

# COUNTING CONSTRUCTIONS AND COERCION: CONTAINER, PORTION AND MEASURE INTERPRETATIONS

PETER R. SUTTON & HANA FILIP  
*Heinrich Heine University*

## ABSTRACT

Counting constructions with mass terms like *two beers* have at least two coerced interpretations depending on context. For example, *two beers* can have either a container (count) interpretation of ‘two glasses filled with beer’ or a portion (count) interpretation of ‘two portions of beer, each (equivalent to) the contents of one glass’. The intriguing puzzle we address, which has escaped attention, is why it is hard to get a measure (mass) interpretation of ‘beer to the amount of two glassfuls’, despite the fact that this and the other two interpretations are available for full pseudo-partitives such as *two glasses of beer*. Our proposal rests on an idea, backed up by co-predication data, that the measure interpretation is derived from the portion interpretation. It follows from this that coerced measure interpretations of counting constructions with mass terms would require coercing an implicit portion concept such as GLASS into a measure interpretation, something which, we argue, cannot easily be done.

## [1] INTRODUCTION

### [1.1] *Interpretations of pseudo-partitives*

It has been observed that pseudo-partitives like *two glasses of beer* are ambiguous between at least two interpretations (i.a. Doetjes 1997; Rothstein 2011): (i) the container interpretation (‘two glasses containing beer’); (ii) the measure interpretation (‘beer to the measure of two glassfuls’). Landman (2016) argues that they are at least four ways ambiguous, with three count interpretations and one mass interpretation. For *two glasses of beer*, the four interpretations with their paraphrases are given in Table 1. In this paper, we will not distinguish between the contents and free portion interpretations, however, because this subtle distinction will not play a role in our analysis. Hence, the focus of this paper will be the semantics of the following three interpretations of the pseudo-partitive construction: CONTAINER, PORTION, and the mass MEASURE interpretation, where ‘portion’ means *free portion or contents*. These three interpretations can be disambiguated in context, as is illustrated in the examples in (1):

Interpretation	Paraphrase	Countability
container	two glasses filled with beer	COUNT
contents	two portions of beer, each the contents of a glass	COUNT
free portion	two one-glassful sized portions of beer	COUNT
measure	beer to the amount of two glassfuls	MASS

TABLE 1: Paraphrases for *two glasses of beer* from Landman (2016)

- (1) a. He turned to reach the two glasses of wine that stood on a bedside table. [BNC]  
 b. i (*sic.*) should set the record straight with Clayart that two glasses of red wine a day have beneficial health results. [UKWaC]  
 c. Two glasses of wine is equal to 3 standard drinks of any alcoholic beverage. [UKWaC]

In (1-a), *two glasses of wine* has a container interpretation, because the verbs *reach* and *stand* lexically select solid objects as denotations of their direct object. In (1-b), *two glasses of red wine* has a portion interpretation, because it is the contents, red wine contained in two glasses, that has the beneficial effect on health. (1-b) also, plausibly, has a measure interpretation of approximately ‘wine to the measure of two glasses-worth’, which has health benefits. In (1-c), *two glasses of wine* has a measure interpretation. The expressed equivalence relation here holds between the alcoholic contents of the alcoholic beverages contained in the respective containers. Moreover, the singular agreement on the verb excludes a container or a portion interpretation of *two glasses of wine*. On count interpretations, *two glasses of wine* is a plural count numerical NP requiring the plural agreement on the verb. The mass interpretation of a measure construction can govern either the singular or plural agreement on the verb (Rothstein 2016, p.16).

#### [1.2] Mass-to-count coercion in numerical NPs

Many mass nouns can be coerced into count interpretations in count syntactic environments in order to resolve a type mismatch and restore compositionality. In English, such contexts include direct modification with numerical expressions in counting constructions (*three wines*) or quantified DPs with distributive quantifiers (*each/every wine*). In order to resolve the type mismatch between a numerical expression and a mass noun, two different shifted count interpretations are readily available, as we see with *two white wines* in the following examples:

- (2) a. John carried two white wines to the table.  
 b. Philippa drank two white wines.

Given that the verb *carried* lexically selects solid objects in the denotation of its direct object argument, in (2-a), the shift of the mass term *white wines* into a count interpretation requires the retrieval of implicit, contextually salient containers from context. In (2-b), *two wines* has a portion reading, because the verb *drink* lexically selects direct objects denoting liquids. Assuming that common containers for wine are glasses, *two wines* might, here, be naturally understood as meaning ‘two portions of wine, each (equivalent to) the contents of a glass’.

In contrast, the measure interpretation of a counting construction consisting of a numerical expression and a mass noun is either hard to get or not available at all. In (3), for example, *left in the bottle* makes a measure interpretation of *two wines* be the most likely interpretation: e.g. ‘wine to the amount of two glassfuls’. Yet this interpretation does not seem to be available for many speakers, and hence the whole sentence is infelicitous.

(3) #There are about two wines left in the bottle.

This novel observation gives us a window on both the semantics of the pseudo-partitive construction, as well as on mechanisms underlying mass-to-count shifts. If a coerced interpretation of *two wines*, for instance, is the same as the interpretation of a full pseudo-partitive like *two glasses of wine*, then assuming that the latter has at least three available interpretations, a container, portion and measure, the same three interpretations should also be available for a coerced interpretation of *two wines*. On this assumption, one would not expect any differences in the availability of the container, portion and measure interpretations for *two wines*, but as we have just seen, this prediction is not borne out, because the measure interpretation is not available in some contexts at all. The main question we address is:

QUESTION: Why should a measure interpretation for numerical NPs like *two wines* be either hard to get or even not available at all?

Our answer is motivated by two independently motivated theses: (i) in pseudo-partitives formed with classifier-like expressions (e.g. *two glasses of wine*), the measure interpretation (‘wine to the measure of two glassfuls’) is derived from the portion interpretation (‘two portions of wine, each the (equivalent to) contents of a glass’); (ii) when mass nouns are directly modified by numerical expressions (as in *two wines*), the measure interpretation is blocked, since it would require coercing an implicit portion concept *GLASS*, something that, we argue, is not possible.

The outline of the paper is as follows. In Section [2], we give a brief overview of two state-of-the-art accounts of the semantics of pseudo-partitives. One, which focuses on their container and measure interpretation, is provided by Rothstein (2011), while the other, proposed by Landman (2016), also concerns the container, contents and measure interpretations. (We subsume the contents and free

portion interpretations under the one portion interpretation.) Our analysis of pseudo-partitives is couched within Type Theory with Records (TTR) (i.a. Cooper 2012) and our previous work on countability and on pseudo-partitives (Sutton & Filip 2017, 2016a; Filip & Sutton 2017) which we introduce in Sections [3.1]-[3.2]. In Section [3.3], we argue for an asymmetry between the measure interpretation, on the one hand, and the container and portion interpretations, on the other hand, based on asymmetries in the acceptability of mixed readings in co-predications. In Sections [3.4]-[3.5], we give a semantic analysis of the pseudo-partitive construction. Finally, in Section [3.6], we show how this analysis allows us to motivate why the coerced measure interpretation of a numerical NP built from a numerical expression and a mass noun, such as *two wines*, is odd or infelicitous.

[2] BACKGROUND: TWO ANALYSES OF PSEUDO-PARTITIVES

[2.1] Rothstein (2011): *The container and measure interpretations of pseudo-partitives* Rothstein (2011)'s account of the CONTAINER interpretation of a pseudo-partitive construction like *three glasses of wine* is based on a function REL that applies to the interpretation of its head *glass* and shifts it to a container classifier. The derivation for *three glasses of wine* is given in (4). In (4), **wine** denotes a kind,  $\cup$ **wine** denotes a predicate,  $*X$  indicates the upward closure of the set  $X$  under mereological sum, and  $\sqcup X$  is the (sum) entity that is the supremum of the set  $X$ .<sup>1</sup>

$$\begin{aligned}
 \llbracket \text{glasses} \rrbracket &= \lambda x. \exists X. X \subseteq * \text{GLASS} : x = \sqcup X \\
 \llbracket \text{glasses of wine} \rrbracket &= (\text{REL}(\llbracket \text{glasses} \rrbracket))(\llbracket \text{wine} \rrbracket) \\
 &= (\lambda z. \lambda x. \exists y. \exists X. X \subseteq * \text{GLASS} : x = \sqcup X \\
 &\quad \wedge \text{CONTAIN}(x, y) \wedge y \in \cup z) (\text{wine}) \\
 &= \lambda x. \exists y. \exists X. X \subseteq * \text{GLASS} : x = \sqcup X \\
 &\quad \wedge \text{CONTAIN}(x, y) \wedge y \in \cup \text{wine} \\
 \llbracket \text{three glasses of wine} \rrbracket &= \lambda x. \exists y. \exists X. X \subseteq * \text{GLASS} : x = \sqcup X \\
 &\quad \wedge \text{CONTAIN}(x, y) \wedge y \in \cup \text{wine} \wedge \text{CARD}(x) = 3
 \end{aligned} \tag{4}$$

The measure interpretation can be understood in terms of a function FUL in (5), which we extrapolate from Rothstein (2011, p. 9). According to Rothstein, it is realised either by the explicit morpheme *-ful* or by its null correlate, yielding the measure interpretation for *three glasses of wine*, as in (6) (Rothstein 2011, p. 9).

$$\llbracket \text{-ful} \rrbracket = \llbracket \emptyset_{ful} \rrbracket = \text{FUL} = \lambda P. \lambda n. \lambda x. \text{MEAS}_{\text{volume}}(x) = \langle P, n \rangle \tag{5}$$

$$\begin{aligned}
 \llbracket \text{three glasses} \rrbracket &= \lambda x. \text{MEAS}_{\text{volume}}(x) = \langle \text{GLASS}, 3 \rangle \\
 \llbracket \text{three glasses of wine} \rrbracket &= \lambda x. x \in \cup \text{wine} \wedge \text{MEAS}_{\text{volume}}(x) = \langle \text{GLASS}, 3 \rangle
 \end{aligned} \tag{6}$$

In summary, we have two functions REL and FUL which apply to the basic interpretation of a receptacle noun, such as *glass*, to derive from it a container classifier and a measure classifier interpretation, respectively.

[1] For an introduction to extensional mereology, see Champollion & Krifka (2016) and references therein.

[2.2] Landman (2016): *The container, contents and measure interpretations of pseudo-partitives*

Landman (2016) is grounded within ‘iceberg semantics’, a theory in which counting does not depend on atomicity. The mass/count distinction applies to lexical nouns, and also to NPs and DPs, which are interpreted as pairs consisting of a body and a base:  $\langle \mathbf{body}, \mathbf{base} \rangle$ . Assuming a complete Boolean algebra  $B$ , the base generates the body under the sum operation:  $\mathbf{body}(X) \subseteq \mathbf{*base}(X)$ . I.e. all elements of the body are sums of elements of the base. For count nouns, the base is the set in terms of which the members of the body are counted. What allows counting is the requirement that the base of their interpretation be (contextually) disjoint. The base is thought of as forming a perspective on the body, which is a subset of an unsorted interpretation domain. The idea is to define the mass/count distinction and the singular/plural distinction in relation to the base, which is taken to allow a simpler and more elegant analysis of mass-count interactions. Under its sortal interpretation, the singular noun *glass* and the plural noun *glasses* are counted in terms of the same disjoint base  $\text{GLASS}_{w,t}$ :

$$\begin{aligned} \llbracket \text{glass} \rrbracket^{w,t} &= \langle \text{GLASS}_{w,t}, \text{GLASS}_{w,t} \rangle \\ \llbracket \text{glasses} \rrbracket^{w,t} &= \langle \mathbf{*GLASS}_{w,t}, \text{GLASS}_{w,t} \rangle \end{aligned} \quad (7)$$

While count nouns are interpreted as i-sets with a disjoint base, mass nouns are i-sets with an overlapping base.

Receptacle nouns can be shifted to CONTAINER and CONTENTS interpretations via the **classifier** function, or to the MEASURE interpretation via the **measure** function (following Khrizman et al. (2015)). The **classifier** function is illustrated below, where  $(X]$  stands for the *part set* of  $X$ :  $\{x \mid x \sqsubset y, y \in X\}$ :

$$\mathbf{classifier}(\llbracket \text{glass} \rrbracket^{w,t}) = \lambda P. \langle \mathbf{body}_P, (\mathbf{body}_P] \cap \mathbf{base}(\llbracket \text{glass} \rrbracket^{w,t}) \rangle \quad (8)$$

if  $P$  is mass or plural.  $\perp$  otherwise.

The container and contents interpretations differ with respect to how  $\mathbf{body}_P$  is interpreted. For the contents interpretation, the interpretation of  $\mathbf{body}_P$  in (8) is given in (9). This enters into the derivation for *glass of wine* in (10): the set of glasses the contents of which are wine.

$$\mathbf{body}_P = \lambda x. \text{GLASS}_{w,t}(x) \wedge \mathbf{body}(P)(\mathbf{contents}_{[\text{GLASS}, P, c], w, t}(x)) \quad (9)$$

$$\begin{aligned} \llbracket \text{glass of wine} \rrbracket^{w,t} &= \langle \mathbf{base}, \mathbf{base} \rangle, \text{ such that:} \\ \mathbf{base} &= \lambda x. \text{GLASS}_{w,t}(x) \wedge \text{WINE}_{w,t}(\mathbf{contents}_{[\text{GLASS}, \text{WINE}, c], w, t}(x)) \end{aligned} \quad (10)$$

The container interpretation amounts to the set of wine portions that are glass

contents and the interpretation of **body**<sub>P</sub> in (8) is as follows:

$$\mathbf{body}_P = \lambda x. \mathbf{body}(P)(x) \wedge \text{GLASS}_{w,t}(\mathbf{contents}_{[\text{GLASS},P,c],w,t}^{-1}(x)) \quad (11)$$

$$\begin{aligned} \llbracket \text{glass of wine} \rrbracket^{w,t} &= \langle \mathbf{base}, \mathbf{base} \rangle \text{ such that:} \\ \mathbf{base} &= \lambda x. \text{WINE}_{w,t}(x) \wedge \text{GLASS}_{w,t}(\mathbf{contents}_{[\text{GLASS},\text{WINE},c],w,t}^{-1}(x)) \end{aligned} \quad (12)$$

Turning to the measure interpretation, the function **measure** applies to the sortal receptacle concept  $\llbracket \text{glass} \rrbracket^{w,t}$ . In (13),  $\mathbf{glass}_{w,t}$  is a measure function for glassfuls of stuff of type  $\langle n, et \rangle$ .  $\uparrow$  is a function from sets of objects to sets of object-measure value pairs.  $\downarrow$  is the inverse of  $\uparrow$ .  $\mathbf{m}$  is a contextually given measure value. For *glass of wine*, on the measure interpretation, this yields (14).

$$\begin{aligned} \mathbf{measure}(\llbracket \text{glass} \rrbracket^{w,t}) &= \lambda N. \lambda P. \langle \mathbf{body}_{P,N}, \mathbf{base}_{P,N} \rangle \\ &\quad \text{if } P \text{ is mass or plural. } \perp \text{ otherwise, such that:} \end{aligned} \quad (13)$$

$$\begin{aligned} \mathbf{body}_{P,N} &= (\mathbf{body}(\llbracket \text{glass} \rrbracket^{w,t}) \circ N) \cap \mathbf{body}(P) \\ \mathbf{base}_{P,N} &= \uparrow(\mathbf{body}_P) \cap \mathbf{base}(\llbracket \text{glass} \rrbracket^{w,t}) \\ &= \{ \langle y, \mathbf{glass}_{w,t}(y) \rangle \mid \mathbf{glass}_{w,t}(y) \leq \mathbf{m} \wedge \exists x [\mathbf{base}(P)(x) \wedge y \sqsubseteq x] \} \end{aligned}$$

$$\begin{aligned} \llbracket \text{glass of wine} \rrbracket^{w,t} &= \langle \mathbf{body}, \mathbf{base} \rangle, \text{ such that:} \\ \mathbf{body} &= \lambda n. \lambda x. \mathbf{glass}_{w,t}(x) = n \wedge \text{WINE}_{w,t}(x) \\ \mathbf{base} &= \{ \langle y, \mathbf{glass}_{w,t}(y) \rangle \mid \mathbf{glass}_{w,t}(y) \leq \mathbf{m} \wedge \exists x [\text{WINE}_{w,t}(x) \wedge y \sqsubseteq x] \} \end{aligned} \quad (14)$$

The **body** is a function from numbers  $n$  to a set of amounts of wine that measure  $n$  with respect to the measure  $\mathbf{glass}_{w,t}$ . The **base** is a set of pairs of amounts of stuff which are parts of something that is wine and numbers on a measure scale  $n$  such that  $n$  is less than a contextually specified amount  $\mathbf{m}$ . For Landman, it is vital that the set of  $y$  in the base ( $\downarrow \mathbf{base}$ ) is an overlapping set in order to derive that the measure interpretation is mass.

For our analysis, what matters the most is the following feature of Landman's and Rothstein's accounts: both rely on specific functions to derive the measure interpretation, as opposed to the contents and container interpretations, all of which apply directly to the interpretation of receptacle nouns, such as *glass*.

### [3] ANALYSIS

Our analysis is guided by the following two hypotheses concerning the interpretation of nouns like *glass* that form a measure phrase in pseudo-partitives:

- (H1) The container and portion interpretations are default interpretations, captured by the dot type **container • portion**.
- (H2) The measure interpretation is derived from the portion interpretation: **measure = g(portion)**.

(H1) and (H2) allow us to motivate the following interpretation patterns:

- (i) The relative ease with which full pseudo-partitives (e.g. *three glasses*

*of wine*) participate in co-predication over their container and portion interpretations.

- (ii) The general difficulty of assigning the measure interpretation to expressions like *three wines*.

In what follows, we will first introduce Type Theory with Records (TTR) (i.a. Cooper 2012) and our mereological enrichments of the theory. We then lay out our theory of the mass/count distinction and of pseudo-partitives within mereological TTR, based on Sutton & Filip (2016a, 2017) and Filip & Sutton (2017)

### [3.1] Type Theory with Records and mereological Type Theory with Records

Type Theory with Records (TTR) (Cooper 2005, 2012; Cooper & Ginzburg 2015) is a theory of types which integrates lexical semantic frame-based representations in the sense of Fillmore (1975, 1976) with compositional semantics in the Montague tradition. Types are ‘rich’ in the sense that there are many basic types instead of only *e*, *t* (and sometimes *s*) in Montague-style semantic theories. For example, there can be types for individuals, locations, times, and situations.

In TTR, one can form arbitrarily complex sets of labelled types called *Record Types* (labels approximate discourse referents in DRT). Record types, displayed in tabular format (15), are sets of fields whose first member is a *label* and whose second member is a *type*:

$$\left[ \begin{array}{l} l_1 : T_1 \\ \dots \\ l_n : T_n \end{array} \right] \quad (15)$$

An example of a record type is given in (16), which represents the type of situation in which a glass contains water.

$$\left[ \begin{array}{l} x : Ind \\ y : Subst \\ s_g : \langle \lambda v. glass(v), \langle x \rangle \rangle \\ s_w : \langle \lambda v. water(v), \langle y \rangle \rangle \\ s_{cgw} : \langle \lambda v_1, \lambda v_2. contain(v_1, v_2), \langle x, y \rangle \rangle \end{array} \right] \quad (16)$$

We assume basic types *Ind* and *Subst* for individuals and substances, respectively. Entities of type *Ind* are things like a woman, a cat, an item of furniture, a grain of rice. Entities of type *Subst* are things like mud, water, and air. In other words, *Ind* and *Subst* track the prelinguistic distinction between substances and objects in the sense of Soja et al. (1991). The record type in (16) also contains predicate types (pTypes). They are built from predicates, which are a kind of type constructor. For example,  $\langle \lambda v_1, \lambda v_2. contain(v_1, v_2), \langle x, y \rangle \rangle$  is a function which takes the values for labels *x* and *y* and returns the type of situation in which the value for *x* contains the value for *y*. Such types are dependent types in the sense that the final type

depends on how values are assigned to labels in other fields. For convenience, we will use the abbreviated conventional notation for pTypes as in (17).

$$\left[ \begin{array}{l} x \quad : \quad Ind \\ y \quad : \quad Subst \\ s_g \quad : \quad glass(x) \\ s_w \quad : \quad water(y) \\ s_{cgw} \quad : \quad contain(x, y) \end{array} \right] \quad (17)$$

Assignments of values to labels, in TTR, are represented as *records*: sets of fields, whose first member is a LABEL (to the left of the '=' in (18)) and whose second member is a VALUE for this label (to the right of the '=' in (18)):

$$\left[ \begin{array}{l} x \quad = \quad v_1 \\ \dots \\ y \quad = \quad v_n \end{array} \right] \quad (18)$$

Records are the entities of which record types are true or false ('proofs' of propositions in type-theoretic terminology, 'witnesses' in a natural language setting). For example, (19) specifies a situation in which there is an individual, *a*, some stuff, *b*, and three witnesses/proofs  $p_1$ ,  $p_2$  and  $p_3$ . Witnesses can be thought of as situations or parts of the world that make type judgements true or false.

$$\left[ \begin{array}{l} x \quad = \quad a \\ y \quad = \quad b \\ s_g \quad = \quad p_1 \\ s_b \quad = \quad p_2 \\ s_{gbw} \quad = \quad p_3 \end{array} \right] \quad (19)$$

Individuals in the domain are typed. The record in (19) is a witness for the record type in (16)/(17) iff:

$$\begin{array}{ll} a \quad \text{is of type} \quad Ind, & p_1 \quad \text{is a witness of} \quad glass(a), \\ b \quad \text{is of type} \quad Subst, & p_2 \quad \text{is a witness of} \quad water(b), \\ & p_3 \quad \text{is a witness of} \quad contain(a, b) \end{array}$$

In summary, we have record types which specify types of situations, and records which provide parts of the world which record types are true of. As such, TTR, as so far stated, is compatible with a standard formal semantics based on situations or partial worlds. However, on a more philosophical level, TTR is argued to be a more cognitively oriented theory which represents the conceptual schemes of agents and the judgements *s/he* makes. A judgement of the form  $r :_A T$  (*A* judges that a record, *r* is of type *T*) may reflect the types that *A* has, however, we might also take records to be reflections of *A*'s conceptual scheme. In this paper,



we adopt this more cognitively geared approach. In particular, we will make use of *individuation schemas* which represent perspectives on entities that are relevant to individuation and enumeration. We give full details shortly, but in brief, the same entities may be subject to a variety of individuation perspectives. For example, for entities  $a, b$  (e.g. a set of two nesting tables), an agent may perceive them as two disjoint entities  $a$  and  $b$ , as a mereological sum  $a \sqcup b$ , or as overlapping and so not individuated (as  $a, b$ , and  $a \sqcup b$  simultaneously).

Natural language predicates denote properties of type  $[x : Ind] \rightarrow RecType$  (which we abbreviate as *Ppty*), a function from records containing individuals, to a record type (where  $T \rightarrow T'$  is a functional type such that  $f : T \rightarrow T'$  iff  $f$  is a function with a domain of entities of type  $T$  and a range of entities of type  $T'$ ). For example, a simplified representation of *glass* would be as in (20) (a full-fledged semantic representation of lexical common nouns is given further below):

$$\lambda r : [x : Ind]. ([ s_{glass} : glass(r.x) ]) \quad (20)$$

In (21),  $r.x$  means that the value to be supplemented is the value of  $x$  in  $r$ . Hence, if provided with a record  $[x = a]$ , (21) would yield the type of situation in which  $a$  is a *glass*:

$$[ s_{glass} : glass(a) ] \quad (21)$$

Finally, we will make use of *singleton types* and *manifest fields* as defined for TTR in Cooper (2012, p. 297). For any type  $T$ , and entity  $a : T$ ,  $T_a$  is a singleton type such that for any  $b : T$ ,  $b$  is a witness to the type  $T_a$  iff  $a = b$ . Singleton types are represented in Record Types via manifest fields where the Record Types in (22) and (23) are examples of notational variants (of which we will use the latter).

$$[ x : Ind_a ] \quad (22)$$

$$[ x = a : Ind ] \quad (23)$$

### [3.2] A semantics for common nouns and the mass/count distinction

Context-sensitivity has become a key factor in the grounding of the mass/count distinction in recent theories (Rothstein 2010; Chierchia 2010; Landman 2011; Sutton & Filip 2016b, 2017). For example, count concepts specify either contextually grounded ‘semantic atoms’ (Rothstein 2010), disjoint sets for counting relative to context (Landman 2011; Sutton & Filip 2016b, 2017), or are quantized at each context (Filip & Sutton 2017), while mass concepts do not. Such context sensitivity was first addressed in connection with a sizeable subclass of count nouns like *fence*, *bouquet*, *wall* for which what counts as ‘one’ entity for counting varies with context. So fencing around a square field can count as one fence in some contexts and as four fences in others (Rothstein 2010). In contrast, the lexical entry of a mass noun like *mud* does not specify a unique partition of individuals in any

context (unless the relevant concept is coerced into a portion interpretation, for instance).

Here we build upon [Sutton & Filip \(2017\)](#) and provide a semantics for common nouns that captures such context sensitivity by means of individuation schemas for counting which may vary with the context. Following [Sutton & Filip \(2017\)](#), we also enrich the domain, standardly assumed in TTR, to include mereological sums, such that for any two entities in the domain  $a, b$ , their sum  $a \sqcup b$  is also in the domain. Following [Krifka \(1989\)](#) (along with e.g. [Sutton & Filip \(2016b\)](#) and [Landman \(2016\)](#)), we presuppose a non-atomic domain (one which is not determinately atomic or atomless).

As in standard formal semantics, and its TTR variant, common nouns like *glass* would specify a predicate such as  $\lambda v.\text{glass}(v)$  which determines the application conditions for the noun such that  $v$  is restricted to be of type *Ind*. The plural *glasses* would require any single entity of type *Ind*, or a sum thereof (given as  $*\text{Ind}$ ), and for the predicate to be applicable to any single glass or sum thereof ( $\lambda v.*\text{glass}(v)$ ):

$$\llbracket \text{glass} \rrbracket = \lambda r : [ x : \text{Ind} ] . [ s : \text{glass}(r.x) ] \quad (24)$$

$$\llbracket \text{glasses} \rrbracket = \lambda r : [ x : *\text{Ind} ] . [ s : *\text{glass}(r.x) ] \quad (25)$$

However, as argued by [Landman \(2016\)](#); [Sutton & Filip \(2017\)](#); [Filip & Sutton \(2017\)](#), we actually need two fields for any common noun. Roughly, one which specifies the counting base, i.e. the entities that count as ‘one’, and the other which amounts to the standard conditions for application, that is, the extension. One reason for keeping the counting base and the application conditions separate is that plurals have the same counting base as their singular counterparts, but an extension which includes both singularities and sums.

The lexical entry for the singular common noun *glass* is given in (26), below, in which *Ppty* abbreviates the type  $[x : \text{Ind}] \rightarrow \text{RecType}$ , functions from records to record types. Records as arguments to this function contain not only witnesses for individuals, but also a schema of individuation  $i : \text{Sch}$ , such that *Sch* abbreviates  $(x : \text{Ind} \rightarrow \text{PType}) \rightarrow (x : \text{Ind} \rightarrow \text{PType})$ . It is analysed as a modifier of a predicate such that what a predicate applies to may vary depending on the schema being applied. For instance, a predicate that inherently denotes an overlapping set of entities, may, in relation to an individuation schema, denote some partition of this set that contains only disjoint entities. To take the example of *fence* once more, whereas  $a, b, c$ , and  $d$  may all fall under the predicate *fence* at individuation schema  $i_1$ , their sum,  $a \sqcup b \sqcup c \sqcup d$  may fall under the predicate *fence* at individuation schema  $i_2$ . The field labelled  $p_{\text{base}}$  (the counting base field) is a manifest field which makes use of singleton types (recall that, if  $l = a : T$ , then if  $b : T_a$ ,  $b = a$ ). This field specifies that the witness/proof for  $p_{\text{base}}$  must be of the type *Ppty* such that it is restricted to the property  $\lambda r' : [y : \text{Ind}]. [s_1 : r.i(\text{glass})(r.y)]$ , i.e. a type,

the only witness for which is the property of being a single glass. Finally, the field labelled  $s_{app}$  specifies the application conditions of *glass*. This is a function from a record that is a witness to an individual and an individuation schema, to a type of situation in which the individual is a glass under that schema.

$$\begin{aligned} \llbracket \text{glass} \rrbracket &= \lambda r : \left[ \begin{array}{l} x: \text{Ind} \\ i: \text{Sch} \end{array} \right] \cdot \left[ \begin{array}{l} p_{cbase} = \lambda r' : [y: \text{Ind}]. [s: r.i(\text{glass})(r.y)] : \text{Ppty} \\ s_{app} : p_{cbase}(r) \end{array} \right] \\ &= \lambda r : \left[ \begin{array}{l} x: \text{Ind} \\ i: \text{Sch} \end{array} \right] \cdot \left[ \begin{array}{l} p_{cbase} = \lambda r' : [y: \text{Ind}]. [s: r.i(\text{glass})(r.y)] : \text{Ppty} \\ s_{app} : [s: r.i(\text{glass})(r.x)] \end{array} \right] \end{aligned} \quad (26)$$

The lexical entry for the plural common noun *glasses* is given in (27) which differs from (26) only in so far as the predicate in the ptype in the field labelled  $s_{app}$  applies to singularities and sums represented with the \*operator.

$$\llbracket \text{glasses} \rrbracket = \lambda r : \left[ \begin{array}{l} x: * \text{Ind} \\ i: \text{Sch} \end{array} \right] \cdot \left[ \begin{array}{l} p_{cbase} = \lambda r' : [y: \text{Ind}] [s_1: r.i(\text{glass})(r.y)] : \text{Ppty} \\ s_{app} : [s_2: *r.i(\text{glass})(r.x)] \end{array} \right] \quad (27)$$

(27) is a function from a record that is a witness to an individual or a sum individual and an individuation schema, to a type of situation in which each individual in the sum is a glass under that schema.

Numerical expressions are interpreted as record types, i.e. the type of witnesses for the singleton type of real numbers equal to  $n$ , e.g. 3, below.

$$\llbracket \text{three} \rrbracket = [n=3 : \mathbb{R}] \quad (28)$$

The function *NMOD* shifts such types to a Ppty modifier. The effect of which is to add a condition that the argument individual has some cardinality relative to the counting base property.  $P(r).p_{cbase}$  is the path labelled  $p_{cbase}$  in  $P$  after  $P$  is applied to  $r$ .

$$\begin{aligned} \text{NMOD} &= \lambda R: [n: \mathbb{R}] \lambda P: \text{Ppty} \lambda r : \left[ \begin{array}{l} x: * \text{Ind} \\ i: \text{Sch} \end{array} \right] \cdot \\ &\quad \left[ \begin{array}{l} N\_type = P(r): \text{RecType} \\ s : \text{Card}(r.x, P(r).p_{cbase}, R.n) \end{array} \right] \end{aligned} \quad (29)$$

$$\begin{aligned} \text{NMOD}(\llbracket \text{three} \rrbracket) &= \lambda P: \text{Ppty} \lambda r : \left[ \begin{array}{l} x: * \text{Ind} \\ i: \text{Sch} \end{array} \right] \cdot \\ &\quad \left[ \begin{array}{l} N\_type = P(r): \text{RecType} \\ s : \text{Card}(r.x, P(r).p_{cbase}, 3) \end{array} \right] \end{aligned} \quad (30)$$

$$\begin{aligned} \text{NMOD}(\llbracket \text{three} \rrbracket)(\llbracket \text{glasses} \rrbracket) &= \lambda r : \left[ \begin{array}{l} x: * \text{Ind} \\ i: \text{Sch} \end{array} \right] \cdot \\ &\quad \left[ \begin{array}{l} N\_type = \left[ \begin{array}{l} p_{cbase} = \lambda r' : [y: \text{Ind}] [s_1: r.i(\text{glass})(r'.y)] : \text{Ppty} \\ s_{app} : [s_2: *r.i(\text{glass})(r.x)] \end{array} \right] : \text{RecType} \\ s_{card} : \text{Card}(r.x, N\_type.p_{cbase}, 3) \end{array} \right] \end{aligned} \quad (31)$$

The counting base property is needed since cardinality is relative to some individuating or counting property. To take an example from [Link \(1983\)](#), if  $x$  counts as 52 with respect to *playing card*, it can count as 1 with respect to *deck of cards*.

After flattening and relabelling, this gives us the function in (32) which is the same as the entry for *glasses*, but with the added condition that the plural individual witnessed in the record counts as 3 with respect to the counting base.

$$= \lambda r : \left[ \begin{array}{l} x : *Ind \\ i : Sch \end{array} \right] \cdot \left[ \begin{array}{l} p_{cbase} = \lambda r' : [y : Ind][s_1 : r.i(\text{glass})(r'.y)] : Ppty \\ s_{app} : [s_2 : *r.i(\text{glass})(r.x)] \\ s_{card} : Card(r.x, p_{cbase}, 3) \end{array} \right] \quad (32)$$

Following [Landman \(2011, 2016\)](#); [Sutton & Filip \(2016b, 2017\)](#), mass nouns are distinguished from count nouns in terms of whether their counting base is disjoint. If counting base predicates specify a disjoint set of entities, the relevant noun is count, if it specifies a non-disjoint set of entities, the relevant noun is mass. Overlap in counting bases makes ‘counting go wrong’ ([Landman 2011](#)). [Sutton & Filip \(2017\)](#) represent this in terms of a difference in types of individuation schemas. Mass noun lexical entries specify a **join individuation schema**,  $i_{\vee}$  which is formed as a join type of all specific individuation schemas. This means that if there are different ways of individuating some stuff, such as furniture, into different items, or water into different portions, at the join individuation schema, all of the different partitions are simultaneously available at a given context. This results in overlaps and a predicate that is undefined for cardinality.

For a mass noun like *wine* we get the entry in (33). The argument record requires a Spelke substance, rather than an individual and the resulting record type specifies that the join individuation schema  $i_{\vee}$  is to be used. (We use ‘Spelke substance’ to refer to substances in the sense of [Soja et al. \(1991\)](#).)

$$[[\text{wine}]] = \lambda r : [x : Subst] \cdot \left[ \begin{array}{l} i = i_{\vee} : Sch \\ p_{cbase} = \lambda r' : [y : Subst] [s : i(\text{wine})(r'.y)] : Ppty \\ s_{app} = [s : i(\text{wine})(r.x)] \end{array} \right] \quad (33)$$

Since there are many ways to partition wine into portions and since these overlap with one another, the resulting concept is mass. For treatments of object mass nouns (e.g. *furniture*) and granular mass nouns (e.g. *rice*), see [Sutton & Filip \(2017\)](#).

### [3.3] *Measure interpretations are derived from portion interpretations: The data from co-predication*

In standard dictionaries, receptacle nouns like *glass* or *jar* are commonly treated as polysemous between the container and the portion interpretations, while the measure interpretation is not a part of their inherent lexical entries. This would seem to be validated by data from co-predication. First, let us consider co-predications over the container (C) and portion (P) interpretations:

- (34) a. The two glasses of wine with tall, thin stems are being drunk (C-P)  
by Rachel and Matt.
- b. Loretta and Fiona are drinking the two glasses of wine with (P-C)  
tall, thin stems.

In (34-a), the subject NP contains *glasses* and its PP modifier *with tall, thin stems* makes salient its container (C) interpretation, and the verb *drink*, which follows, selects its portion (P) interpretation. In (34-b), we have the reverse (P-C) order. This shows that nouns, such as *glass*, *bottle*, *pot*, have simultaneously accessible container (C) and portion (P) interpretations, which can be easily accessed, modulo selectional restrictions of their co-constituents.

In (35-a) and (35-b) below, we have co-predication over the portion (P) and measure (M) interpretations of pseudo-partitives, which some people accept, while others find them less than fully felicitous, but in any case, they are not as straightforwardly acceptable as the co-predictions over the portion (P) and container (C) interpretations.

- (35) a. (#) The two glasses of wine with a sour flavour were the last (P-M)  
two in the bottle from two days ago.
- b. (#) The last two glasses of wine in the bottle were drunk by (M-P)  
Carl at lunch and Harry at dinner.

The ease with which the container (C) and portion (P) interpretations of nouns like *glass*, *bottle*, *pot* are accessible in co-predications (but not the measure (M) interpretation) suggests that these are in a sense their ‘default’ interpretations. We can motivate this observation hypothesising that such nouns are interpreted at a dot type, in the sense of (Pustejovsky 1995)<sup>2</sup>: **container • portion**, where the container (C) and portion (P) interpretations are treated as two distinct and equal aspects of their lexical meaning. Pustejovsky’s paradigm example is *book*, as in *Amy picked up and read a book*, where *pick up* selects a physical object (**phys**), while *read* ‘informational print matter’ aspect of the meaning of *book*, which is taken to be of the dot type **phys • info** (Pustejovsky 2011, pp. 1410-1411).

The relative difficulty of accessing the measure (M) interpretation of nouns like *glass*, *bottle*, *pot* can be motivated, if we assume that the measure (M) interpretation is derived from the portion (P) interpretation, as we hypothesise in (H2): there is a function *g* such that **measure** = *g*(**portion**). Intuitively, *two glasses of wine* when understood in the mass sense of ‘wine to the amount of two glassfuls’ cannot be analysed in terms of a container (C) classifier interpretation

[2] The suggestion that the semantics of receptacle nouns such as *glass* requires something akin to dot types was originally made in Partee & Borschev (2012). Partee & Borschev (2012) do not provide such an analysis, however.

(which is count), and neither in terms of a portion (P) classifier interpretation (which is also count). This is because the measure (M) interpretation, which is mass, is not tied to glasses, qua physical containers or their contents/portions of stuff they contain, but rather to *GLASS* in its abstract non-standard measurement function. This also motivates why on our account the container (C) and portion (P) interpretation are unified under the dot type and treated as part of the lexical meaning of nouns like *glass*, *bottle*, *pot*, and the measure (M) interpretation requires a lexical shift. In support of this lexical derivational move, we propose the explanation that co-predications with the measure (M) and the container (C) interpretation are odd, because the measure (M) interpretation blocks access to the container (C) interpretation. This, we suggest, is because the measure (M) interpretation is a result of the shift from the **portion** constituent of the dot type **container • portion**, and the type for the **container** is ‘lost’ as a result of shifting from the portion to the measure interpretation. This is what we see in sentences such as (36-a) and (36-b).

- (36) a. #The two glasses of wine with tall, thin stems were the last two left in the bottle. (C-M)  
 b. #The last two glasses of wine in the bottle have thin stems. (M-C)

In sum, the above data and observations confirm our guiding hypotheses in (H1) and (H2) regarding the lexical semantic properties of nouns like *glass*, *bottle*, *cup*, *pot*. Hypothesising that they are of the dot type **container • portion** (H1) correctly predicts that their container (C) and portion (P) interpretations are simultaneously available and can be easily accessed in co-predications. Hypothesising that the measure (M) interpretation is a result of a shift from the portion constituent of the dot type (H2) motivates the observation that the measure (M) and portion (P) interpretations cannot be easily simultaneously accessed in co-predications, or are impossible to access simultaneously for some speakers at least. In Section [3.6] we will show how (H1) and (H2) jointly predict why it should be the case that the measure (M) interpretation of counting constructions like *two white wines*, as in (3), is hard to get, or even impossible, when the container (C) and portion (P) interpretations, as in (2-a) and (2-b), respectively, are straightforwardly available. This is the main puzzle of this paper. However, first we will turn to our formal analysis of the container (C), portion (P) and measure (M) interpretations for pseudo-partitives.

### [3.4] *Container and portion interpretations for pseudo-partitives*

We assume that container and portion interpretations of pseudo-partitives are derived via a function *CL* (standing for ‘classifier’) applied to the interpretation of nouns like *bottles*, *cups*, *glass* which have an inherently non-classifier denotation

involving physical objects with a certain receptacle shape.  $CL$  is a function which turns a property like  $\llbracket \text{glass} \rrbracket$ , which is not inherently relational, to a relational classifier concept (that requires a property as an argument, i.e. the contents of the receptacle). Thus  $CL$  is a function from two properties (e.g.  $\llbracket \text{glass} \rrbracket$  and  $\llbracket \text{wine} \rrbracket$ ) to a function from a record to the TTR equivalent of a dot type (Cooper 2011), that is, where there are two fields, one for each of the types: container and portion (labelled  $s_{ctr}$  and  $s_{por}$ ) in (34). The  $P$  and  $Q$  variables stand for properties such as those expressed by  $\llbracket \text{glass} \rrbracket$  and  $\llbracket \text{wine} \rrbracket$ .

The resulting type is given in (38). The type labelled  $s_{ctr}$  in (38) is a record type which retains the content of  $\llbracket \text{glass} \rrbracket$ .  $par$  is a label for a record of the type to the right of the colon, which introduces the existential requirement that a situation witnesses individuals or substances and an individuation schema. The conditions labelled  $s_{in}$  thereby state that, for each glass, there is some stuff to which *wine* can be applied in that glass. In other words, a property for glasses that each contain wine. The type labelled  $s_{por}$  in (38) is a record type which retains the content of  $\llbracket \text{wine} \rrbracket$ , but crucially, portions it in some way provided by the the schema labelled *r.i.* That is, we get some disjoint partition of wine. The type labelled  $par$  existentially binds two conditions (labelled  $s_{in}$ ), one of which states that, for each disjoint partition of wine, there is some glass it is from. In other words, a property for portions of wine, each the contents of a glass. The relation **in.each**( $x, z$ ) means *in each  $x$ , there is  $z$*  (e.g. in each glass, there is wine). The relation **each.in**( $x, z$ ) means *in each  $z$ , there is  $x$* . For example, each (portion of) wine, is in a glass. As such these relations suppress some complexity relating to quantifier scope.<sup>3</sup>

$$CL = \lambda P:Ppty \lambda Q:Ppty \lambda r : \left[ \begin{array}{l} x: *Ind \vee Subst \\ i: Sch \end{array} \right].$$

$$\left[ \begin{array}{l} s_{ctr}: \left[ \begin{array}{l} N\_type = P(r) : RecType \\ par: \left[ \begin{array}{l} z: *Ind \vee Subst \\ i = i_{\vee} : Sch \end{array} \right] \\ r.x : *Ind \\ s_{in} : Q(par).s_{app} \wedge [s_{contain} : \mathbf{in.each}(r.x, par.z)] \end{array} \right] \\ s_{por}: \left[ \begin{array}{l} N\_type = Q(r) : RecType \\ par: \left[ \begin{array}{l} z: *Ind \\ i = i_{\vee} : Sch \end{array} \right] \\ s_{in} : P(par).s_{app} \wedge [s_{within} : \mathbf{each.in}(r.x, par.z)] \end{array} \right] \end{array} \right] \quad (37)$$

[3] Using predicate logic, the following approximations hold (such that  $contain(x,z)$  means that  $x$  contains  $z$ ): **in.each**( $x, z$ )  $\leftrightarrow \forall x \exists z contain(x, z)$ ; **each.in**( $x, z$ )  $\leftrightarrow \forall z \exists x contain(x, z)$ . However, we do not wish to be committed to there being e.g. an actual glass involved for every glass-portion reading, hence, we suspect that the relation **each.in**( $x, z$ ) needs to be intensionalised (such that, e.g. each portion of wine could be contained in a glass whether or not an actual glass is present in the relevant situation). We leave a TTR analysis of these relations for further research.



$$\llbracket \text{glasses of wine} \rrbracket = CL(\llbracket \text{glasses} \rrbracket)(\llbracket \text{wine} \rrbracket) = \lambda r : \left[ \begin{array}{l} x: *Ind \vee Subst \\ i: Sch \end{array} \right].$$

$$\left[ \begin{array}{l} s_{ctr} : \left[ \begin{array}{l} p_{cbase} = \lambda r' : [y : Ind] [s_1 : r.i(\text{glass})(r'.y)] : Ppty \\ s_{app} : *r.i(\text{glass})(r.x) \\ par : \left[ \begin{array}{l} z : *Ind \vee Subst \\ i = i_{\vee} : Sch \end{array} \right] \\ r.x : *Ind \\ s_{in} : \left[ \begin{array}{ll} s_{app} & : par.i(\text{wine})(par.z) \\ s_{contain} & : \mathbf{in. each}(r.x, par.z) \end{array} \right] \end{array} \right] \\ s_{por} : \left[ \begin{array}{l} p_{cbase} = \lambda r' : [y : Ind] [s_1 : r.i(\text{wine})(r'.y)] : Ppty \\ s_{app} : r.i(\text{wine})(r.x) \\ par : \left[ \begin{array}{l} z : *Ind \\ i = i_{\vee} : Sch \end{array} \right] \\ s_{in} : \left[ \begin{array}{ll} s_{app} & : *par.i(\text{glass})(par.z) \\ s_{within} & : \mathbf{each.in}(r.x, par.z) \end{array} \right] \end{array} \right] \end{array} \right] \quad (38)$$

One of our chief improvements over previous analyses is that the container (C) and portion (P) interpretations are both available simultaneously, somewhat along the lines akin to those captured by a dot type. This allows us to motivate why *glass(es)* can be straightforwardly interpreted as either container or portion within the same sentence, such as in (34-a) and (34-b).

### [3.5] Measure interpretations of pseudo-partitives

Measure interpretations of full pseudo-partitives, we argue, are derived from the portion interpretation of inherent receptacle nouns like *glass*. Take, for instance, the measure interpretation of *glasses* in *two glasses of wine*, which is a property of wine that measures 2 with respect to a contextually provided dimension (most commonly volume) and a quantity (such as *glass-sized portion*) that determines the value on the scale for that dimension. So a type like **Measure**(*x*, **volume**, **litre**, 2) would be a type in which *x* measures 2 in terms of volume in litres. Similarly, **Measure**(*x*, **volume**, **portion\_glass**, 2) would be a type in which *x* measures 2 in terms of volume based upon glass-sized portions of wine.

This idea is formally represented via a function *MSR* (standing for ‘measure’):

$$MSR = \lambda \mathcal{P} : Ppty(Ppty) \lambda Q : Ppty \lambda R : [n : \mathbb{R}] \lambda r : \left[ \begin{array}{l} y: *Ind \vee Subst \\ d: Dimension \end{array} \right].$$

$$\left[ \begin{array}{l} par_2 : \left[ \begin{array}{l} x: *Ind \vee Subst \\ i: Sch \end{array} \right] \\ s_{msr} : \mathbf{Measure}(r.y, r.d, (\mathcal{P}(Q)(par_2).s_{por}), R.n) \end{array} \right] \quad (39)$$

*MSR* maps properties, such as  $CL(\llbracket \text{glass} \rrbracket)$ , to a function from real numbers  $n$  to a function from a property to be measured (e.g.  $\llbracket \text{wine} \rrbracket$ ) to a property of substances or pluralities of individuals such that they are the type of measuring  $n$  with respect to a volume scale marked in terms of contents of glasses of, e.g. wine.



The quantity for the **Measure** relation is provided by applying the witness of  $par_2$  to the interpretation of e.g.  $CL[[\text{glasses}]]$  and selecting the record type for the portion labelled  $s_{por}$  (which is what is meant by  $(P(par_2).s_{por})$  in (39)). This means that, although *MSR* applies to  $[[\text{glasses}]]$ , it operates only on the portion concept (labelled  $s_{por}$ ) and the container concept drops out of the resulting record type. The result of applying *MSR* to  $CL([[glasses]])$  with the further argument  $[[\text{wine}]]$  is given in (40).

$$MSR(CL([[glasses]]))([[wine]]) = \lambda R: [n : \mathbb{R}] \lambda r : \left[ \begin{array}{l} y: *Ind \vee Subst \\ d: Dimension \end{array} \right].$$

$$\left[ \begin{array}{l} par_2: \left[ \begin{array}{l} x: *Ind \vee Subst \\ i: Sch \end{array} \right] \\ \\ s_{msr}: \mathbf{Measure}(r.y, r.d, \left[ \begin{array}{l} p_{cbase} = \lambda r': [y: Ind]. [s_1: par_2.i(wine)(r'.y)]: Ppty \\ s_{app}: par_2.i(wine)(par_2.x) \\ par: \left[ \begin{array}{l} z: *Ind \\ i_v: Sch \end{array} \right] \\ s_{in}: \left[ \begin{array}{l} s_{app} : *par.i(glass)(par.z) \\ s_{within}: \mathbf{each.in}(par_2.y, par.z) \end{array} \right] \end{array} \right], R.n \end{array} \right] \quad (40)$$

This gives us the desired result, a function from real numbers  $n$  to a property of substances or pluralities of individuals such that they measure  $n$  with respect to a volume scale marked in terms of glass-sized portions of wine.

An important feature of this analysis is that all access to the container interpretation is lost as a result of applying *MSR* and the portion interpretation is embedded within the **Measure** relation: shifting to a measure interpretation such as *wine to the measure of two glasses* blocks access to glasses-as-containers. Thus we can explain the infelicity of (36-a) and (36-b). Furthermore, given that there is no direct access to the portion interpretation after applying the *MSR* function, we can also motivate why full pseudo-partitives (e.g. *two glasses of wine*) cannot participate in co-predication over their portion and measure interpretations, according to some native speakers at least, as we have seen in (35-a) and (35-b) above.

### [3.6] Mass-to-count coercion and measure interpretations

Having motivated our hypotheses (H1) and (H2), and also provided a formal analysis of the container (C), portion (P) and measure (M) interpretations, we now turn to our puzzle regarding coercion: Why is it easy to get coerced container (C) and portion (P) interpretations for expressions, such as *two wines*, but much harder to get measure (M) interpretations?

We will set out by briefly outlining what coercion is in the context of the count/mass distinction. It is triggered by a type mismatch which arises when a quantifier, a numerical or some expression of quantity selecting for a count noun is combined with a mass noun, or vice versa). To fix the type mismatch, the relevant mass or count noun must first shift into the relevant count or mass interpretation. For example, if a numerical expression *two* is directly combined with

*wines*, a mass noun in the plural, there is a type mismatch, because both  $\llbracket\text{two}\rrbracket$  and the plural morphology require a noun denoting a concept that specifies a disjoint counting base, but this is what  $\llbracket\text{wine}\rrbracket$  fails to do. The type mismatch may be resolved by inserting into the interpretation of *two wines* some individuating concept (e.g.  $\text{GLASS}$ ) that is recoverable from the context and which ‘apportions’ wine into the requisite disjoint set: e.g.  $\llbracket\text{two}\rrbracket(\text{GLASS}(\llbracket\text{wine}\rrbracket))$ .

On our analysis, see Section [3.4], the most straightforward interpretation of numerical expressions in cases such as *two wines* is as a numerical determiner, i.e.  $\text{NMOD}(\llbracket\text{two}\rrbracket)$ . However,  $\text{NMOD}(\llbracket\text{two}\rrbracket)(\llbracket\text{wine}\rrbracket)$  involves a type mismatch, since  $\llbracket\text{wine}\rrbracket$  does not specify a disjoint counting base. One possibility is to insert a classifier concept based on some receptacle(s) salient in the context, such as  $\text{CL}(\llbracket\text{glass}\rrbracket)$  (see also Section [3.4]), which resolves the type mismatch, making available the container or the portion interpretation based on  $\text{CL}(\llbracket\text{glass}\rrbracket)$ .<sup>4</sup>

Now for the answer to the main puzzle. There are two considerations that militate against getting the measure interpretation for a counting construction like *two wines*. One has to do with the way in which we derive the measure interpretation. If, as in the container and portion case, we interpret *two* as the numerical determiner  $\text{NMOD}(\llbracket\text{two}\rrbracket)$  (the usual interpretation for a numerical expression that directly modifies a noun), then even if we insert the relevant measure concept, namely,  $\text{MSR}(\text{CL}(\llbracket\text{glasses}\rrbracket))$  in order to repair the type mismatch between *two* and *wines*, there is still a type mismatch. This is because the measure interpretation requires that the measure concept combine with a numeral, such as  $\llbracket\text{two}\rrbracket$ , rather than with a numerical determiner, such as  $\text{NMOD}(\llbracket\text{two}\rrbracket)$  (following Rothstein (2011), i.a.).

The other consideration concerns general constraints on coercion and interpretive effort. It is standardly assumed that a type mismatch driven coercion arises when the properties (semantic type) of the *explicit* argument do not match the requirements of the *explicit* functor. However, coercion is not standardly assumed to be driven by a type mismatch between an *implicit* (contextually determined) functor and an *implicit* (contextually determined) argument. Now, the interpretation of counting constructions like *two white wines*, as in (2-a), (2-b), is commonly taken to involve recovering a suitable receptacle concept, e.g. the classifier-like  $\text{CL}(\llbracket\text{glass}\rrbracket)$  concept from the context. This resolves the type clash between a numerical expression and a mass noun, because it supplies the container interpretation for the wine (2-a) or an apportioning of the wine (2-b), both of which can be subjected to grammatical counting operations. Assuming that the dot type **container • portion** is a suitable type for receptacle nouns, such as *glass*, when used relationally, as well as for the corresponding contextually-determined classifier-like  $\text{CL}(\llbracket\text{glass}\rrbracket)$  concept, we should then expect the type clashes in (2-a)

[4] An alternative route to an interpretation, not discussed here, would be to shift  $\llbracket\text{wine}\rrbracket$  to a set of subkinds, as in *The off-license has over 100 wines in stock*.

and (2-b) be relatively easy to resolve. For instance, *carry* in (2-a) will select for a physical object of type **container**, while *drink* in (2-b) for **portion**, following the standard assumption that dot types allow for predications which are licensed over either of the two dot constitutive types.

The measure interpretation, however, would have to be derived by shifting the dot type **container • portion** (see (H1)) into a **measure** type, following our hypothesis (H2) that the measure meaning is derived from the portion one. This derivational step, on our account, is achieved by means of the *MSR* function. When it comes to examples like *two wines*, in order to get the coerced measure interpretation of say ‘wine to the measure of two glassfuls’, the *MSR* function would have to operate on an *implicit* portion meaning of the contextually-determined  $CL(\llbracket \text{glass} \rrbracket)$  concept, which is the result of coercion: ‘two glass-sized portions of wine’. In other words, *MSR* would have to apply to *implicit* linguistic material, which was previously inserted to repair a mismatch between *two* and *wines*, in order to yield the intended measure interpretation. Now, if an agent has already retrieved a classifier-like portion concept from the context in order to repair a type mismatch between *two* and *wines*, there is no type mismatch to fix, and no coercion is predicted to be available, if we understand coercion in the standard way. Neither is there any other impetus to trigger the application of the *MSR* function to the *implicit* portion meaning.

We can speculate that natural languages do not make such complex coercive operations available at least in part because interpreters find the reinterpretation of implicit linguistic material already used to repair e.g. the mismatch between a numerical determiner and a mass noun to be conceptually highly unwieldy. That is, once we coerce *two wines* to mean ‘two glass-sized portions of wine’, for instance, and thus resolve the ‘original’ type mismatch by means of the implicit classifier-like  $CL(\llbracket \text{glass} \rrbracket)$  concept, it is highly cognitively costly, if not impossible, to try to coerce this shifted portion interpretation into the abstract measure interpretation, which intuitively ‘detaches’ the measure from both its container and portion anchoring, in our case, from its anchoring to the classifier-like  $CL(\llbracket \text{glass} \rrbracket)$  concept, which is, albeit, implicit and recovered from context.

A key feature of our analysis is thus that the measure interpretation of pseudo-partitives is generally derived from the portion interpretation of classifier-like concepts, based on receptacle nouns (e.g. *glass*, *bottle*, *basket*, *pitcher*). Accounts which derive the measure interpretation directly from nouns that inherently denote receptacles do not have the means to proffer such an explanation.

#### SUMMARY AND CONCLUSIONS

We have introduced two types of data points not previously discussed in the literature, First, not all interpretations of pseudo-partitives formed with classifier-like expressions are equally felicitous under co-predication. This especially holds

of container (C) and measure (M) interpretations, as in (36-a) and (36-b). Second, when mass nouns, such as *wine*, are directly modified with numerical expressions, it is not easy to get a coerced measure (M) interpretation, e.g. *two wines* cannot (easily) mean something like ‘wine to the measure of two glasses-worth’.

To account for these data we have built upon and were inspired by the work of Rothstein and Landman. Our novel contribution mainly lies in showing that the measure interpretation of pseudo-partitives formed with classifier-like expressions like *glass* is derived from the portion interpretation, and not directly from their inherent non-classifier interpretation of physical receptacles pure and simple. That is to say that, instead of shifting  $\llbracket\text{glass}\rrbracket$  to a measure interpretation and then combining it with  $\llbracket\text{wine}\rrbracket$ , as Rothstein and Landman, for instance, propose, we argue that *glass* has an interpretation that is akin to a dot type for a container (C) and a portion (P) interpretation, and that the portion part of this meaning is shifted to a measure (M), namely, ‘wine to the measure of a glassful’. This difference between our account and others formed the basis for our explanation for why container and measure interpretations are not easily co-predicated over. Our account suggests the following partial order for felicity of combinations: C-P/P-C > M-P/P-M > C-M/M-C. When it comes to the interpretation of full pseudo-partitives, we argued that their classifier-like container and portion interpretations are default interpretations, and proposed that they are, therefore, of the dot type **container • portion**. This explains the felicity of C-P/P-C co-predication. The portion (P) and measure (M) interpretations can be accessed, by some people at least, by reconstructing the portion interpretation from the measure, as the measure MSR function contains the type for the portion in that the property for portion specifies the scale for the measure function for something like ‘glassful’ (see equation 40). The measure interpretation excludes access to the container interpretation, since, on our account, the type for the container is ‘lost’ as a result of shifting from the portion to the measure interpretation.

Finally, we used our proposal for full pseudo-partitive constructions to give us a window on mass-to-count coercion. On our analysis the measure interpretation is derived from the portion interpretation. This provided a reason why it is hard to access the measure interpretation when numerical expressions are directly combined with mass nouns. Finding a salient classifier in the context involves any type clash and so further shifting this classifier concept into a measure concept is not motivated (the original type clash is resolved), and is cognitively burdensome, since it would require reinterpreting implicit concepts.

#### ACKNOWLEDGMENTS

We would like to thank the participants the *Workshop on Approaches to Coercion and Polysemy* (CoPo 2017) for helpful feedback. Thanks, especially, to Robin Cooper and Keren Khrizman for useful discussion. This research was funded by the Ger-

man Research Foundation (DFG) as a part of the Collaborative Research Centre 991, project C09.

## REFERENCES

- Champollion, Lucas & Manfred Krifka. 2016. Mereology. In Maria Aloni (ed.), *The cambridge handbook of formal semantics*, 513–541. Cambridge University Press. doi:10.1017/cbo9781139236157.014.
- Chierchia, Gennaro. 2010. Mass nouns, vagueness and semantic variation. *Synthese* 174. 99–149. doi:10.1007/s11229-009-9686-6.
- Cooper, Robin. 2005. Records and record types in semantic theory. *Journal of Logic and Computation* 15(2). 99–112.
- Cooper, Robin. 2011. Copredication, quantification and frames. In Sylvain Pogodalla & Jean-Philippe Prost (eds.), *Logical aspects of computational linguistics, number 6736 in lecture notes in computer science*, 64–79. Springer.
- Cooper, Robin. 2012. Type theory and semantics in flux. In R. Kempson, T. Fernando & N. Asher (eds.), *Philosophy of linguistics, handbook of the philosophy of science*, 271–323. Elsevier.
- Cooper, Robin & Jonathan Ginzburg. 2015. Type theory with records for natural language semantics. In Shalom Lappin & Chris Fox (eds.), *Handbook of contemporary semantic theory, second edition*, 375–407. Wiley.
- Doetjes, Jenny. 1997. *Quantifiers and selection*. University of Leiden PhD dissertation.
- Filip, Hana & Peter R. Sutton. 2017. Singular count NPs in measure constructions. *Semantics and Linguistic Theory* 27. 340–357.
- Fillmore, Charles J. 1975. An alternative to checklist theories of meaning. *Proceedings of the First Annual Meeting of the Berkeley Linguistics Society* 1. 123–131.
- Fillmore, Charles J. 1976. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences* 280(1). 20–32.
- Khrizman, Keren, Fred Landman, Suzi Lima, Susan Rothstein & Brigitta R. Schvarcz. 2015. Portion readings are count readings, not measure readings. *Proceedings of the 20<sup>th</sup> Amsterdam Colloquium* 197–216.
- Krifka, Manfred. 1989. Nominal reference, temporal constitution and quantification in event semantics. In Renate Bartsch, J. F. A. K. van Benthem & P. van Emde Boas (eds.), *Semantics and contextual expression*, 75–115. Foris Publications.

- Landman, Fred. 2011. Count nouns – mass nouns – neat nouns – mess nouns. *The Baltic International Yearbook of Cognition* 6. 1–67.
- Landman, Fred. 2016. Iceberg semantics for count nouns and mass nouns: The evidence from portions. *The Baltic International Yearbook of Cognition Logic and Communication* 11. 1–48.
- Link, Godehard. 1983. The logical analysis of plurals and mass terms: A lattice-theoretic approach. In Rainer Bäuerle, Urs Egli & Arnim von Stechow (eds.), *Meaning, use and the interpretation of language*, 303–323. Berlin: de Gruyter.
- Partee, Barbara H. & Vladimir Borschev. 2012. Sortal, relational, and functional interpretations of nouns and russian container constructions. *Journal of Semantics* 29. 445–486.
- Pustejovsky, James. 1995. *The generative lexicon*. MIT Press.
- Pustejovsky, James. 2011. Coercion in a general theory of argument selection. *Linguistics* 49(6). 1401–1431.
- Rothstein, Susan. 2010. Counting and the mass/count distinction. *Journal of Semantics* 27(3). 343–397. doi:10.1093/jos/ffq007.
- Rothstein, Susan. 2011. Counting, measuring and the semantics of classifiers. *Baltic International Yearbook of Cognition, Logic and Communication* 6. 1–42.
- Rothstein, Susan. 2016. Counting, measuring and the semantics of classifiers. *Baltic International Yearbook of Cognition, Logic and Communication* 11. 1–42.
- Soja, Nancy, Susan Carey & Elizabeth Spelke. 1991. Ontological categories guide young children's inductions of word meaning: Object terms and substance terms. *Cognition* 38. 179–211. doi:10.1016/0010-0277(91)90051-5.
- Sutton, Peter & Hana Filip. 2016a. Counting in context: Count/mass variation and restrictions on coercion in collective artifact nouns. *Semantics and Linguistic Theory* 26. 350–370. doi:10.3765/salt.v26i0.3796.
- Sutton, Peter R. & Hana Filip. 2016b. Mass/count variation, a mereological, two-dimensional semantics. *The Baltic International Yearbook of Cognition Logic and Communication* 11. 1–45.
- Sutton, Peter R. & Hana Filip. 2017. Individuation, reliability, and the mass/count distinction. *Journal of Language Modelling* 5(2). 303–356.

## AUTHOR CONTACT INFORMATION

Peter R. Sutton

Heinrich Heine University Düsseldorf

[peter.r.sutton@icloud.com](mailto:peter.r.sutton@icloud.com)

Hana Filip

Heinrich Heine University Düsseldorf

[hana.filip@gmail.com](mailto:hana.filip@gmail.com)