ) of e. In terms of the input/output set Stage described in the introduction, $\delta(e)$ can be equated with the set $\delta_{\mathsf{Stage}}(e)$ of logical forms given by

$$\delta_{\mathsf{Stage}}(e) \quad = \quad \{\varphi \in \Phi \ : \ (ler, \vec{\varphi}\varphi\vec{\psi}) \in \mathsf{Stage} \text{ for some } l, r \in E^* \text{ and } \vec{\varphi}, \vec{\psi} \in \Phi^* \text{ with length}(l) = \text{length}(\vec{\varphi})\} \ .$$

The function $\delta_{\mathsf{Stage}} : E \to Power(\Phi)$ considerably simplifies Stage, quantifying away, for every $e \in E$, the text l, r surrounding e (not to mention the translations $\vec{\varphi}$ and $\vec{\psi}$). This information is crucial in, for instance, examples e_1 and e_2 of §1.1. There is no recovering it through the further step

$$\delta_{\mathsf{Stage}}(e) \ \mapsto \ \dot{\bigvee}\delta_{\mathsf{Stage}}(e) \ ,$$

completing the proposed disjunctive analysis. From δ_{Stage}, Stage can be approximated only crudely by the set Stage^d given by

$$\bigcup_{n \geq 0} \ \ \{(e_1 \cdots e_n, \varphi_1 \cdots \varphi_n) \in E^n \times \Phi^n \ : \ \varphi_i \in \delta_{\mathsf{Stage}}(e_i) \text{ for } 1 \leq i \leq n\} \ .$$

[2]The dotted notation was suggested by a referee, to distinguish the connective from other kinds introduced below.

It is easy to see that the mapping $\cdot^d$ sending Stage to Stage^d (via δ_{Stage}) is a *closure operation* — i.e.,

Proposition 1.

(a) $\mathsf{Stage} \subseteq \mathsf{Stage}^d$.

(b) $\mathsf{Stage}^d = (\mathsf{Stage}^d)^d$ *(as* $\delta_{\mathsf{Stage}^d} = \delta_{\mathsf{Stage}}$*).*

(c) If $\mathsf{Stage}_1 \subseteq \mathsf{Stage}_2$ *then* $\mathsf{Stage}_1^d \subseteq \mathsf{Stage}_2^d$.

There is an adjunction in Proposition 1, which the reader versed in category theory has doubtless noticed. The same can be said about the following improved approximation of Stage. Factoring back in sequences l and $r \in E^*$ that surround an expression e, let us sharpen the set $\delta_{\mathsf{Stage}}(e)$ to the set $\gamma_{\mathsf{Stage}}(l, e, r)$ of logical forms φ to which e can be translated, given that l occurs to the left of e, and r to its right. That is, define the function $\gamma_{\mathsf{Stage}} : E^* \times E \times E^* \rightarrow Power(\Phi)$ by

$$\begin{aligned}\gamma_{\mathsf{Stage}}(l, e, r) \quad = \quad & \{\varphi \in \Phi \ : \ (ler, \vec{\varphi}\varphi\vec{\psi}) \in \mathsf{Stage} \text{ for some } \vec{\varphi}, \vec{\psi} \in \Phi^* \\ & \qquad \text{with length}(l) = \text{length}(\vec{\varphi})\} \ .\end{aligned}$$

From γ_{Stage}, Stage can be reconstructed as the set Stage^g given by

$$\begin{aligned}\bigcup_{n \geq 0} \{(e_1 \cdots e_n, \varphi_1 \cdots \varphi_n) \in E^n \times \Phi^n \quad : \quad & \varphi_i \in \gamma_{\mathsf{Stage}}(e_1 \cdots e_{i-1}, e_i, e_{i+1} \cdots e_n) \\ & \text{for } 1 \leq i \leq n\} \ .\end{aligned}$$

Proposition 2. $\cdot^g$ *is a closure operation, and* $\mathsf{Stage}^g \subseteq \mathsf{Stage}^d$.

Next, call Stage *deterministic* if for all $l, r \in E^*$ and $e \in E$,

$$|\gamma_{\mathsf{Stage}}(l, e, r)| \quad \leq \quad 1 \ .$$

Proposition 3. *If* Stage *is deterministic, then* $\mathsf{Stage}^g = \mathsf{Stage}$.

The question arises: is there any harm in ruling out overgeneration in Stage^g by stipulating that Stage be deterministic?

An example might help. Consider discourse $\hat{e}_1\hat{e}_2$:

$\hat{e}_1$: `Doris informed Chloe she aced the exam.`

$\hat{e}_2$: `In fact, she was the only Texan who passed.`

It is natural to expect the occurrences of the pronoun `she` in $\hat{e}_1$ and $\hat{e}_2$ to have an identical interpretation — that is, to *co-vary* — so that $\hat{e}_1\hat{e}_2$ can mean $e_1^D e_2^D$ or $e_1^C e_2^C$ but not, for example, $e_1^D e_2^C$.

$e_1^D e_2^D$: `Doris informed Chloe Doris aced the exam.`
`In fact, Doris was the only Texan who passed.`

$e_1^C e_2^C$: `Doris informed Chloe Chloe aced the exam.`
`In fact, Chloe was the only Texan who passed.`

$e_1^D e_2^C$: `Doris informed Chloe Doris aced the exam.`
`In fact, Chloe was the only Texan who passed.`

Accordingly, consider two translators $\alpha[D]$ and $\alpha[C]$ such that for all $\varphi, \varphi' \in \Phi$,

$$(\hat{e}_1\hat{e}_2, \varphi\varphi') \in \mathsf{Stage}_{\alpha[D]} \quad \text{iff} \quad (e_1^D e_2^D, \varphi\varphi') \in \mathsf{Stage}_{\alpha[D]}$$

and

$$(\hat{e}_1\hat{e}_2, \varphi\varphi') \in \mathsf{Stage}_{\alpha[C]} \quad \text{iff} \quad (e_1^C e_2^C, \varphi\varphi') \in \mathsf{Stage}_{\alpha[C]} \ .$$

Which translator is "correct" may depend on further inputs — for instance, on whether $\hat{e}_1\hat{e}_2$ is followed by $\hat{e}_3$ or by e_3'.

$\hat{e}_3$: `Chloe, a Californian, had not bothered to take the exam.`

e_3': `Doris graded the exams very harshly, and had hoped to flunk every American.`

Under a disjunctive approach to ambiguity, $\hat{e}_1$ would translate to $\varphi^C \dot{\vee} \varphi^D$, where $(\hat{e}_1, \varphi^C) \in \mathsf{Stage}_{\alpha[C]}$ and $(\hat{e}_1, \varphi^D) \in \mathsf{Stage}_{\alpha[D]}$. In general, given translators α and α', we could form the translator $\alpha\dot{\vee}\alpha'$ (overloading the symbol $\dot{\vee}$) such that for all $e \in E$ and $\psi \in \Phi$,

$$(e, \psi) \in \mathsf{Stage}_{\alpha\dot{\vee}\alpha'} \quad \text{iff} \quad (\exists \varphi, \varphi')\ (e, \varphi) \in \mathsf{Stage}_{\alpha}, (e, \varphi') \in \mathsf{Stage}_{\alpha'} \text{ and } \psi = \varphi\dot{\vee}\varphi' \ .$$

Specifying the outputs of $\alpha\dot{\vee}\alpha'$ to inputs $e_1 \cdots e_n$ becomes more challenging, however, for lengths $n > 1$, as illustrated already by co-variation in $\hat{e}_1\hat{e}_2$. Moving from $\dot{\bigvee}\delta_{\mathsf{Stage}}(e)$ to $\dot{\bigvee}\gamma_{\mathsf{Stage}}(e_1 \cdots e_{i-1}, e_i, e_{i+1} \cdots e_n)$, the problem with the clause

$$(\hat{e}_1\hat{e}_2, \psi_1\psi_2) \in \mathsf{Stage}_{\alpha[C]\dot{\vee}\alpha[D]} \quad \text{iff} \quad (\exists \varphi_1^C, \varphi_2^C, \varphi_1^D, \varphi_2^D) \ (\hat{e}_1\hat{e}_2, \varphi_1^C\varphi_2^C) \in \mathsf{Stage}_{\alpha[C]}, \ (\hat{e}_1\hat{e}_2, \varphi_1^D\varphi_2^D) \in \mathsf{Stage}_{\alpha[D]}, \ \psi_1 = \varphi_1^C\dot{\vee}\varphi_1^D, \text{ and } \psi_2 = \varphi_2^C\dot{\vee}\varphi_2^D$$

is that it fails to connect the translations in $\alpha[C]$ of $\hat{e}_1$ and $\hat{e}_2$ (or the translations in $\alpha[D]$ of $\hat{e}_1$ and $\hat{e}_2$). And matters only get worse when adding $\hat{e}_3$ or e_3' to $\hat{e}_1\hat{e}_2$. An alternative to $\alpha\dot{\vee}\alpha'$ is the translator $\alpha + \alpha'$ with input/output relation

$$\mathsf{Stage}_{\alpha+\alpha'} \ = \ \mathsf{Stage}_{\alpha} \cup \mathsf{Stage}_{\alpha'} \ ,$$

establishing the required connections by externalizing the indeterminacy of $\dot{\vee}$. With non-determinism slipping in behind $+$,

$$\mathsf{Stage}_{\alpha[C]+\alpha[D]} \ \neq \ (\mathsf{Stage}_{\alpha[C]+\alpha[D]})^g,$$

as $(\mathsf{Stage}_{\alpha[C]+\alpha[D]})^g$ fails to exclude the reading $e_1^D e_2^C$ of $\hat{e}_1\hat{e}_2$. But surely there is a standard technique for eliminating non-determinism that applies here?

2.2 The subset construction and disjunction

While, in general, neither δ_{Stage} nor γ_{Stage} may do justice to Stage, there is nothing lost in repackaging Stage as the function $\theta_{\mathsf{Stage}} : E^* \to Power(\Phi^*)$ defined for every $\vec{e} \in E^*$ by

$$\theta_{\mathsf{Stage}}(\vec{e}) \quad = \quad \{\vec{\varphi} \ : \ (\vec{e}, \vec{\varphi}) \in \mathsf{Stage}\}$$

(with Stage deterministic if

$$|\theta_{\mathsf{Stage}}(\vec{e})| \quad \leq \quad 1$$

for every $\vec{e} \in E^*$). This is the *subset construction* familiar from conversions of non-deterministic finite automata to deterministic finite automata accepting the same language (e.g. Hopcroft and Ullman [7]). To make the transformation to deterministic state-transitions explicit, let us associate with θ_{Stage} the set of "states"

$$E^*[\mathsf{Stage}] \quad = \quad \{\vec{e} \in E^* \ : \ \theta_{\mathsf{Stage}}(\vec{e}) \neq \emptyset\}$$

(i.e. the "acceptable" inputs) and, for every $e \in E$, the state-transition relation

$$R^e[\mathsf{Stage}] \quad = \quad \{(\vec{e}, \vec{e}e) \ : \ \vec{e} \in E^*[\mathsf{Stage}] \text{ and } \vec{e}e \in E^*[\mathsf{Stage}]\} \ .$$

Clearly, for every $e \in E$, $R^e[\mathsf{Stage}]$ is deterministic (i.e. a partial function on $E^*[\mathsf{Stage}]$).

Now, recalling that $\mathsf{Stage}_{\alpha+\alpha'} = \mathsf{Stage}_\alpha \cup \mathsf{Stage}_{\alpha'}$ and assuming that $\alpha \dot{\vee} \alpha'$ is deterministic, it is reasonable to suppose that

(i) for all $\vec{e} \in E^*[\mathsf{Stage}_{\alpha+\alpha'}]$, there is exactly one sequence $\vec{\varphi} \in \Phi^*$ such that $(\vec{e}, \vec{\varphi}) \in \mathsf{Stage}_{\alpha \dot{\vee} \alpha'}$

and conversely,

(ii) for every $(\vec{e}, \vec{\varphi}) \in \mathsf{Stage}_{\alpha \dot{\vee} \alpha'}$, $\vec{e} \in E^*[\mathsf{Stage}_{\alpha+\alpha'}]$.

The bijection $\pi : E^*[\mathsf{Stage}_{\alpha+\alpha'}] \to \mathsf{Stage}_{\alpha \dot{\vee} \alpha'}$ so defined yields for every $e \in E$, a copy $\stackrel{e}{\Rightarrow}$ of $R^e[\mathsf{Stage}_{\alpha+\alpha'}]$ in $\mathsf{Stage}_{\alpha \dot{\vee} \alpha'}$, given by

$$\stackrel{e}{\Rightarrow} \quad = \quad \{(\pi(\vec{e}), \pi(\vec{e}e)) \ : \ \vec{e} \in domain(R^e[\mathsf{Stage}_{\alpha+\alpha'}])\} \ .$$

That is to say, if we define an *E-Kripke frame* to be a pair $(W, \{\stackrel{e}{\to}\}_{e \in E})$ consisting of a set W of "worlds" and an E-indexed family of "e-accessibility relations" $\stackrel{e}{\to} \subseteq W^2$ (for $e \in E$), then π can, in fact, be construed as an isomorphism between the E-Kripke frames $(E^*[\mathsf{Stage}_{\alpha+\alpha'}], \{R^e[\mathsf{Stage}_{\alpha+\alpha'}]\}_{e \in E})$ and $(\mathsf{Stage}_{\alpha \dot{\vee} \alpha'}, \{\stackrel{e}{\Rightarrow}\}_{e \in E})$

$$(E^*[\mathsf{Stage}_{\alpha+\alpha'}], \{R^e[\mathsf{Stage}_{\alpha+\alpha'}]\}_{e \in E}) \quad \cong \quad (\mathsf{Stage}_{\alpha \dot{\vee} \alpha'}, \{\stackrel{e}{\Rightarrow}\}_{e \in E}) \ .$$

Having described the E-Kripke frame of $\alpha \dot{\vee} \alpha'$ up to isomorphism, the question remains: for every $\vec{e} \in E^*[\mathsf{Stage}]$, exactly what does $\alpha \dot{\vee} \alpha'$ translate $\vec{e}$ to? A partial answer is provided by specifying the set of logical forms in Φ entailed by the translation of $\vec{e}$ by $\alpha\dot{\vee}\alpha'$.

With that in mind, let us define an E, Φ-*couple* to be a pair $(\mathsf{Stage}, \vdash)$ such that $(\epsilon, \epsilon) \in \mathsf{Stage} \subseteq \bigcup_{n\geq 0}(E^n \times \Phi^n)$ and $\vdash\ \subseteq \Phi^* \times \Phi$, the intent being that

$$\vec{\varphi} \vdash \varphi \quad \text{can be read:} \quad \varphi \text{ is entailed (jointly) by } \vec{\varphi} .$$

Given an E, Φ-couple $(\mathsf{Stage}, \vdash)$, define the relation $\vdash^{\mathsf{Stage}} \subseteq E^*[\mathsf{Stage}] \times \Phi$ by

$$\vec{e} \vdash^{\mathsf{Stage}} \varphi \quad \text{iff} \quad \text{for every } \vec{\varphi} \in \theta_{\mathsf{Stage}}(\vec{e}),\ \vec{\varphi} \vdash \varphi$$

for every $\vec{e} \in E^*[\mathsf{Stage}]$ and $\varphi \in \Phi$. Inasmuch as we may expect that for all $\Psi \subseteq \Phi$ and $\varphi \in \Phi$,

$$\dot{\bigvee}\Psi \vdash \varphi \quad \text{iff} \quad \text{for every } \psi \in \Psi,\ \psi \vdash \varphi ,$$

we may stipulate that the translation of $\vec{e}$ by $\alpha\dot{\vee}\alpha'$ entails φ precisely when $\vec{e} \vdash^{\mathsf{Stage}_{\alpha+\alpha'}} \varphi$. The force of this stipulation will become clearer in §3.2, with the help of some modal logic.

2.3 The language $\mathcal{L}_\circ(E, \Phi)$

We have in the preceding paragraph described ingredients for interpreting modal formulas $\langle e\rangle\varphi$ so that

$$\langle e\rangle\varphi \quad \text{means:} \quad e \text{ can be translated to a logical form entailing } \varphi .$$

Proceeding more systematically, let $\mathcal{L}_\circ(E, \Phi)$, or $\mathcal{L}_\circ$ for short, be the set of $\mathcal{L}_\circ$-*formulas* A generated inductively from logical forms $\varphi \in \Phi$ via modal operators $\langle e\rangle$ (for every $e \in E$) plus the Boolean connectives $\neg$ and $\wedge$

$$A \quad ::= \quad \varphi \ \mid\ \langle e\rangle A \ \mid\ \neg A \ \mid\ A \wedge A .$$

It is important here not to confuse the $\mathcal{L}_\circ$-connectives $\wedge$ and $\neg$ with connectives in Φ such as $\dot{\vee}$ which are accordingly dotted. (All logical forms $\varphi \in \Phi$ are, as $\mathcal{L}_\circ$-formulas, atomic and share none of $\mathcal{L}_\circ$'s connectives.) Given an E, Φ-couple $(\mathsf{Stage}, \vdash)$, let $\Vdash_\theta \subseteq E^*[\mathsf{Stage}] \times \mathcal{L}_\circ$ be the extension of $\vdash^{\mathsf{Stage}}$ given by

$$\begin{aligned}
\vec{e} \Vdash_\theta \varphi \quad & \text{iff} \quad \vec{e} \vdash^{\mathsf{Stage}} \varphi \\
\vec{e} \Vdash_\theta \langle e\rangle A \quad & \text{iff} \quad (\exists \vec{e'} \in E^*[\mathsf{Stage}])\ \vec{e}\ R^e[\mathsf{Stage}]\ \vec{e'} \ \text{ and } \ \vec{e'} \Vdash_\theta A \\
& \text{iff} \quad \theta_{\mathsf{Stage}}(\vec{e}e) \neq \emptyset \ \text{ and } \ \vec{e}e \Vdash_\theta A \\
\vec{e} \Vdash_\theta \neg A \quad & \text{iff} \quad \text{not } \vec{e} \Vdash_\theta A \\
\vec{e} \Vdash_\theta A \wedge B \quad & \text{iff} \quad \vec{e} \Vdash_\theta A \ \text{ and } \ \vec{e} \Vdash_\theta B
\end{aligned} \tag{2}$$

for all $\vec{e} \in E^*[\mathsf{Stage}]$. (It is understood that for all $\vec{e} \notin E^*[\mathsf{Stage}]$ and all $A \in \mathcal{L}_\circ$, $\vec{e} \not\Vdash_\theta A$.) The predicate $\Vdash_\theta$ is exactly what Kripke semantics returns, given the E-Kripke frame $(E^*[\mathsf{Stage}], \{R^e[\mathsf{Stage}]\}_{e\in E})$ from θ_{Stage} and a valuation induced by $\vdash^{\mathsf{Stage}}$.

More about Stage and $\vdash$ are revealed by a predicate $\Vdash \subseteq \mathsf{Stage} \times \mathcal{L}_\circ$ interpreting $\mathcal{L}_\circ$-formulas against stages $(\vec{e}, \vec{\varphi}) \in \mathsf{Stage}$ (rather than, as in $\Vdash_\theta$, acceptable sequences $\vec{e}$) such that

$$(\vec{e}, \vec{\varphi}) \Vdash \varphi \quad \text{iff} \quad \vec{\varphi} \vdash \varphi$$

and the usual Boolean clauses

$$\begin{aligned} (\vec{e}, \vec{\varphi}) \Vdash \neg A \quad &\text{iff} \quad \text{not } (\vec{e}, \vec{\varphi}) \Vdash A \\ (\vec{e}, \vec{\varphi}) \Vdash A \wedge B \quad &\text{iff} \quad (\vec{e}, \vec{\varphi}) \Vdash A \text{ and } (\vec{e}, \vec{\varphi}) \Vdash B \ . \end{aligned}$$

As for $\langle e\rangle A$, a couple of possibilities (reflecting different e-accessibility relations on Stage) are

$$(\vec{e}, \vec{\varphi}) \Vdash \langle e\rangle A \quad \text{iff} \quad (\exists \varphi \in \Phi)\ (\vec{e}e, \vec{\varphi}\varphi) \in \mathsf{Stage} \text{ and } (\vec{e}e, \vec{\varphi}\varphi) \Vdash A$$

(investigated in Fernando [4]) and, adding a prime $'$ to $\langle e\rangle$ to distinguish it from the previous interpretation,

$$(\vec{e}, \vec{\varphi}) \Vdash \langle e\rangle' A \quad \text{iff} \quad (\exists \vec{\psi} \in \Phi^*)\ (\vec{e}e, \vec{\psi}) \in \mathsf{Stage} \text{ and } (\vec{e}e, \vec{\psi}) \Vdash A \ .$$

(The difference is that $\langle e\rangle'$ allows revision, elaborated on in §5.1.) Let us write $\mathcal{L}'$ for $\mathcal{L}_\circ$ with all modalities $\langle e\rangle$ replaced by primed modalities $\langle e\rangle'$, and $\mathcal{L}$ for the smallest set containing Φ that is closed under $\wedge, \neg, \langle e\rangle$ and $\langle e\rangle'$, for all $e \in E$. Notice that the relation $\Vdash$ defined above applies not just to $\mathcal{L}_\circ$ but to $\mathcal{L}$. To relate $\Vdash$ to $\Vdash_\theta$, it will be useful to extend $\Vdash$ further to formulas $\Box A$ saying roughly that A is forced by all translations of the inputs —

$$\begin{aligned} (\vec{e}, \vec{\varphi}) \Vdash \Box A \quad \text{iff} \quad &\text{for every } \vec{\psi} \in \Phi^* \text{ such that } (\vec{e}, \vec{\psi}) \in \mathsf{Stage}, \\ &(\vec{e}, \vec{\psi}) \Vdash A \end{aligned}$$

for every $(\vec{e}, \vec{\varphi}) \in \mathsf{Stage}$. Underlying $\Box$ is the accessibility relation $\sim$ on Stage such that

$$(\vec{e}, \vec{\varphi}) \sim (\vec{e'}, \vec{\varphi'}) \quad \text{iff} \quad \vec{e} = \vec{e'}$$

for all stages $(\vec{e}, \vec{\varphi})$ and $(\vec{e'}, \vec{\varphi'})$. Let $\mathcal{L}_\circ(\Box), \mathcal{L}'(\Box)$ and $\mathcal{L}(\Box)$ be the result of adding a closure condition for $\Box$ to the conditions defining $\mathcal{L}_\circ, \mathcal{L}'$ and $\mathcal{L}$, respectively. An $\mathcal{L}(\Box)$-formula A is Stage-*valid* if $(\vec{e}, \vec{\varphi}) \Vdash A$ for every $(\vec{e}, \vec{\varphi}) \in \mathsf{Stage}$. A question familiar from multi-dimensional modal logic (Marx and Venema [11]) is whether $\Box$ commutes with the other modalities. With this in mind, call Stage *prefix-closed* if whenever $(\vec{e}e, \vec{\varphi}\varphi) \in \mathsf{Stage}$, $(\vec{e}, \vec{\varphi}) \in \mathsf{Stage}$.

Proposition 4. *For every E, Φ-couple* $(\mathsf{Stage}, \vdash)$*, $\mathcal{L}(\Box)$-formula A and $e \in E$, the following $\mathcal{L}(\Box)$-formulas are* Stage*-valid:*

$$\begin{aligned} \langle e\rangle'\Diamond A \ &\equiv\ \Diamond\langle e\rangle' A \\ \langle e\rangle' A \ &\equiv\ \langle e\rangle'\Diamond A \\ \langle e\rangle\Diamond A \ &\supset\ \langle e\rangle' A \\ \Diamond\langle e\rangle A \ &\supset\ \langle e\rangle' A \\ \langle e\rangle\top \quad \supset \quad (\langle e\rangle\Diamond A \ &\equiv\ \langle e\rangle' A) \end{aligned}$$

where $\Diamond$ *abbreviates* $\neg\Box\neg$, $A \equiv B$ *abbreviates* $(A \supset B) \wedge (B \supset A)$, *and* $\top$ *is an* $\mathcal{L}_\circ$*-tautology such as* $\neg(\varphi \wedge \neg\varphi)$, *for any* $\varphi \in \Phi$. *Moreover,*

$$\Diamond\langle e\rangle A \;\equiv\; \langle e\rangle' A$$

is Stage*-valid, if* Stage *is prefix-closed.*

Next, given an $\mathcal{L}_\circ$-formula A, let $A^\Box$ be the $\mathcal{L}'(\Box)$-formula defined by

$$\begin{array}{rcl} \varphi^\Box & = & \Box\varphi \\ (\langle e\rangle A)^\Box & = & \langle e\rangle' A^\Box \\ (A \wedge B)^\Box & = & A^\Box \wedge B^\Box \\ (\neg A)^\Box & = & \neg A^\Box \; . \end{array}$$

Proposition 5. *For every* $\vec{e} \in E^*[\mathsf{Stage}]$ *and every* $\mathcal{L}_\circ$*-formula* A*, the following are equivalent:*

(i) $\vec{e} \mid\mid\!\!-_\theta A$

(ii) $(\vec{e}, \vec{\varphi}) \mid\mid\!\!- A^\Box$ *for every* $\vec{\varphi} \in \theta_{\mathsf{Stage}}(\vec{e})$

(iii) $(\vec{e}, \vec{\varphi}) \mid\mid\!\!- A^\Box$ *for some* $\vec{\varphi} \in \theta_{\mathsf{Stage}}(\vec{e})$.

In view of the latter part of Proposition 4, we can, assuming Stage is prefix-closed, establish Proposition 5 for a translation from $\mathcal{L}_\circ$-formulas A to $\mathcal{L}_\circ(\Box)$-formulas A°, defined as in $A^\Box$ except that $\langle e\rangle$ is translated as $\Diamond\langle e\rangle$

$$(\langle e\rangle A)^\circ \;=\; \Diamond\langle e\rangle A^\circ \; .$$

Furthermore, defining an $\mathcal{L}_\circ$-formula A to be θ_{Stage}*-valid* if $\vec{e} \mid\mid\!\!-_\theta A$ for every $\vec{e} \in E^*[\mathsf{Stage}]$,

Proposition 6. *For every* $\mathcal{L}_\circ$*-formula* A*, if* A *is* Stage*-valid then so is* A°*. Thus, if* Stage *is prefix-closed, then every* $\mathcal{L}_\circ$*-formula that is* Stage*-valid is* θ_{Stage}*-valid.*

Proposition 6 can be proved by induction on $\mathcal{L}_\circ$-formulas A, with the induction hypothesis strengthened by the additional clause: if A is forced by no stage, then neither is A°.

Propositions 5 and 6 hold also with both $\varphi^\Box$ and φ° redefined to be $\Diamond\varphi$, if the base clause (2) for $\mid\mid\!\!-_\theta$ is modified to

$$\vec{e} \mid\mid\!\!-_\theta \varphi \quad \text{iff} \quad \text{for } \textit{some } \vec{\varphi} \in \theta_{\mathsf{Stage}}(\vec{e}),\ \vec{\varphi} \vdash \varphi \; ,$$

corresponding to a more liberal disjunctive analysis of ambiguity. (In fact, we may replace “all” and “some” by a generalized quantifier Q, if we could form the generalization φ_Q of $\Box\varphi$ and $\Diamond\varphi$.)

3 Inertial interpretation and underspecification

First, a brief review of the previous section. Taking for granted a set E of expressions e and a set Φ of logical forms φ, we

(i) stepped from e (and δ_{Stage}) to sequences $e_1 \cdots e_n$ (and γ_{Stage}) to factor in the contribution of the surrounding text to the translation of e

and

(ii) paired $e_1 \cdots e_n$ with sequences $\varphi_1 \cdots \varphi_n$ to handle non-determinism, and interpret a set $\mathcal{L}_\circ$ of modal formulas.

In fact, given a notion Stage of translation and a notion $\vdash$ of entailment (i.e. an E, Φ-couple $(\mathsf{Stage}, \vdash)$), various interpretations of $\mathcal{L}_\circ(E, \Phi)$ were proposed, one of which is singled out in this section, and related to prefix closure, a disjunctive conception of ambiguity, and underspecification.

3.1 Prefix closure

Recall the sentences $\hat{e}_1, \hat{e}_2, \hat{e}_3$ and e'_3, and the translators $\alpha[C]$ and $\alpha[D]$ from §2.1. Given that $\alpha[C]$ reads $\hat{e}_1\hat{e}_2$ as $e_1^C e_2^C$, it is natural to expect the input $\hat{e}_1\hat{e}_2\hat{e}_3$ to defeat $\alpha[C]$. Unless, that is, on seeing $\hat{e}_3$, $\alpha[C]$ can recover from its interpretation of $\hat{e}_1\hat{e}_2$ and re-interpret it, within the larger input $\hat{e}_1\hat{e}_2\hat{e}_3$, as $e_1^D e_2^D$. But then shouldn't $\hat{e}_1\hat{e}_2$ also be read, in isolation, as $e_1^D e_2^D$? This is the case with the non-deterministic translator $\alpha[C]+\alpha[D]$, about which incidentally there is no need to talk of recovering from an interpretation of $\hat{e}_1\hat{e}_2$. On seeing $\hat{e}_1\hat{e}_2$, $\alpha[C]+\alpha[D]$ can be said to be entertaining two possible readings, only one of which should survive if $\hat{e}_3$ or e'_3 turns up next.

The preceding paragraph assumes that an input sequence $e_1 e_2 \cdots e_n$ is processed incrementally, beginning with sentence e_1, and then e_2, and so on up to e_n. This assumption holds for a good deal of work on discourse interpretation, including *Discourse Representation Theory* (DRT, Kamp and Reyle [8]).[3] Without insisting that every input/output set $\mathsf{Stage} \subseteq \bigcup_{n\geq 0}(E^n \times \Phi^n)$ arise in such a manner, let us carve out, for every expression $e \in E$, a binary relation $[e\rangle \subseteq \mathsf{Stage}^2$ containing all transitions triggered by the input e in which the previous translations persist:

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

[3] The logical forms φ in DRT are called *discourse representation structures* (DRSs), and are constructed in Kamp and Reyle [8] by an algorithm that translates e_i into a DRS φ_i incorporating the translations $\varphi_1 \cdots \varphi_{i-1}$ of the previous inputs $e_1 \cdots e_{i-1}$. More recent formulations of DRT (e.g. Groenendijk and Stokhof [5]) feature an explicit *merge* operation (e.g. relational composition), allowing a clear separation between the components φ_i (so that φ_i need not incorporate $\varphi_1 \cdots \varphi_{i-1}$). See also van Benthem [1] and Visser and Vermeulen [14].

The notion of incrementality here is very much in the spirit of Kempson et al [9], where an account is developed of how a sentence is interpreted as each word in the sentence is read from left to right. The present work proceeds at a somewhat different level, away from details of *sub*sentential structure (leaving the choice of syntactic formalism open).

for all $(\vec{e}, \vec{\varphi}), (\vec{e'}, \vec{\varphi'}) \in \mathsf{Stage}$. The relation $[e\rangle$ is the e-accessibility relation used in §2.3 to interpret the modality $\langle e\rangle$ relative to stages. Let us collect the stages $(e_1 \cdots e_n, \varphi_1 \cdots \varphi_n) \in \mathsf{Stage}$ supported by the chain of transitions

$$(\epsilon, \epsilon)\ [e_1\rangle\ (e_1, \varphi_1)\ [e_2\rangle\ (e_1 e_2, \varphi\varphi_2)\ [e_3\rangle\ \cdots\ [e_n\rangle\ (e_1 \cdots e_n, \varphi_1 \cdots \varphi_n)\ ,$$

together in the *prefix-closed fragment* $\mathsf{Stage}_\circ$

$$\mathsf{Stage}_\circ \;=\; \bigcup_{n \geq 0} \{(e_1 \cdots e_n, \varphi_1 \cdots \varphi_n) \in E^n \times \Phi^n \;:\; (e_1 \cdots e_i, \varphi_1 \cdots \varphi_i) \in \mathsf{Stage} \text{ for } 1 \leq i \leq n\}\ .$$

It is easy to see that if $||\!\!-_\circ$ is the forcing relation associated with the prefix-closed fragment $\mathsf{Stage}_\circ$ of Stage, then for every $\mathcal{L}_\circ$-formula A,

$$(\epsilon, \epsilon) \;||\!\!-\; A \quad \text{iff} \quad (\epsilon, \epsilon) \;||\!\!-_\circ\; A\ . \tag{3}$$

To push through an inductive proof of (3), it is useful to replace (ϵ, ϵ) in (3) by an arbitrary $(\vec{e}, \vec{\varphi}) \in \mathsf{Stage}_\circ$. Returning to (ϵ, ϵ), however, let us define

$$\mathsf{Stage} \;\equiv^{\mathcal{L}_\circ}\; \mathsf{Stage}'$$

to mean that for every relation $\vdash\ \subseteq \Phi^* \times \Phi$ and every $\mathcal{L}_\circ$-formula A,

$$(\epsilon, \epsilon) \;||\!\!-\; A \quad \text{iff} \quad (\epsilon, \epsilon) \;||\!\!-'\; A\ ,$$

where $||\!\!-$ is the forcing predicate of $(\mathsf{Stage}, \vdash)$ and $||\!\!-'$ is $(\mathsf{Stage}', \vdash)$'s. The language $\mathcal{L}_\circ$ captures prefix-closed fragments in the following sense.

Proposition 7. $\mathsf{Stage}_\circ = \mathsf{Stage}'_\circ$ *iff* $\mathsf{Stage} \equiv^{\mathcal{L}_\circ} \mathsf{Stage}'$.

The forward direction $\Rightarrow$ of Proposition 7 is a corollary of (3), while the converse drops out by choosing $\vdash$ such that for all $\varphi_1, \ldots, \varphi_n, \varphi \in \Phi$,

$$\varphi_1 \cdots \varphi_n \vdash \varphi \quad \text{iff} \quad \varphi = \varphi_n\ .$$

(In practice, of course, we will have other entailment relations in mind.)

A more symmetric restriction of Stage than $\mathsf{Stage}_\circ$ suggested by an anonymous referee is the fragment

$$\mathsf{Stage}_\bigtriangledown \;=\; \bigcup_{n \geq 0} \{(e_1 \cdots e_n, \varphi_1 \cdots \varphi_n) \in E^n \times \Phi^n \;:\; (e_h \cdots e_i, \varphi_h \cdots \varphi_i) \in \mathsf{Stage} \text{ for } 1 \leq h \leq i \leq n\}$$

which is closed also under suffixes. $\mathsf{Stage}_\bigtriangledown$ is the intersection of $\mathsf{Stage}_\circ$ with a similar fragment generated from (ϵ, ϵ) by transitions $\langle e]\ \subseteq \mathsf{Stage}^2$ that add e to the left (rather than to the right) of a stage:

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

for all $(\vec{e}, \vec{\varphi}), (\vec{e'}, \vec{\varphi'}) \in \mathsf{Stage}$. Insofar as speech is heard (and presumably interpreted) forwards in time, however, the direction $[e\rangle$ is prima facie more salient for discourse interpretation than $\langle e]$ — the possibility of playback being outside the norm (as is the leap of imagination in reading a book from back to front). While some inertial principle may plausibly be invoked to justify persistence in $[e\rangle$, it is conceivable that $\langle e]$ calls for an agile mind, rather than a sluggish one subject to inertia. Be that as it may, let us leave the direction $\langle e]$ for some other occasion.

3.2 From disjunction to underspecified representations

Next, consider the problem of inertially interpreting the subset construction θ. That is, given an E,Φ-couple $(\mathsf{Stage}, \vdash)$, find a set $\mathsf{Stage}^\theta \subseteq \bigcup_{n\geq 0}(E^n \times \Phi^n)$ such that the E,Φ-couple $(\mathsf{Stage}^\theta, \vdash)$ corresponds to θ_{Stage} in that

(c1) for every $\vec{e} \in E^*[\mathsf{Stage}]$, there exists exactly one sequence $\vec{\varphi'} \in \Phi^*$ such that $(\vec{e}, \vec{\varphi'}) \in \mathsf{Stage}^\theta$,

(c2) for every $(\vec{e}, \vec{\varphi'}) \in \mathsf{Stage}^\theta$, $\vec{e} \in E^*[\mathsf{Stage}]$

and, with an eye to the valuation induced by $\vdash^{\mathsf{Stage}}$,

(c3) for all $(\vec{e}, \vec{\varphi'}) \in \mathsf{Stage}^\theta$ and $\varphi \in \Phi$,

$$\vec{\varphi'} \vdash \varphi \quad \text{iff} \quad \vec{e} \vdash^{\mathsf{Stage}} \varphi \ .$$

Returning to the discourse $\hat{e}_1\hat{e}_2$ from §2.1, if the set Stage of $\alpha[C]+\alpha[D]$ satisfies

$$\begin{aligned} \theta_{\mathsf{Stage}}(\hat{e}_1) &= \{\varphi_1^C,\ \varphi_1^D\} \\ \theta_{\mathsf{Stage}}(\hat{e}_1\hat{e}_2) &= \{\varphi_1^C\varphi_2^C,\ \varphi_1^D\varphi_2^D\} \end{aligned}$$

then a candidate $\hat{\varphi}_1\hat{\varphi}_2$ for $(\hat{e}_1\hat{e}_2, \hat{\varphi}_1\hat{\varphi}_2) \in \mathsf{Stage}^\theta$ is

$$\begin{aligned} \hat{\varphi}_1 &= \varphi_1^C \dot{\vee} \varphi_1^D \\ \hat{\varphi}_2 &= (\varphi_1^C \dot{\wedge} \varphi_2^C)\ \dot{\vee}\ (\varphi_1^D \dot{\wedge} \varphi_2^D) \ . \end{aligned}$$

More generally, assuming readings $\vec{\varphi} \in \theta_{\mathsf{Stage}}(\vec{e})$ can be merged into single logical forms $\dot{\bigwedge}\vec{\varphi}$ such that for every $\varphi \in \Phi$,

$$\vec{\varphi} \vdash \varphi \quad \text{iff} \quad (\dot{\bigwedge}\vec{\varphi}) \vdash \varphi$$

and assuming disjunctions $\dot{\bigvee}\Psi$ of subsets $\Psi \subseteq \Phi$ can be formed such that (as in §2.2)

$$(\dot{\bigvee}\Psi) \vdash \varphi \quad \text{iff} \quad \text{for every } \psi \in \Psi,\ \psi \vdash \varphi \ ,$$

a likely candidate for Stage^θ is the set of pairs $(e_1 \cdots e_n, \varphi'_1 \cdots \varphi'_n) \in E^n \times \Phi^n$, for $n \geq 0$, such that $e_1 \cdots e_n \in E^*[\mathsf{Stage}]$ and for $1 \leq i \leq n$,

$$\varphi'_i \quad = \quad \dot{\bigvee}\ \{\dot{\bigwedge}\varphi_1 \cdots \varphi_i : \varphi_1 \cdots \varphi_i \in \theta_{\mathsf{Stage}}(e_1 \cdots e_i)\} \ . \tag{4}$$

Conditions (c1) and (c2) above hold for all functions $\dot{\bigwedge} : \Phi^* \rightarrow \Phi$ and $\dot{\bigvee} : \mathit{Power}(\Phi) \rightarrow \Phi$. As for (c3), it helps if $\vdash$ validates properties such as (Cut) and (LWeak)

$$\text{(Cut)}\ \frac{\varphi' \vdash \chi \qquad \chi\varphi' \vdash \varphi}{\varphi' \vdash \varphi} \qquad\qquad \text{(LWeak)}\ \frac{\varphi' \vdash \varphi}{\chi\varphi' \vdash \varphi}$$

and treats $\dot{\bigwedge}$ and $\dot{\bigvee}$ enough like classical conjunction and disjunction. I leave to the interested reader a specification of assumptions under which (c3) is satisfied, turning instead to certain problems with (4).

Notice that for $\varphi'_1 \cdots \varphi'_n$ given by (4), the expectation is that

$$\varphi'_1 \cdots \varphi'_n \vdash \varphi \quad \text{iff} \quad \varphi'_n \vdash \varphi$$

for all $\varphi \in \Phi$, with $\varphi'_1 \cdots \varphi'_{n-1}$ incorporated already in φ'_n. Were we to keep $\varphi'_1 \cdots \varphi'_{n-1}$ out of φ'_n, disjunction may break the necessary connections, as illustrated by $\hat{e}_1\hat{e}_2$ (the danger being that $\alpha[C]$'s translation $\varphi^C_1\varphi^C_2$ might mix with $\alpha[D]$'s translation $\varphi^D_1\varphi^D_2$ to yield say, $\varphi^C_1\varphi^D_2$). It would appear that (4) is incompatible with the methodological tendency to (a) focus on sufficiently disambiguated (albeit idealized) readings and (b) maximize the separation between the contributions of the different components e_i of $e_1 \cdots e_n$.[4] It is far from clear if anything like (4) or the entire set $\theta_{\mathsf{Stage}}(\vec{e})$ of readings of $\vec{e}$ can be feasibly computed (in theory or practice) — to say nothing of cognitive implausibility, a widely held view being that humans jump to particular readings from which they may need to backtrack.

An alternative to (4) is to design some set U of "unspecified representations" together with a map $\delta^U : U \to Power(\Phi^*)$, decomposing Stage in terms of some translation $\theta^U_{\mathsf{Stage}} : E^* \to U$ such that for all $(\vec{e}, \vec{\varphi}) \in E^* \times \Phi^*$,

$$(\vec{e}, \vec{\varphi}) \in \mathsf{Stage} \quad \text{iff} \quad \vec{\varphi} \in \delta^U(\theta^U_{\mathsf{Stage}}(\vec{e})) \ .$$

By making δ^U independent of Stage, we rule out the "lazy" solution given by $U = E^*$, and $\theta^U_{\mathsf{Stage}}$ equal to the identity function on E^*. At the other extreme, for U to be *under*specified (as opposed to merely unspecified), δ^U cannot be simply equated with the identity function on $Power(\Phi)$, the point of underspecification being to avoid computing θ_{Stage} (which setting δ^U to the identity on $Power(\Phi)$ would require). The idea is that underspecified representations are outputs of a partial translation $\theta^U_{\mathsf{Stage}}$ which feed into a further translation process δ^U that brings out the non-determinism stored by $\theta^U_{\mathsf{Stage}}$. (The slogan, if you will, is "underspecification as partial evaluation before non-determinism.") Relaxing the determinism of $\theta^U_{\mathsf{Stage}}$ and fixing some set $\mathsf{Stage}^U \subseteq \bigcup_{n \geq 0}(E^n \times \Phi^n)$, we might approximate Stage instead by a set Stage' such that $\mathsf{Stage} = \mathsf{Stage}' \circ \mathsf{Stage}^U$ — i.e.

$$(\vec{e}, \vec{\varphi}) \in \mathsf{Stage} \quad \text{iff} \quad (\exists \vec{u}) \ (\vec{e}, \vec{u}) \in \mathsf{Stage}' \text{ and } (\vec{u}, \vec{\varphi}) \in \mathsf{Stage}^U$$

for all $(\vec{e}, \vec{\varphi}) \in E^* \times \Phi^*$ — for which it is useful to assume $\Phi \subseteq E$ (so as to allow outputs to be fed as inputs), a move we will return to in §5.2. For now, note

[4] See the footnote in §3.1 for more on (b). As for (a), note that the DRS construction algorithm in Kamp and Reyle [8] deals with ambiguity through non-determinism, and not by consulting the whole set $\theta_{\mathsf{Stage}}(\vec{e})$ or by forming some disjunction such as (4).

that given some relation $\vdash \subseteq \Phi^* \times \Phi$, we can bypass Stage^U by requiring only that for every $\mathcal{L}_\circ$-formula A and every $\vec{e} \in E^*$,

$$\vec{e} \mid\!\vdash_\theta A \quad \text{iff} \quad \vec{e} \mid\!\vdash'_\theta A \tag{5}$$

(with $\mid\!\vdash_\theta \subseteq E^*[\mathsf{Stage}] \times \mathcal{L}_\circ$ and $\mid\!\vdash'_\theta \subseteq E^*[\mathsf{Stage}'] \times \mathcal{L}_\circ$). Restricting the $\mathcal{L}_\circ$-formulas A in (5) allows Stage' to fall further short of θ_{Stage}.

4 Sequence meaning

Next, we turn to the question of what a sequence $\vec{e}$ of expressions means. Ambiguity aside, there is a well-known problem in reducing the meaning of an expression e to the entailments of its logical form φ, illustrated by discourses such as the following, due to Barbara Partee.

> `Exactly one of the ten balls is not in the bag. It is under the sofa.`
>
> `Exactly nine of the ten balls are in the bag.` $^?$`It is under the sofa.`

Assuming the first sentences of the two sequences above translate to logical forms that participate in exactly the same $\vdash$-entailments, the difference in acceptability suggests that logical equivalence induced by $\vdash$ must be refined for a discourse congruence (on sequences).

Proceeding more precisely, let us focus on synonymy (given by some notion of meaning of expressions), with an eye to the problem (1) of compositionality. Call an equivalence relation $\approx$ on E^* an *s-congruence* if for all $\vec{e}, \vec{e'}, \vec{e''}, \vec{e'''} \in E^*$,

$$\vec{e} \approx \vec{e'} \text{ and } \vec{e''} \approx \vec{e'''} \quad \text{imply} \quad \vec{e}\vec{e''} \approx \vec{e'}\vec{e'''} \ .$$

From an s-congruence, we can extract a solution to compositionality (1) by flattening the set E_b of terms under the function $\cdot_\flat : E_b \to E^*$ mapping $t \in E_b$ to

$$t_\flat \quad = \quad \begin{cases} t & \text{if } t \in E \\ t'_\flat t''_\flat & \text{otherwise, where } t = b(t', t''). \end{cases}$$

Proposition 8. *Given an s-congruence $\approx$, define $[\![\cdot]\!]_\approx : E_b \to Power(E^*)$ by*

$$[\![t]\!]_\approx \quad = \quad \{\vec{e} \ : \ t_\flat \approx \vec{e}\}$$

and $\beta_\approx : Power(E^*)^2 \to Power(E^*)$ by

$$\beta_\approx(x, x') \quad = \quad \{\vec{e''} \ : \ (\exists \vec{e} \in x)(\exists \vec{e'} \in x') \ \vec{e''} \approx \vec{e}\vec{e'}\} \ .$$

Then for all $t, t' \in E_b$,

$$[\![b(t, t')]\!]_\approx \quad = \quad \beta_\approx([\![t]\!]_\approx, [\![t']\!]_\approx)$$

(*as* $[\![\cdot]\!]_{\approx}$ *is a composition of homomorphisms), and moreover, for all* $t'' \in E_b$,

$$\beta_{\approx}([\![t]\!]_{\approx}, \beta_{\approx}([\![t']\!]_{\approx}, [\![t'']\!]_{\approx})) \quad = \quad \beta_{\approx}(\beta_{\approx}([\![t]\!]_{\approx}, [\![t']\!]_{\approx}), [\![t'']\!]_{\approx})$$

(since $b(t, b(t', t''))_\flat = b(b(t, t'), t'')_\flat$*).*

Now, the plan of the present section is to isolate s-congruences that bring $\vdash$ into the picture. Progressively more fine-grained distinctions are drawn as more structure is considered:[5] from the subset construction θ to an externalization of the non-determinism θ buries, and finally to sequences $\vec{e}$ as binary relations on stages (modulo a certain well-known equivalence spelled out below).

4.1 S-congruences from $\mathcal{L}_\circ$

Given an E, Φ-couple $(\mathsf{Stage}, \vdash)$, let $\approx_\theta$ be the equivalence relation on E^* induced by the subset construction θ via $|\!\vdash_\theta$

$$\begin{array}{lll} e_1 \cdots e_n \approx_\theta e'_1 \cdots e'_{n'} & \text{iff} & \text{for all } A \in \mathcal{L}_\circ, \\ & & \quad \langle e_1 \rangle \cdots \langle e_n \rangle A \;\equiv\; \langle e'_1 \rangle \cdots \langle e'_{n'} \rangle A \\ & & \text{is } \theta_{\mathsf{Stage}}\text{-valid} \end{array}$$

for all $n, n' \geq 0$ and $e_1, \ldots, e_n, e'_1, \ldots, e'_{n'} \in E$ (agreeing that if $\vec{e} \notin E^*[\mathsf{Stage}]$, then $\vec{e} \not|\!\vdash_\theta A$ for all A). Similarly, replacing $|\!\vdash_\theta$ by $|\!\vdash$, let $\approx_\circ$ be the equivalence relation on E^* given by

$$\begin{array}{lll} e_1 \cdots e_n \approx_\circ e'_1 \cdots e'_{n'} & \text{iff} & \text{for all } A \in \mathcal{L}_\circ, \\ & & \quad \langle e_1 \rangle \cdots \langle e_n \rangle A \;\equiv\; \langle e'_1 \rangle \cdots \langle e'_{n'} \rangle A \\ & & \text{is } \mathsf{Stage}\text{-valid} \end{array}$$

(again agreeing that for all $(\vec{e}, \vec{\varphi}) \notin \mathsf{Stage}$, $(\vec{e}, \vec{\varphi}) \not|\!\vdash A$). For all $e, e' \in E$, define $\sim_{ee'}$ to be the binary relation on $\mathcal{L}_\circ$-formulas A and A' such that

$$\begin{array}{lll} A \sim_{ee'} A' & \text{iff} & \text{after replacing occurrences of } e \text{ by } e', \\ & & A \text{ and } A' \text{ are identical }. \end{array}$$

Proposition 9. *Fix an* E, Φ*-couple* $(\mathsf{Stage}, \vdash)$.

(a) Both $\approx_\theta$ *and* $\approx_\circ$ *are s-congruences.*

(b) For all $e, e' \in E$,

$$\begin{array}{lll} e \approx_\theta e' & \textit{iff} & \textit{for all } A \in \mathcal{L}_\circ \textit{ and } A' \sim_{ee'} A, \\ & & A \equiv A' \textit{ is } \theta_{\mathsf{Stage}}\textit{-valid} \end{array}$$

and similarly for $\approx_\circ$.

[5] The progressive refinements of equivalences parallel the co-inductive construction of the coarsest congruence refining an equivalence relation (Theorem 6 in Fernando [3], p. 594), the difference here being that explicit definitions of (several) congruences are given.

Examples such as Partee's above complicate extensions in the sense of Hodges [6] (i.e. preserving previously defined equivalences) of logical ($\vdash$-)equivalence on Φ to E (assumed, for convenience, to include Φ).

(c) *Assuming* Stage *is prefix-closed,*

$$\vec{e} \approx_\theta \vec{e'} \quad \textit{iff} \quad \textit{for all } A \in \mathcal{L}_\circ \textit{ and } \vec{e''} \in E^*, \\ \vec{e''}\vec{e} \mid\!\vdash_\theta A \;\; \textit{iff} \;\; \vec{e''}\vec{e'} \mid\!\vdash_\theta A$$

for all $\vec{e}, \vec{e'} \in E^*$.

(d) *If* Stage *is prefix-closed, then* $\approx_\circ \subseteq \approx_\theta$.

Part (a) follows from a chain of equivalences

$$\begin{aligned} \langle e_1\rangle \cdots \langle e_n\rangle\langle e''_1\rangle \cdots \langle e''_{n''}\rangle A &\equiv \langle e_1\rangle \cdots \langle e_n\rangle\langle e'''_1\rangle \cdots \langle e'''_{n'''}\rangle A \\ &\qquad \text{assuming } e''_1 \cdots e''_{n''} \approx e'''_1 \cdots e'''_{n'''} \\ &\equiv \langle e'_1\rangle \cdots \langle e'_{n'}\rangle\langle e'''_1\rangle \cdots \langle e'''_{n'''}\rangle A \\ &\qquad \text{assuming } e_1 \cdots e_n \approx e'_1 \cdots e'_{n'} \end{aligned}$$

for $\approx$ equal to $\approx_\theta$ or $\approx_\circ$. Part (b) can be proved by a routine induction on $\mathcal{L}_\circ$-formulas. Part (c) is immediate, assuming Stage is prefix-closed, while part (d) is a direct corollary of Proposition 6.

Prefix-closed translation sets Stage in which $\approx_\circ$ differs from $\approx_\theta$ are easy to construct.

(**E1**) Fix $e_0, e, e' \in E - \Phi$, and $\varphi_0, \varphi \in \Phi$, and let Stage contain the pairs

$$(e_0, \varphi_0), \;\; (e_0 e, \varphi_0\varphi), \;\; (e_0, \psi_0) \text{ and } (e_0 e', \psi_0\varphi)$$

where ψ_0 is $\varphi_0 \dot{\wedge} \varphi_0$. Then it is trivial to complete the definition of Stage (and $\vdash$) so that $e \approx_\theta e'$, but $(e_0, \varphi_0) \mid\!\vdash \langle e\rangle\varphi$, while $(e_0, \varphi_0) \not\mid\!\vdash \langle e'\rangle\varphi$.

4.2 An s-congruence from stages

Turning now from $\mathcal{L}_\circ$-formulas A to stages classified by A via $\mid\!\vdash$, let $\underline{\leftrightarrow}_0$ be the equivalence on Stage given by

$$(\vec{e}, \vec{\varphi}) \;\underline{\leftrightarrow}_0\; (\vec{e'}, \vec{\varphi'}) \quad \text{iff} \quad (\forall \varphi \in \Phi)\; \vec{\varphi} \vdash \varphi \text{ iff } \vec{\varphi'} \vdash \varphi$$

for all $(\vec{e}, \vec{\varphi}), (\vec{e'}, \vec{\varphi'}) \in$ Stage (the understanding being that $\underline{\leftrightarrow}_0 \subseteq \mathsf{Stage}^2$). Next, define the binary relation $\rightleftharpoons_0$ on E such that

$$e \rightleftharpoons_0 e' \quad \text{iff} \quad (\forall (\vec{e}, \vec{\varphi}) \in \mathsf{Stage})(\forall \vec{e'} \sim_{ee'} \vec{e})(\exists \vec{\varphi'} \in \Phi^*)\; (\vec{e}, \vec{\varphi}) \;\underline{\leftrightarrow}_0\; (\vec{e'}, \vec{\varphi'})$$

for all expressions $e, e' \in E$, where $\sim_{ee'}$ is the binary relation on E^* such that

$$\vec{e} \sim_{ee'} \vec{e'} \quad \text{iff} \quad \text{after replacing occurrences of } e \text{ by } e', \\ \vec{e} \text{ and } \vec{e'} \text{ are identical}$$

for all $\vec{e}, \vec{e'} \in E^*$. To see that $\rightleftharpoons_0$ is an equivalence relation on E, the interesting bit to check is that $\rightleftharpoons_0$ is transitive:

Suppose $e \rightleftharpoons_0 e'$ and $e' \rightleftharpoons_0 e''$. Given $(\vec{e}, \vec{\varphi}) \in \mathsf{Stage}$ and $\vec{e''} \sim_{ee''} \vec{e}$, we have to produce a logical form $\vec{\varphi''}$ such that $(\vec{e}, \vec{\varphi}) \underline{\leftrightarrow}_0 (\vec{e''}, \vec{\varphi''})$. With that in mind, let $\vec{e'}$ be $\vec{e}$ with all occurrences of e replaced by e', and let $\vec{e'''}$ be $\vec{e''}$ also with all occurrences of e replaced by e'. Now, replacing all occurrences of e' by e'' in $\vec{e'}$ gives the same sequence, call it $\vec{d}$, as replacing all occurrences of e' by e'' in $\vec{e'''}$. The required logical form $\vec{\varphi''}$ can be formed along the path

$$\vec{e} \ \sim_{ee'} \ \vec{e'} \ \sim_{e'e''} \ \vec{d} \ \sim_{e'e''} \ \vec{e'''} \ \sim_{ee'} \ \vec{e''} \ ,$$

using the assumption that $e \rightleftharpoons_0 e'$ and $e' \rightleftharpoons_0 e''$, plus the transitivity of $\underline{\leftrightarrow}_0$.

A counter-example to

$$e \approx_\circ e' \quad \text{implies} \quad e \rightleftharpoons_0 e'$$

can be obtained with $\Phi, \vdash$ given by classical propositional logic:

(**E2**) Fix $e, e' \in E - \Phi$, and let $p_0, p_1, p_2, \ldots$ be an infinite list of distinct proposition variables. Assume that for all $\varphi \in \Phi$,

$$(e, \varphi) \in \mathsf{Stage} \quad \text{iff} \quad \varphi \in \{p_0 \dot{\wedge} p_0,\ p_0 \dot{\wedge} p_1,\ p_0 \dot{\wedge} p_2,\ \ldots\}$$

and

$$(e', \varphi) \in \mathsf{Stage} \quad \text{iff} \quad \varphi \in \{p_0 \dot{\wedge} p_1,\ p_0 \dot{\wedge} p_2,\ p_0 \dot{\wedge} p_3,\ \ldots\} \ .$$

It is easy to arrange $e \approx_\circ e'$, and that there be no φ' such that $(e, p_0 \dot{\wedge} p_0)$ and (e', φ') force the same proposition variables p_i.

Nor does the converse, $e \rightleftharpoons_0 e'$ implies $e \approx_\circ e'$, hold, as shown by (**E1**) above. An obvious repair is to strengthen $\rightleftharpoons_0$ to $\rightleftharpoons_0^+$, defined on expressions e_1 and e_2 by

$$\begin{aligned} e_1 \rightleftharpoons_0^+ e_2 \quad \text{iff} \quad & (\forall (\vec{e}, \vec{\varphi}) \in \mathsf{Stage})(\forall \vec{e'} \in E^*)(\forall \vec{e''} \sim_{e_1 e_2} \vec{e'}) \\ & (\forall \vec{\varphi'} \in \Phi^* \text{ such that } (\vec{e}\vec{e'}, \vec{\varphi}\vec{\varphi'}) \in \mathsf{Stage}) \\ & (\exists \vec{\varphi''} \in \Phi^*) \ \ (\vec{e}\vec{e'}, \vec{\varphi}\vec{\varphi'}) \ \underline{\leftrightarrow}_0 \ (\vec{e}\vec{e''}, \vec{\varphi}\vec{\varphi''}) \ . \end{aligned}$$

Now, for all $e, e' \in E$,

$$e \rightleftharpoons_0^+ e' \quad \text{implies} \quad e \rightleftharpoons_0 e' \text{ and } e \approx_\circ e' \ .$$

(The conclusion $e \rightleftharpoons_0 e'$ is immediate; as for $e \approx_\circ e'$, argue by induction on A that whenever $e_1 \cdots e_n \in E^n$ and $e'_1 \cdots e'_n \sim_{ee'} e_1 \cdots e_n$ with $e \rightleftharpoons_0^+ e'$, the $\mathcal{L}_\circ$-formula

$$\langle e_1 \rangle \cdots \langle e_n \rangle A \ \equiv \ \langle e'_1 \rangle \cdots \langle e'_n \rangle A$$

is Stage-valid.) What about the converse: is $\rightleftharpoons_0^+$ just the intersection of the two equivalences? An elaboration of (**E2**) and (**E1**) settles the question negatively.

(**E3**) Fix distinct $e_0, e, e' \in E - \Phi$, and let $p_0, p_1, p_2, \ldots$ be an infinite list of distinct proposition variables. Assume that for all $\varphi_0, \varphi \in \Phi$,

$$
\begin{array}{rcl}
(e_0, \varphi_0) \in \mathsf{Stage} & \text{iff} & \varphi_0 \in \{p_0,\ p_0 \dot{\wedge} p_0\} \\
(e_0 e, \varphi_0 \varphi) \in \mathsf{Stage} & \text{iff} & (\varphi_0 \text{ is } p_0 \text{ and } \varphi \in \{p_0, p_1, p_2, \ldots\}) \\
 & & \text{or } \ (\varphi_0 \text{ is } p_0 \dot{\wedge} p_0 \text{ and } \varphi \text{ is } p_0)
\end{array}
$$

and

$$
\begin{array}{rcl}
(e_0 e', \varphi_0 \varphi) \in \mathsf{Stage} & \text{iff} & (\varphi_0 \text{ is } p_0 \text{ and } \varphi \in \{p_1, p_2, p_3, \ldots\}) \\
 & & \text{or } \ (\varphi_0 \text{ is } p_0 \dot{\wedge} p_0 \text{ and } \varphi \text{ is } p_0)\ .
\end{array}
$$

While $e \rightleftharpoons_0 e'$ and $e \approx_\circ e'$, there is no φ such that $(e_0 e', p_0 \varphi)$ and $(e_0 e, p_0 p_0)$ force the same proposition letters.

The equivalence $\underline{\leftrightarrow}_0$ is essentially insensitive to the structure on stages given by the transitions $[e\rangle$. To correct this deficiency, set for every $n \geq 0$,

$$
\underline{\leftrightarrow}_{n+1} \quad = \quad (\underline{\leftrightarrow}_n)^{bf}
$$

where, for every binary relation $R \subseteq (E^* \times \Phi^*)^2$, R^{bf} is defined to be the set of pairs $((\vec{e}, \vec{\varphi}), (\vec{e'}, \vec{\varphi'})) \in R$ such that for every $e \in E$,

("back") for every $\varphi' \in \Phi$ such that $(\vec{e'} e, \vec{\varphi'} \varphi') \in \mathsf{Stage}$, there exists $\varphi \in \Phi$ such that $(\vec{e} e, \vec{\varphi} \varphi) \in \mathsf{Stage}$ and $(\vec{e} e, \vec{\varphi} \varphi)\ R\ (\vec{e'} e, \vec{\varphi'} \varphi')$

and

("forth") for every $\varphi \in \Phi$ such that $(\vec{e} e, \vec{\varphi} \varphi) \in \mathsf{Stage}$, there exists $\varphi' \in \Phi$ such that $(\vec{e'} e, \vec{\varphi'} \varphi') \in \mathsf{Stage}$ and $(\vec{e} e, \vec{\varphi} \varphi)\ R\ (\vec{e'} e, \vec{\varphi'} \varphi')$.

A relation R contained in $\underline{\leftrightarrow}_0$ is a *bisimulation* (Park [12]) if $R \subseteq R^{bf}$. The chain

$$
\underline{\leftrightarrow}_0 \ \supseteq \ \underline{\leftrightarrow}_1 \ \supseteq \ \underline{\leftrightarrow}_2 \ \supseteq \cdots
$$

extends to $\underline{\leftrightarrow}_\alpha$ for all ordinals $\alpha > 0$ by setting

$$
\underline{\leftrightarrow}_\alpha \quad = \quad \bigcap_{\beta < \alpha} \underline{\leftrightarrow}_\beta{}^{bf}\ .
$$

It is well-known (and straightforward to prove) that

(a) every bisimulation is a subset of $\underline{\leftrightarrow}_\alpha$ for every ordinal α

(b) there is an ordinal α for which $\underline{\leftrightarrow}_\alpha = \underline{\leftrightarrow}_{\alpha+1}$ $(= \bigcap_\beta \underline{\leftrightarrow}_\beta = \underline{\leftrightarrow}_\gamma$ for all $\gamma > \alpha)$, and moreover, $\underline{\leftrightarrow}_\alpha$ is the $\subseteq$-largest bisimulation

(c) for all $(\vec{e}, \vec{\varphi}), (\vec{e'}, \vec{\varphi'}) \in E^* \times \Phi^*$,

$$
(\vec{e}, \vec{\varphi})\ \underline{\leftrightarrow}_\omega\ (\vec{e'}, \vec{\varphi'}) \quad \text{iff} \quad (\forall A \in \mathcal{L}_\circ)\ (\vec{e}, \vec{\varphi}) \Vdash A \text{ iff } (\vec{e'}, \vec{\varphi'}) \Vdash A
$$

and

(d) the least ordinal α witnessing (b) may well be larger than ω (so long as for some $(\vec{e}, \vec{\varphi}) \in \mathsf{Stage}$, $e \in E$, there are infinitely many $\varphi \in \Phi$ such that $(\vec{e}e, \vec{\varphi}\varphi) \in \mathsf{Stage}$).

The relation $\bigcap_\beta \underline{\leftrightarrow}_\beta$ mentioned in (b) is often called *bisimilarity*, and denoted $\underline{\leftrightarrow}$. Generalizing $\rightleftharpoons_0^+$ to ordinals α, define the equivalence $\rightleftharpoons_\alpha^+$

$$e_1 \rightleftharpoons_\alpha^+ e_2 \quad \text{iff} \quad \begin{array}{l} (\forall (\vec{e}, \vec{\varphi}) \in \mathsf{Stage})(\forall \vec{e'} \in E^*)(\forall \vec{e''} \sim_{e_1 e_2} \vec{e'}), \\ (\forall \vec{\varphi'} \in \Phi^* \text{ such that } (\vec{e}\vec{e'}, \vec{\varphi}\vec{\varphi'}) \in \mathsf{Stage}) \\ (\exists \vec{\varphi''} \in \Phi^*) \;\; (\vec{e}\vec{e'}, \vec{\varphi}\vec{\varphi'}) \underline{\leftrightarrow}_\alpha (\vec{e}\vec{e''}, \vec{\varphi}\vec{\varphi''}) \end{array}$$

on all expressions e_1 and e_2. For all $e_1 \cdots e_n, e'_1 \cdots e'_{n'} \in E^*$, let

$$e_1 \cdots e_n \rightleftharpoons^\circ e'_1 \cdots e'_{n'} \quad \text{iff} \quad \begin{array}{l} [e_1\rangle \circ \cdots \circ [e_n\rangle \text{ and } [e'_1\rangle \circ \cdots \circ [e'_{n'}\rangle \\ \text{"relate the same stages modulo } \underline{\leftrightarrow}\text{"} \end{array}$$

where the right hand side means that for all $(\vec{e}, \vec{\varphi}) \in \mathsf{Stage}$,

(i) for all $\varphi_1 \cdots \varphi_n$ such that for $1 \leq i \leq n$, $(\vec{e}e_1 \cdots e_i, \vec{\varphi}\varphi_1 \cdots \varphi_i) \in \mathsf{Stage}$, there exist $\varphi'_1 \cdots \varphi'_{n'}$ such that

$$(\vec{e}e_1 \cdots e_n, \vec{\varphi}\varphi_1 \cdots \varphi_n) \underline{\leftrightarrow} (\vec{e}e'_1 \cdots e'_{n'}, \vec{\varphi}\varphi'_1 \cdots \varphi'_{n'})$$

and for $1 \leq i' \leq n'$, $(\vec{e}e'_1 \cdots e'_i, \vec{\varphi}\varphi'_1 \cdots \varphi'_{i'}) \in \mathsf{Stage}$

and

(ii) for all $\varphi'_1 \cdots \varphi'_{n'}$ such that for $1 \leq i' \leq n'$, $(\vec{e}e'_1 \cdots e'_{i'}, \vec{\varphi}\varphi'_1 \cdots \varphi'_{i'}) \in \mathsf{Stage}$, there exist $\varphi_1 \cdots \varphi_n$ such that

$$(\vec{e}e_1 \cdots e_n, \vec{\varphi}\varphi_1 \cdots \varphi_n) \underline{\leftrightarrow} (\vec{e}e'_1 \cdots e'_{n'}, \vec{\varphi}\varphi'_1 \cdots \varphi'_{n'})$$

and for $1 \leq i \leq n$, $(\vec{e}e_1 \cdots e_i, \vec{\varphi}\varphi_1 \cdots \varphi_i) \in \mathsf{Stage}$.

Proposition 10. *Given an* E, Φ*-couple* $(\mathsf{Stage}, \vdash)$, $\rightleftharpoons^\circ$ *is an s-congruence* $\subseteq$*-contained in* $\approx_\circ$, *and* $\rightleftharpoons^\circ$ *restricted to* E *is* $\rightleftharpoons_\alpha^+$ *for* α *such that* $\underline{\leftrightarrow}_\alpha = \underline{\leftrightarrow}$.

To see, for instance, that $\rightleftharpoons^\circ$ is an s-congruence, suppose $e_1 \cdots e_n \rightleftharpoons^\circ e'_1 \cdots e'_{n'}$ and $a_1 \cdots a_m \rightleftharpoons^\circ a'_1 \cdots a'_{m'}$, and let us verify the first conjunct (i) for

$$e_1 \cdots e_n a_1 \cdots a_m \;\; \rightleftharpoons^\circ \;\; e'_1 \cdots e'_{n'} a'_1 \cdots a'_{m'} \; .$$

Let $(\vec{e}, \vec{\varphi}) \in \mathsf{Stage}$ and fix $\varphi_1 \cdots \varphi_n, \psi_1 \cdots \psi_m$ such that

$$\text{for } 1 \leq i \leq n, \qquad (\vec{e}e_1 \cdots e_i, \vec{\varphi}\varphi_1 \cdots \varphi_i) \in \mathsf{Stage}$$

and

$$\text{for } 1 \leq j \leq m, \qquad (\vec{e}e_1 \cdots e_n a_1 \cdots a_j, \vec{\varphi}\varphi_1 \cdots \varphi_n \psi_1 \cdots \psi_j) \in \mathsf{Stage} \; .$$

The assumption that $e_1 \cdots e_n \rightleftharpoons^\circ e'_1 \cdots e'_{n'}$ yields $\varphi'_1 \cdots \varphi'_{n'}$ such that

$$(\vec{e}e_1 \cdots e_n, \vec{\varphi}\varphi_1 \cdots \varphi_n) \;\; \underline{\leftrightarrow} \;\; (\vec{e}e'_1 \cdots e'_{n'}, \vec{\varphi}\varphi'_1 \cdots \varphi'_{n'}) \tag{6}$$

and

$$\text{for } 1 \leq i' \leq n', \qquad (\vec{e}e'_1 \cdots e'_{i'}, \vec{\varphi}\varphi'_1 \cdots \varphi'_{i'}) \in \mathsf{Stage} \ .$$

By the definition of $\underline{\leftrightarrow}$, (6) can be extended to

$$\begin{aligned}(\vec{e}e_1 \cdots e_n a_1 \cdots a_m, \vec{\varphi}\varphi_1 \cdots \varphi_n \psi_1 \cdots \psi_m) \ \underline{\leftrightarrow} \\ (\vec{e}e'_1 \cdots e'_{n'} a_1 \cdots a_m, \vec{\varphi}\varphi'_1 \cdots \varphi'_{n'} \chi_1 \cdots \chi_m)\end{aligned} \tag{7}$$

for some $\chi_1 \cdots \chi_m$ such that

$$\text{for } 1 \leq j \leq m, \qquad (\vec{e}e'_1 \cdots e'_{n'} a_1 \cdots a_j, \vec{\varphi}\varphi'_1 \cdots \varphi'_{n'} \chi_1 \cdots \chi_j) \in \mathsf{Stage} \ .$$

The assumption that $a_1 \cdots a_m \rightleftharpoons^{\circ} a'_1 \cdots a'_{m'}$ gives $\psi'_1 \cdots \psi'_{m'}$ such that

$$\text{for } 1 \leq j' \leq m', \qquad (\vec{e}e'_1 \cdots e'_{n'} a'_1 \cdots a'_{j'}, \vec{\varphi}\varphi'_1 \cdots \varphi'_{n'} \psi'_1 \cdots \psi'_{j'}) \in \mathsf{Stage}$$

and

$$\begin{aligned}(\vec{e}e'_1 \cdots e'_{n'} a_1 \cdots a_m, \vec{\varphi}\varphi'_1 \cdots \varphi'_{n'} \chi_1 \cdots \chi_m) \ \underline{\leftrightarrow} \\ (\vec{e}e'_1 \cdots e'_{n'} a'_1 \cdots a'_{m'}, \vec{\varphi}\varphi'_1 \cdots \varphi'_{n'} \psi'_1 \cdots \psi'_{m'})\end{aligned}$$

which by (7) and the transitivity of $\underline{\leftrightarrow}$ gives

$$\begin{aligned}(\vec{e}e_1 \cdots e_n a_1 \cdots a_m, \vec{\varphi}\varphi_1 \cdots \varphi_n \psi_1 \cdots \psi_m) \ \underline{\leftrightarrow} \\ (\vec{e}e'_1 \cdots e'_{n'} a'_1 \cdots a'_{m'}, \vec{\varphi}\varphi'_1 \cdots \varphi'_{n'} \psi'_1 \cdots \psi'_{m'}) \ ,\end{aligned}$$

as required. Note that the step from (6) to (7) becomes problematic if $\underline{\leftrightarrow}$ is replaced by $\underline{\leftrightarrow}_\alpha$, where $\underline{\leftrightarrow}_\alpha$ is different from $\underline{\leftrightarrow}$.[6]

5 Conclusion: further directions

The picture of meaning developed above is based on an analysis of an E, Φ-couple $(\mathsf{Stage}, \vdash)$ in terms largely of relations $[e\rangle \subseteq \mathsf{Stage}^2$, for $e \in E$, and $\vdash$. To appreciate the co-inductive character of this approach, a comparison with the "structural approach to operational semantics" of Plotkin [13] is telling.[7] Transitions there, such as $ap \xrightarrow{a} p$, unwind or decompose a process, contrasting sharply with the transitions $(\vec{e}, \vec{\varphi}) \ [e\rangle \ (\vec{e}e, \vec{\varphi}\varphi)$ above, where labels e are, as it were, external to $(\vec{e}, \vec{\varphi})$. (Turning $ap \xrightarrow{a} p$ around to $p \xleftarrow{a} ap$, we come closer to

[6] The necessity of basing $\rightleftharpoons^{\circ}$ on $\underline{\leftrightarrow}$, rather than on $\underline{\leftrightarrow}_\alpha$ for some α such that $\underline{\leftrightarrow}_{\alpha+1}$ differs from $\underline{\leftrightarrow}_\alpha$, can be seen by constructing stages $(e_{\alpha+\omega}, \varphi_{\alpha+\omega})$ and $(e_{\alpha+\omega+1}, \varphi_{\alpha+\omega+1})$ that provide "copies" of the ordinals $\alpha+\omega$ and $\alpha+\omega+1$ respectively, in the sense of the "ordinal processes" of Klop [10]. The crucial property of the system $\beta \mapsto c(\beta)$ of copies $c(\beta)$ of ordinals β is that

$$c(\beta_1) \ \underline{\leftrightarrow}_{\beta'} \ c(\beta_2) \quad \text{ iff } \quad \beta_1 = \beta_2 < \beta' \text{ or } \min(\beta_1, \beta_2) \geq \beta'$$

for all ordinals β_1, β_2 and β'. To make sure $e_{\alpha+\omega} \ \rightleftharpoons^{+}_{\alpha} \ e_{\alpha+\omega+1}$, ω is added to α to take into account the quantification $(\forall \vec{e'} \in E^*)$ in the definition of $\rightleftharpoons^{+}_{\alpha}$ from $\underline{\leftrightarrow}_\alpha$.

[7] See also Fernando [3], especially §3, for more on the tension between inductive and co-inductive readings of the equation (1) for compositionality.

the transitions $(\vec{e}, \vec{\varphi})\ \langle e]\ (e\vec{e}, \varphi\vec{\varphi})$ mentioned in §3.1.) If the point of Plotkin [13] is to develop (structural) induction on program constructs by delving inside a process, the thrust above is to enlarge a stage, exploring the wider structure of an E, Φ-couple $(\mathsf{Stage}, \vdash)$. In this regard, Proposition 7 above would seem to point to a particular limitation of the transitions $[e\rangle$ (although the proof above of the direction $\Leftarrow$ does depend on a particular choice of $\vdash$).

To go beyond the prefix-closed fragment $\mathsf{Stage}_\circ$ (blocking the direction $\Rightarrow$ of Proposition 7), an anonymous referee has suggested introducing, for every $n \geq 0$, relations $[e_1 \cdots e_n\rangle \subseteq \mathsf{Stage}^2$ labeled by sequences $e_1 \cdots e_n \in E^n$ such that

$$(\vec{e}, \vec{\varphi})\ [e_1 \cdots e_n\rangle\ (\vec{e'}, \vec{\varphi'}) \quad \text{iff} \quad \vec{e'} = \vec{e}e_1 \cdots e_n \text{ and } \vec{\varphi'} = \vec{\varphi}\varphi_1 \cdots \varphi_n \text{ for some } \varphi_1 \cdots \varphi_n$$

for all $(\vec{e}, \vec{\varphi}), (\vec{e'}, \vec{\varphi'}) \in \mathsf{Stage}$, supporting modal operators $\langle e_1 \cdots e_n\rangle$ such that

$$(\vec{e}, \vec{\varphi}) \mid\vdash \langle e_1 \cdots e_n\rangle A \quad \text{iff} \quad \text{for some } \varphi_1 \cdots \varphi_n \in \Phi^n \text{ such that } (\vec{e}e_1 \cdots e_n, \vec{\varphi}\varphi_1 \cdots \varphi_n) \in \mathsf{Stage},\ (\vec{e}e_1 \cdots e_n, \vec{\varphi}\varphi_1 \cdots \varphi_n) \mid\vdash A\ .$$

Assuming Stage is prefix-closed, $\langle e_1 \cdots e_n\rangle A$ is equivalent to $\langle e_1\rangle \cdots \langle e_n\rangle A$, and $[e_1 \cdots e_n\rangle$ is just $[e_1\rangle \circ \cdots \circ [e_n\rangle$, the key behind the equivalence $\rightleftharpoons^\circ$. But the question is whether a relation $[ee'\rangle \neq [e\rangle \circ [e'\rangle$ is properly digested through the recipe in Proposition 8 for cooking $[\![\cdot]\!]_\approx$ up from an s-congruence $\approx$. To analyze $[ee'\rangle \neq [e\rangle \circ [e'\rangle$ alongside $[e\rangle$ and $[e'\rangle$, it may be best to try some other interpretation $[\![\cdot]\!]$ for which

$$[\![b(e_0, b(e, e'))]\!] \ \neq\ [\![b(b(e_0, e), e'))]\!]$$

for every $e_0 \in E$, if not perhaps enrich the set E_b of terms with at least one connective other than b (rather than overloading b with different transitions, to which Proposition 8 may, in principle, apply).[8]

5.1 Non-inertial transitions bounded

Whether or not $[e_1 \cdots e_n\rangle$ can justifiably be called incremental for $n > 1$, it is inertial in that whenever $(\vec{e}, \vec{\varphi})\ [e_1 \cdots e_n\rangle\ (\vec{e'}, \vec{\varphi'})$, not only is $\vec{e'} = \vec{e}e_1 \cdots e_n$, but $\vec{\varphi}$ is a prefix of $\vec{\varphi'}$, unlike the accessibility relations interpreting the primed modal operators $\langle e\rangle'$ in §2.3. With $[e_1 \cdots e_n\rangle$, however, we can approximate these non-inertial relations by defining, for every $n \geq 0$, relations $[e\rangle_n$ that allow backtracking up to n expressions as follows. For $(\vec{e}, \vec{\varphi}) \in E^m \times \Phi^m$ where $m < n$, we backtrack to (ϵ, ϵ):

$$(\vec{e}, \vec{\varphi})\ [e\rangle_n\ (\vec{e'}, \vec{\varphi'}) \quad \text{iff} \quad (\epsilon, \epsilon)\ [\vec{e}e\rangle\ (\vec{e'}, \vec{\varphi'})$$

[8] That said, there is no denying the richness of monoidal processing, as brought out in Visser and Vermeulen [14], compared to which the interpretation in Proposition 8 is incomparably crude.

(recalling that (ϵ, ϵ) is always assumed to belong to Stage). Otherwise, for all $(\vec{e}e_1 \cdots e_n, \vec{\varphi}\varphi_1 \cdots \varphi_n)$ and $(\vec{e'}, \vec{\varphi'}) \in \mathsf{Stage}$, define

$$(\vec{e}e_1 \cdots e_n, \vec{\varphi}\varphi_1 \cdots \varphi_n) \ [e\rangle_n \ (\vec{e'}, \vec{\varphi'})$$

iff

$$(\exists i) \ 1 \leq i \leq n \text{ and } (\vec{e}e_1 \cdots e_{i-1}, \vec{\varphi}\varphi_1 \cdots \varphi_{i-1}) \ [e_i \cdots e_n e\rangle \ (\vec{e'}, \vec{\varphi'}) \ .$$

(Thus, $[e\rangle_0$ is just $[e\rangle$.) The slogan is "revision via re-bracketting," the strategy being to relax the one-expression-at-a-time transition $[e\rangle$ by backtracking to $(\vec{e}e_1 \cdots e_{i-1}, \vec{\varphi}\varphi_1 \cdots \varphi_{i-1})$, and then feeding the larger chunk $e_i \cdots e_n e$ as a single increment to it. Bounding n would then be a way of localizing revision, with modal operators $\langle e\rangle_n$ interpreted by $[e\rangle_n$ approximating $\Diamond\langle e\rangle$.

5.2 Between translators

Instead of insisting that rebracketting $(e_0 e)e'$ to $e_0(ee')$ revises interpretation, we may interpret $e_0 e$ partially so as to allow a further input e' to resolve underspecification in e as well as e_0. In §3.2 above, this led to the idea of inputting the outputs of one translator into another translator, the more general theme being the step up a level of abstraction to constructions on translators. This theme was implicit already in the discussion of $\alpha\dot{\vee}\alpha'$ and $\alpha + \alpha'$, but has otherwise been slighted (as I have concentrated on analyzing b relative to a fixed translator). I close with some words on expanding the modal language $\mathcal{L}_\circ$ to reify the machines α.

Beyond the verbal context represented by a stage $(\vec{e}, \vec{\varphi})$, there are non-verbal contextual elements α that determine an E, Φ-couple $(\mathsf{Stage}_\alpha, \vdash_\alpha)$ for $\mathcal{L}_\circ$. I have referred to α as a machine above, but we may also identify it with some deictic parameter (such as speaker, addressee) that fixes an E, Φ-couple $(\mathsf{Stage}_\alpha, \vdash_\alpha)$ — indeed, even E and Φ. Choosing E and Φ to be large enough so that $E = \Phi$, let us fix some set I of α's (leaving exactly what that is open), decorate $\mathcal{L}_\circ$-stages $(\vec{e}, \vec{\varphi})$ with α's, and collect certain triples $(\vec{e}, \vec{\varphi})_\alpha$ in a set

$$\mathsf{STAGE} \ \subseteq \ \bigcup_{n \geq 0} (E^n \times E^n \times I) \ .$$

It then suffices to assume that for every $\alpha \in I$, there is a relation $\vdash_\alpha \subseteq E^* \times E$ so that for all $e \in E$ and $(\vec{e}, \vec{\varphi})_\alpha \in \mathsf{STAGE}$,

$$\begin{array}{rcl} (\vec{e}, \vec{\varphi})_\alpha \Vdash e & \text{iff} & \vec{\varphi} \vdash_\alpha e \\ (\vec{e}, \vec{\varphi})_\alpha \Vdash \langle e\rangle A & \text{iff} & (\exists\varphi) \ (\vec{e}e, \vec{\varphi}\varphi)_\alpha \in \mathsf{STAGE} \text{ and } (\vec{e}e, \vec{\varphi}\varphi)_\alpha \Vdash A \end{array}$$

plus the usual Boolean clauses for $\wedge$ and $\neg$. We are now in a position to expand $\mathcal{L}_\circ$ with formulas that exploit α. But showing such an expansion is not pointless will have to be done elsewhere.[9]

[9] Some initial steps are recorded in my contribution to the Twelfth Amsterdam Colloquium (December 1999).

Acknowledgments

My thanks to two anonymous referees and Dag Westerståhl for very helpful suggestions and detailed corrections.

References

[1] Johan van Benthem. *Exploring Logical Dynamics.* CSLI, Stanford, 1996.

[2] Kees van Deemter. Towards a logic of ambiguous expressions. In K. van Deemter and S. Peters, editors, *Semantic Ambiguity and Underspecification.* CSLI Lecture Notes Number 55, Stanford, 1996.

[3] Tim Fernando. Ambiguity under changing contexts. *Linguistics and Philosophy*, 20(6), 1997.

[4] Tim Fernando. A modal logic for non-deterministic discourse processing. *Journal of Logic, Language and Information*, 8(4), 1999. **Corrigendum**: the axiom scheme $(\varphi \supset \psi) \equiv (\varphi > \psi)$ in §6 (p.465) should be weakened to $(\varphi > \psi) \supset (\varphi \supset \psi)$.

[5] J. Groenendijk and M. Stokhof. Dynamic predicate logic. *Linguistics and Philosophy*, 14, 1991.

[6] Wilfrid Hodges. Formal features of compositionality. This issue.

[7] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Language and Computation.* Addison-Wesley, 1979.

[8] H. Kamp and U. Reyle. *From Discourse to Logic.* Kluwer Academic Publishers, Dordrecht, 1993.

[9] R. Kempson, W. Meyer-Viol and D. Gabbay. *Dynamic Syntax: The Flow of Language Understanding.* Draft: 5 October 1999. To be published by Blackwell, Oxford.

[10] Jan Willem Klop. Bisimulation semantics. Lectures given at the REX workshop, Noordwijkerhout, May 1988.

[11] M. Marx and Y. Venema. *Multi-Dimensional Modal Logic.* Applied Logic Series Number 4. Kluwer Academic Publishers, 1997.

[12] David Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Proc. 5th GI Conference*, LNCS 104. Springer-Verlag, Berlin, 1981.

[13] Gordon D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.

[14] A. Visser and C.F.M. Vermeulen. Dynamic bracketing and discourse representation. *Notre Dame Journal of Formal Logic*, 37(2), 1996.