

# Semantic Theory 2011, Exercise 7

## 1 Thematic roles and type theory

- (a) Sentences (1) and (2) can be represented by the type-logical expressions (1a) and (2a), where give' and receive' abbreviate the order-free  $\lambda$ -expressions from the slides. Expand the expressions and simplify step by step using  $\beta$ -reduction in its modified version.

(1) John gave Mary the book  
(2) Mary received the book from John

(1a)  $\text{give}'(\text{mary}'_{\text{rec}})(\text{the\_book}'_{\text{pat}})(\text{john}'_{\text{ag}})$

(2a)  $\text{receive}'(\text{the\_book}'_{\text{pat}})(\text{john}'_{\text{ag}})(\text{mary}'_{\text{rec}})(\text{john}'_{\text{pat}})$

Note: The representations are of type  $\langle e, t \rangle$ . Don't care about the binding of the event argument for now.

- (b) Give corresponding representations for the verbs sell' and buy'. Look up the appropriate frame-semantic role-names ("Frame Elements") in the FrameNet database (<http://framenet.icsi.berkeley.edu/> → "View data" → "lexical units"). Build representations of the sentences and reduce them as in (a).

(3) *John sold the car to Bill for 3,000€*

(4) *Bill bought the car from John for 3,000€*

## 2 Thematic roles and quantified NPs

So far, we have shown how roles can be integrated into a simple version of type-theoretic semantics with type  $e$  noun phrases, using order-free  $\lambda$ -abstraction with role indices. How can we model composition driven by role information with full NPs of type  $\langle \langle e, t \rangle, t \rangle$ ? Two thoughts:

First: The concept of order-free abstraction is motivated by the syntactic variation in the syntactic realization of the predicate-argument structure of verbs (as well as some nouns and adjectives). Other operators take their arguments in a way that is determined by syntactic structure. E.g., determiners consistently take the nominal expression under their sister node as their first argument. The finiteness/tense operator is applied in the last step of composition, taking an event predicate and returning the sentence representation. Therefore, it makes sense to combine unordered role-indexed with ordered, structure-driven abstraction. Determiners are defined in the usual way, event predicates may be given the following representation, where different "lambdas" indicate the order of application.

$$\lambda\{x_{\text{ag}}, y_{\text{pat}}, z_{\text{rec}}\} \lambda e. \text{give}(e) \wedge \text{ag}(e, x) \wedge \text{pat}(e, y) \wedge \text{rec}(e, z)$$

Second: The effect of applying full type  $\langle \langle e, t \rangle, t \rangle$  NPs to a relational expression consists in binding one argument, and „passing“ the rest of the  $\lambda$ -prefix to the outside across the quantifier.

In standard type-theoretic semantics, this mechanism is encoded in the representation of the verb:

watch  $\Rightarrow \lambda Q \lambda x [Q(\lambda y. \text{watch}(x, y))]$

watch a movie  $\Rightarrow \lambda Q \lambda x [Q(\lambda y. \text{watch}(x, y))] (\text{a-movie}')$

$\Leftrightarrow_{\beta} \lambda x [\text{a-movie}' (\lambda y. \text{watch}(x, y))]$

The NP representation *a-movie'* binds the object argument (*y*) and passes the rest of the prefix (in this case consisting of “ $\lambda x$ ” only) to the outside. Alternatively, we can model this binding effect by introducing a special composition operation of “argument binding”, which links an NP with a simple *n*-place relation:

$Q \otimes \lambda x_1 \dots \lambda x_n. R(x_1, \dots, x_n) : \Leftrightarrow \lambda x_2 \dots \lambda x_n. Q(\lambda x_1. R(x_1, \dots, x_n))$

Assuming the representation of *watch* to be just  $\lambda y \lambda x. \text{watch}(x, y) : \langle e, \langle e, t \rangle \rangle$ , binding operation with *a-movie'* gives  $\text{a-movie}' \otimes \lambda y \lambda x. \text{watch}(x, y)$ , which is equivalent to  $\lambda x [\text{a-movie}' (\lambda y. \text{watch}(x, y))]$

This treatment is particularly attractive in combination with role-driven composition. In this case, the operator is defined as:

$Q_i \otimes \lambda X. R[X] : \Leftrightarrow \lambda (X - \{x_i\}). Q(\lambda x_i. R[X])$

“ $R[X]$ ” expresses that the variables in *X* occur free in the expression.

Now consider sentence (5):

(5) Every student gave Mary a book.

(a) Expand the representation (5a), which uses the binding operator in the syntax-driven variant, and reduce as far as possible, using the definition of  $\otimes$  and  $\beta$ -reduction.

(5a) PAST(every-student'  $\otimes$  (a-book'  $\otimes$  (Mary'  $\otimes$  give')))

Assume  $\text{give}' := \lambda x \lambda y \lambda z \lambda e. \text{give}^*(e, x, y, z)$  as the representation of the verb *give* in the structure-driven version of semantic composition, assume the standard type  $\langle \langle e, t \rangle, t \rangle$  representations for the noun phrases and for the PAST operator.

(b) Do the same with representation (5b), the role-driven variant:

(5b) PAST(every-student'<sub>ag</sub>  $\otimes$  (a-book'<sub>pat</sub>  $\otimes$  (Mary'<sub>rec</sub>  $\otimes$  give')))

Assume that everything is the same as above, except that

$\text{give}' = \lambda \{x_{ag}, y_{pat}, z_{rec}\} \lambda e. \text{give}(e) \wedge \text{ag}(e, x) \wedge \text{pat}(e, y) \wedge \text{rec}(e, z)$