Implicit Arguments: Event Modification or Option Type Categories?

Chris Blom[1], Philippe de Groote[2], Yoad Winter[3], and Joost Zwarts[3]

[1] CAI Master Program, Utrecht University
[2] LORIA/INRIA
[3] UiL OTS, Utrecht University

Abstract. We propose a unified syntactic-semantic account of passive sentences and sentences with an unspecified object (*John read*). For both constructions, we employ *option types* for introducing implicit arguments into the syntactic-semantic categorial mechanism. We show the advantages of this approach over previous proposals in the domains of scope and unaccusatives. Unlike pure syntactic treatments, option types immediately derive the obligatory narrow scope of existential quantification over an implicit argument's slot. Unlike purely semantic, event-based treatments, our proposal naturally accounts for syntactic contrasts between passives and unaccusatives, as in *the door *(was) opened by John.*

1 Introduction

Many verbs allow adjacent constituents to be optional, as illustrated in (1).

(1) a. John read (the book).
   b. John introduced Paul (to Mary).
   c. The vase was broken (by John).

Unspecified objects (UOs, Levin [12]) as in (1a) are licensed with verbs like *eat, drink, bake* etc. In semantic terms, the optionality of object NPs with such verbs is similar to the optionality of subcategorized PPs as in (1b) and *by* phrases in passive sentences like (1c). When these constituents are missing the sentence still makes an existence claim with respect to the unfilled predicate slot [5]. This is illustrated by the equivalences in (2).

(2) John read $\Leftrightarrow$ John read something.
John introduced Paul $\Leftrightarrow$ John introduced Paul to someone.
The vase was broken $\Leftrightarrow$ The vase was broken by someone or something.

Because of this 'semantic implicitness' of the verbal slots in (2), we refer to the underlined constituents in (1) as optional *arguments.*[1]

Our aim in this paper is to give a basic syntactic-semantic account of optional arguments in categorial grammar and explain their semantic properties. The main two problems we deal with are therefore:

P1 How can an argument be missing while the sentence still makes an existential import involving the unfilled slot (2)?

P2 How can verbs with an optional argument compose with this argument when it is present, neutralizing the purely existential effect?

After reviewing some previous accounts and their limitations, we propose a new account of these two questions using *option types* in Abstract Categorial Grammar (ACG [6,13]).

2 Meaning Postulates and Event Modification

Existential implicit arguments must be interpreted with narrow scope (Fodor & Fodor, [5]). Consider the following examples.

(3) a. John did not read yesterday.
 b. John did not read something yesterday.

(4) a. The door was not opened yesterday.
 b. The door was not opened by something/someone yesterday.

The overt indefinites in (3b) and (4b) are scopally ambiguous. By contrast, in (3a) and (4a) the only reading is existential narrow scope: "John did not read anything" and "the door was not opened by anything/anyone", respectively. As Fodor & Fodor observe, obligatory narrow scope of quantifiers over implicit arguments challenges pure syntactic theories that introduce an existential argument at some covert syntactic level. F&F account for contrasts as in (3) using a meaning postulate connecting a transitive entry of UO verbs with another, intransitive entry. As F&F show, this immediately accounts for narrow scope effects in cases like (3a). A similar idea governs ambiguity-based derivations of passives, with or without a *by* phrase (Bach

[1] We will not try to give here a full syntactic analysis of the distinction between optional arguments and adjuncts. First, the disability to iterate (optional and obligatory) arguments (cf. **John ate/bought a sandwich a hamburger*) extends to other constituents that syntacticians may consider as adjuncts: **John went to Paul to Mary*, **John baked a cake for Mary for Sue*, **John cut the rope with a knife with an axe*, etc. Another challenge for defining the distribution of optional arguments is that some typical adjuncts may also have an existence entailment when missing. For instance, *John will sing* entails *John will sing sometime.* Note however that many adjuncts are clearly distinguished from optional arguments in having no existential entailment: *Mary worked* does not entail *Mary worked for someone*, *John baked a cake* does not entail *John baked a cake for someone*, *John went* does not entail *John went somewhere* (he may have evaporated in space), and *John cut the rope* does not entail *John cut the rope with something* (he may have had super-natural powers).

[1], Dowty [4]). However, this 'classical' approach to passives is quite cumbersome (Landman [9]). For instance, to account for the optionality of the two underlined constituents in a passive sentence such as *Paul was introduced* *by John* *to Mary*, classical accounts would have to assign four different meanings to the passive form of the verb *introduce*, connected to each other using meaning postulates.

A more promising semantic approach to passives and UOs was suggested by Carlson [2], who treats both *by* phrases in passives and objects with UO verbs as neo-Davidsonian event modifiers. Carlson treats the object in *John read 'Lolita'* as an adjunct that modifies reading events. When the object is missing (*John read*), Carlson treats the existence entailment as an implied property of events: when there is a reading, something must be read (cf. Larson [10], Lasersohn [11]). Similarly, Carlson treats the understood agent in a passive sentence like *Mary was praised* as a property of 'praising events': when there is a praising there is a praiser. These intuitions can be implemented as a single meaning postulate on predicates over events, which nicely improves F&F's approach. A similar event-based account of passives is proposed by Landman in [9].

However, Carlson's approach becomes less appealing when considering *unaccusative verbs*. A sentence like *the door opened* is not equivalent to *someone/something opened the door*. This shows that 'opening events' do not require an agent. Hence, the understood agent in passives like *the door was opened* is not explained.[2] Furthermore, treating *by* phrases as adjuncts does not account for their ungrammaticality with unaccusatives: if *by* phrases simply modify events by specifying an agent, why are they systematically disallowed in cases like **the door opened by John*? To block a meaning like "John opened the door" for such strings, Carlson (p.268) must resort to ad hoc syntactic assumptions.

3 Option Types in Abstract Categorial Grammar

The considerations in section 2 support a semantic approach that is more consistent with current syntactic theories about implicit arguments (Landau [8]). We propose the following principle (cf. Dekker [3]).[3]

[2] This is no problem for Landman's proposal, which uses the passive morphology for deriving the semantic requirement that an agent exists in events described by passive sentences. However for this reason, Landman's approach does not naturally extend to UOs, where intransitive forms are homonymic to transitives. Hence, in such situations overt morphology alone cannot account for the requirement that a patient/theme exists in all events described by UO verbs.

[3] Dekker proposes treating some optionality phenomena using dynamic binding. Extending Dekker's approach to passives and UOs may have comparable advantages to the present account. It would require augmenting Dekker's proposal with a clear distinction between verb modifiers (e.g. many adverbials), which can iterate, and optional arguments that cannot iterate (e.g. optional objects and *by* phrases in passives). In Dekker's system the former must be treated as dynamic argument modifiers, whereas the latter are treated as static argument saturators. Once this distinction is properly treated, we believe that Dekker's approach can be extended to an approach that is very similar to our account of optional arguments – see especially Dekker's remarks on this issue on [3, p.573].

Optional Argument Principle (OAP): *Verbal forms may specify optional arguments in their syntactic-semantic type. The existential semantic interpretation of an implicit argument is lexically introduced.*

The OAP allows us to account for the two problems mentioned above for Carlson's approach. First, the understood agent in passives like *the door was opened* is analyzed as a result of an optional argument. This argument can be materialized as a *by* phrase, but it also leads to the requirement that an agent exists when the *by* phrase is missing. By contrast, an unaccusative sentence like *the door opened* is analyzed, following Carlson, as a simple intransitive statement. The semantic (uni-directional) entailment from *the door was opened* to *the door opened* can be handled in event semantics, as in Carlson's account.[4] However, in our approach, unlike Carlson's proposal, there is no general strategy of *by* phrase adjunction. Thus, unaccusative sentences like *the door opened*, which are analyzed as intransitive sentences, do not license *by* phrases.

To implement OAP-based optionality in categorial grammar, we mark optionality by adding the symbol 'o' on the optional argument's type. For instance:

Intransitive verbs (*smile*, unaccusative *open*):
$\mathsf{np} \to \mathsf{s}$
Transitive verbs with obligatory object (*praise*, unergative *open*):
$\mathsf{np} \to (\mathsf{np} \to \mathsf{s})$
Transitive verbs with optional object (*read*):
$\mathsf{np}^{\mathsf{o}} \to (\mathsf{np} \to \mathsf{s})$
Passive forms of transitive verbs (*was read, was praised, was opened*):[5]
$\mathsf{np}^{\mathsf{o}}_{\mathsf{by}} \to (\mathsf{np} \to \mathsf{s})$

Option types are simple cases of *sum types* in functional programming [7]. There are two ways of filling in an optional slot of a function: by providing an argument of the appropriate type, or by using a universal filler, marked '$*$'. Let F be a function of type $\mathsf{a} \to \mathsf{b}$, with an obligatory argument slot of abstract type a. To make this slot into an optional one, an *option operator* distinguishes the case where the argument is present from usages of the filler. For the second case, we use a *default result*. Formally, an object x of type a^{o} is either of type a or the $*$-filler (disjoint from a's domain). Let d be a default result of type b. The **option** operator of type $(\mathsf{a}^{\mathsf{o}} \times (\mathsf{a} \to \mathsf{b}) \times \mathsf{b}) \to \mathsf{b}$ is defined by:

$\mathbf{option}(x, F, d) = d$ if $x = *$; otherwise $\mathbf{option}(x, F, d) = F(x)$.

The function $\lambda x.\mathbf{option}(x, F, d)$ is therefore of type $\mathsf{a}^{\mathsf{o}} \to \mathsf{b}$.

In ACG, the connection between the morpho-syntactic level and the semantic level is handled by mapping each simple *abstract type* (np, s etc.) to a morpho-syntactic type and a semantic type. We assume the simple morpho-syntactic type

[4] For space considerations we do not develop here this event-based account but refer the reader to [14], where an event semantics is handled within an ACG framework compatible with the current proposal.

[5] The abstract type $\mathsf{np}_{\mathsf{by}}$ is used for *by* phrases in passive constructions, and is needed for morpho-syntactic reasons alone (see section 5). The semantics of this type for the purposes of this paper is treated as equivalent to the semantics of np types.

f for *strings* and the simple semantic types e and t for *entities* and *truth-values* respectively. We assume the following mappings of the simple abstract type np and s to morpho-syntactic and semantic types:

$\mathsf{np} \mapsto \langle f, e \rangle$: a noun phrase surfaces as a string and denotes an entity

$\mathsf{s} \mapsto \langle f, t \rangle$: a sentence surfaces as a string and denotes a truth-value

Complex types like $\mathsf{np} \to \mathsf{s}$ (for intransitive verbs) or $\mathsf{np} \to (\mathsf{np} \to \mathsf{s})$ (for transitive verbs) are inductively mapped to the corresponding morpho-syntactic and semantic types according to the following rule.

Let a *be an abstract type that is mapped to the morpho-syntactic type* a_1 *and the semantic type* a_2*. Let* b *be an abstract type that is mapped to the morpho-syntactic type* b_1 *and the semantic type* b_2*. Then* $\mathsf{a} \to \mathsf{b}$ *is an abstract type that is mapped to the morpho-syntactic type* $a_1 \to b_1$ *and the semantic type* $a_2 \to b_2$*. Likewise,* a° *is an abstract type that is mapped to the morpho-syntactic type* a_1^o *and the semantic type* a_2^o*.*

In short we denote:

$\mathsf{a} \mapsto \langle a_1, a_2 \rangle$
$\mathsf{b} \mapsto \langle b_1, b_2 \rangle$
$(\mathsf{a} \to \mathsf{b}) \mapsto \langle a_1 b_1, a_2 b_2 \rangle$
$\mathsf{a}^\circ \mapsto \langle a_1^o, a_2^o \rangle$

For instance:

$(\mathsf{np} \to \mathsf{s}) \mapsto \langle ff, et \rangle$
: an intransitive verb surfaces as a function from strings to strings, and denotes a function from entities to truth-values

$(\mathsf{np} \to (\mathsf{np} \to \mathsf{s})) \mapsto \langle f(ff), e(et) \rangle$
: a transitive verb surfaces as a function from strings to functions from strings to strings, and denotes a function from entities to functions from entities to truth-values

Specifically, we assume the entry SMILE for the intransitive verb *smile*:

$\mathsf{SMILE} = \lambda s_f . s \bullet \mathit{smiled}_f \;:\; \mathbf{smile}_{et}$

In words: the morpho-syntactic entry for the verb *smile* is the function sending every subject string s to the corresponding sentence string composed of s and the string *smiled*; the corresponding semantic function is the function **smile** from entities to truth-values.

Similarly, we assume the following ACG entry PRAISE for the transitive verb *praise*:

$\mathsf{PRAISE} = \lambda o_f . \lambda s_f . s \bullet \mathit{praised}_f \bullet o \;:\; \mathbf{praise}_{e(et)}$

In words: the morpho-syntactic entry for the verb *praise* is the function sending every object string o to the function mapping every subject string s to the corresponding sentence string composed of o, s and the string *praised*; the corresponding semantic function is the function **praise** from entities to functions from entities to truth-values.

Simple sentences like *John smiled* and *John praised Lolita* are analyzed at the abstract level as $\texttt{SMILE}(\texttt{JOHN})$ and $\texttt{PRAISE}(\texttt{LOLITA})(\texttt{JOHN})$, respectively. Using the verbal entries above and the treatment of names as entity-denoting strings, these sentences are associated with the following strings and truth-values:

$\texttt{SMILE}(\texttt{JOHN}) = \textit{John} \bullet \textit{smiled} \;:\; \mathbf{smile}(\mathbf{john})$

$\texttt{PRAISE}(\texttt{LOLITA})(\texttt{JOHN}) = \textit{John} \bullet \textit{praised} \bullet \textit{Lolita} \;:\; \mathbf{praise}(\mathbf{lolita})(\mathbf{john})$

Using option types in ACG we extend this simple treatment to transitive verbs with an optional object. For instance, we assume the following ACG entry $\texttt{READ}$ for the transitive verb *read*, of abstract type $\mathsf{np}^{\circ} \to (\mathsf{np} \to \mathsf{s})$:

$$\begin{aligned} \texttt{READ} = {} & \lambda o_f.\lambda s_f.\mathbf{option}(o\;,\;\lambda u_f.s \bullet \textit{read}_f \bullet u\;,\; s \bullet \textit{read}) \;: \\ & \lambda o_e.\lambda s_e.\mathbf{option}(o\;,\;\lambda u_e.\mathbf{read}_{e(et)}(u)(s)\;,\;\exists x.\mathbf{read}(x)(s)) \end{aligned}$$

In words: the morpho-syntactic entry for the verb *read* uses the object string o in the object position if it is given, and leaves the object position null if the object argument is not given; the corresponding semantic function uses the object entity o as the first argument of the binary function **read**, and existentially saturates the position if the object argument is not given.

In this way, the sentences *John read* and *John read Lolita* are associated with the following strings and truth-values:

$\texttt{READ}(\texttt{JOHN}) = \textit{John} \bullet \textit{read} \;:\; \exists x.\mathbf{read}(x)(\mathbf{john})$

$\texttt{READ}(\texttt{LOLITA})(\texttt{JOHN}) = \textit{John} \bullet \textit{read} \bullet \textit{Lolita} \;:\; \mathbf{read}(\mathbf{lolita})(\mathbf{john})$

The treatment of passives is analogous. For instance, the passive form *was praised* (*by*) is analyzed as follows.

$$\begin{aligned} \texttt{PRAISE}_{pass} = {} & \lambda o_f.\lambda s_f.\mathbf{option}(o\;,\;\lambda u_f.s \bullet \textit{was-praised}_f \bullet u\;,\; s \bullet \textit{was-praised}_f) \;: \\ & \lambda o_e.\lambda s_e.\mathbf{option}(o\;,\;\lambda u_e.\mathbf{praise}_{e(et)}(s)(u)\;,\;\exists x.\mathbf{praise}(s)(x)) \end{aligned}$$

The sentences *Lolita was praised* and *Lolita was praised by John* are associated with the following strings and truth-values (for the definition of $\texttt{BY}$ see Table 1):

$\texttt{PRAISE}_{pass}(\texttt{LOLITA}) = \textit{Lolita} \bullet \textit{was-praised} \;:\; \exists x.\mathbf{praise}(\mathbf{lolita})(x)$

$\texttt{PRAISE}_{pass}(\texttt{BY}(\texttt{JOHN}))(\texttt{LOLITA}) = \textit{Lolita} \bullet \textit{was-praised} \bullet \textit{by} \bullet \textit{John} \;:\; \mathbf{praise}(\mathbf{lolita})(\mathbf{john})$

4 Some Exceptional Cases

There are some exceptional kinds of UO verbs and passive verbs that do not give rise to existential entailments when the object or *by* phrase is missing. Consider the following examples.

(5) a. John kicked $\overset{?}{\Rightarrow}$ John kicked something.

b. John bit $\overset{?}{\Rightarrow}$ John bit something.

(6) a. Mary was left alone $\overset{?}{\Rightarrow}$ Someone left Mary alone. [2].

b. The traps were avoided $\overset{?}{\Rightarrow}$ Someone avoided the traps. [11]

Similar cases were used by Carlson [2] to argue for event-oriented modification by objects and *by* phrases. Carlson assumes that since existence entailments do not appear with some such cases, this motivates an ontology-based account of the existence entailments that as a rule appear with these constructions. In addition to the reasons we gave in section 2 for doubting Carlson's general approach to UOs and passives, it should be remarked that lexical UOs like *kick* and *bite* are rather rare.[6] Because of their rarity, the UO verb alternations that such transitive verbs show in English may be a result of an accidental lexical ambiguity/polysemy, rather than a general process, as Carlson assumes. With passives as in (6), it is even more unclear whether there is any need to assume a general non-existential strategy for passives. Note that both sentences in (6) involve a negative component for the verbal element (*alone* = with no one, *avoid* = not get close to). Arguably, the non-existential effect is due to an existential reading of the agent argument which is in the scope of these negative components. Thus, examples such as (6) do not provide evidence for "non-existential passives".

5 A General Strategy of 'Optionalization'

So far in this paper we have exemplified specific entries of various option types and the way they are interpreted. We would now like to define a general procedure of 'optionalization'. Thus, given an ACG lexicon $\mathcal{L}$ without option types, we would like to transform $\mathcal{L}$ into an lexicon $OPT(\mathcal{L})$ with option types. This is done using a specification of some of the arguments of entries in $\mathcal{L}$ as optional arguments and using this specification for systematically optionalizing $\mathcal{L}$.

Consider the optionality-free toy ACG verbal lexicon in Table 1.[7] This lexicon contains the verbal forms discussed in section 3, but without any treatment of optionality: all arguments are treated as obligatory, against common linguistic judgements about the verbal arguments that are underlined in Table 1. Our general method transforms such a description into a proper ACG lexicon with types for optional arguments and the **option** operator in entry values as required by their types. This general method guarantees an economical representation for the two features common to all the optional arguments treated: no effect of the presence or lack of an optional argument on the *form* of the verb; and narrow scope existential quantification over the semantic slot in the verb's *meaning.*

Our general procedure for mapping such an option-free lexicon $\mathcal{L}$ to the corresponding optionalized lexicon uses option-free entries of abstract type $\underline{\mathsf{a}} \to \mathsf{b}$, where the a argument is targeted as requiring optionality. Our general procedure of 'optionalization' maps such an entry to a lexicon entry of abstract type $\mathsf{a}^\circ \to \mathsf{b}$, with the proper morpho-syntactic and semantic treatment. For instance, the entry READ in Table 1 does not respect the optionality of the verb's object argument. However, the procedure below will guarantee that the underlined argument in

[6] In her extensive typology of verb alternations in English, Levin [12] characterizes verbs such as *kick* or *bite* as showing "characteristic property alternations". This is because of sentences like *our horse kicks* or *our dog bites regularly*, which indicate a tendency towards kicking or biting. Levin does not mention simple past tense sentences as in (5), where transitive verbs without objects give rise to episodic readings.

[7] To achieve a more conspicuous notation we avoid parentheses in this table, reading the type notation $\alpha \to \beta \to \gamma$ with right-association as in $\alpha \to (\beta \to \gamma)$.

Table 1. Toy ACG lexicon without option types

Entry name	Type	Value
BY	$\mathsf{np}\to\mathsf{np}_{\mathsf{by}}$	$\lambda x_f.by \bullet x \;:\; \lambda x_e.x$
TO	$\mathsf{np}\to\mathsf{np}_{\mathsf{to}}$	$\lambda x_f.to \bullet x \;:\; \lambda x_e.x$
SMILE	$\mathsf{np}\to\mathsf{s}$	$\lambda s_f.s \bullet \mathit{smiled}_f \;:\; \mathbf{smile}_{et}$
PRAISE	$\mathsf{np}\to\mathsf{np}\to\mathsf{s}$	$\lambda o_f.\lambda s_f.s \bullet \mathit{praised}_f \bullet o : \mathbf{praise}_{eet}$
PRAISE$_{pass}$	$\underline{\mathsf{np}_{\mathsf{by}}}\to\mathsf{np}\to\mathsf{s}$	$\lambda b_f.\lambda s_f.s \bullet \mathit{was\text{-}praised}_f \bullet b \;:\; \lambda x_e.\lambda y_e.\mathbf{praise}(y)(x)$
READ	$\underline{\mathsf{np}}\to\mathsf{np}\to\mathsf{s}$	$\lambda o_f.\lambda s_f.s \bullet \mathit{read}_f \bullet o \;:\; \mathbf{read}_{eet}$
READ$_{pass}$	$\underline{\mathsf{np}_{\mathsf{by}}}\to\mathsf{np}\to\mathsf{s}$	$\lambda b_f.\lambda s_f.s \bullet \mathit{was\text{-}read}_f \bullet b \;:\; \lambda x_e.\lambda y_e.\mathbf{read}_{eet}(y)(x)$
INTRODUCE	$\underline{\mathsf{np}_{\mathsf{to}}}\to\mathsf{np}\to\mathsf{np}\to\mathsf{s}$	$\lambda t_f.\lambda o_f.\lambda s_f.s \bullet \mathit{introduced}_f \bullet o \bullet t : \mathbf{introduce}_{eeet}$
INTRODUCE$_{pass}$	$\underline{\mathsf{np}_{\mathsf{to}}}\to\underline{\mathsf{np}_{\mathsf{by}}}\to\mathsf{np}\to\mathsf{s}$	$\lambda t_f.\lambda b_f.\lambda s_f.s \bullet \mathit{was\text{-}introduced}_f \bullet b \bullet t \;:$ $\lambda r_e.\lambda a_e.\lambda p_e.\mathbf{introduce}(r)(p)(a)$

the type $\underline{\mathsf{np}}\to\mathsf{np}\to\mathsf{s}$ will be properly treated as optional in the resulting lexicon, with an entry for the same verb of type $\mathsf{np}^{\circ}\to\mathsf{np}\to\mathsf{s}$, and the appropriate values in its morpho-syntactic and semantic treatment.

Let $\varphi = \varphi_1 : \varphi_2$ be a lexical entry of abstract type $\mathsf{a} \to \mathsf{b}$, where the argument type a is defined as desirably optional. Such entries are marked $\underline{\mathsf{a}} \to \mathsf{b}$ in Table 1. In order to define optionalization inductively, we assume that the abstract types a and b have no underlined sub-parts. To capture correctly the behavior of optional arguments, we assume further that these types satisfy the following restrictions:

- The concrete morpho-syntactic type of a is string:
  $\mathsf{a} \mapsto \langle f, \tau_{\mathsf{a}}\rangle$.
- The concrete semantic type of b is boolean:
  $\mathsf{b} \mapsto \langle \sigma_{\mathsf{b}}, \tau_{\mathsf{b}}\rangle$, where τ_{b} is a boolean type.[8]

The 'string' requirement is needed to allow filling in optional slots by the empty string; the 'boolean' requirement is needed to allow existential quantification over morpho-syntactically empty slots.

Given these assumptions on the type of the entry φ, we define the corresponding optionalized entry $opt(\varphi) = \varphi_1' : \varphi_2'$ of type $\mathsf{a}^{\circ} \to \mathsf{b}$ as follows:

$\varphi_1' = \lambda x_{f^{\circ}}.\mathbf{option}(x, \varphi_1, \varphi_1(\epsilon))$

$\varphi_2' = \lambda x_{\tau_{\mathsf{a}}{}^{\circ}}.\mathbf{option}(x, \varphi_2, \mathit{CLOS}(\varphi_2))$

The operator CLOS existentially saturates the first argument of φ_2, which is inductively defined as follows:

If $\tau_{\mathsf{b}} = t$ then $\mathit{CLOS}(\varphi_2) = \exists x_{\tau_{\mathsf{a}}}.\varphi_2(x)$.

Otherwise, let $\tau_{\mathsf{b}} = \tau_1\tau_2$, where τ_2 is boolean by induction. We define:

$\mathit{CLOS}(\varphi_2) = \lambda y_{\tau_1}.\mathit{CLOS}(\lambda x_{\tau_{\mathsf{a}}}.\varphi_2(x)(y))$.

For instance, for the constant **read** of type $e(et)$, we have:

$\mathit{CLOS}(\mathbf{read}) = \lambda y_e.\mathit{CLOS}(\lambda x_e.\mathbf{read}(x)(y)) = \lambda y.\exists x.\mathbf{read}(x)(y)$

[8] Standardly, we define the set of 'boolean' semantic types as the smallest set of semantic types that contains t and any type $\tau_1\tau_2$ where τ_2 is boolean.

As a result, the optionalized version of the entry for READ in Table 1 is $opt(\mathsf{READ}) = \varphi'_1 : \varphi'_2$ s.t.:

$$\begin{aligned}
\varphi'_1 &= \lambda x_{f^o}.\mathbf{option}(x,\ \lambda o_f.\lambda s_f.s \bullet read_f \bullet o,\ \lambda s_f.s \bullet read_f \bullet \epsilon)\\
\varphi'_2 &= \lambda x_{e^o}.\mathbf{option}(x,\ \mathbf{read}_{eet},\ CLOS(\mathbf{read}_{eet}))\\
&= \lambda x_{e^o}.\mathbf{option}(x,\ \mathbf{read}_{eet},\ \lambda y.\exists x'.\mathbf{read}(x')(y))
\end{aligned}$$

The definition of the *opt* operator is not yet sufficient in order to optionalize entries of abstract type $\mathsf{a} \to \mathsf{b}$ where b has underlined types in it. This is necessary, as illustrated by the entry $\mathsf{INTRODUCE}_{pass}$ in Table 1, representing the form and meaning of the verbal passive *be introduced* (*by X to Y*), which has two optional arguments. To allow optionalization of such entries as well, let $\varphi = \varphi_1 : \varphi_2$ be a lexical entry of an abstract type X. The type X may be primitive. In case $\mathsf{X} = \mathsf{a} \to \mathsf{b}$, we adopt the same assumptions above on a (string morpho-syntactic type) and b (boolean semantic type), but now b possibly has underlined types in it. Given these assumptions, we inductively define the optionalized entry $OPT(\varphi)$:

If X is primitive, then the entry $OPT(\varphi) = \varphi$.
If $\mathsf{X} = \mathsf{a} \to \mathsf{b}$ (the argument a should not be optionalized) , then the optionalized entry is inductively defined by $OPT(\varphi) = \lambda x_{\mathsf{a}}.OPT(\varphi(x))$.
If $\mathsf{X} = \underline{\mathsf{a}} \to \mathsf{b}$ (the argument a should be optionalized) , then the optionalized entry is inductively defined by $opt(\lambda x_{\mathsf{a}}.OPT(\varphi(x)))$.

For instance, applying *OPT* to the entry $\mathsf{INTRODUCE}_{pass}$ in Table 1, of type $\underline{\mathsf{np_{to}}} \to \underline{\mathsf{np_{by}}} \to \mathsf{np} \to \mathsf{s}$ (first two arguments are optional), results in:

$$\begin{aligned}
&OPT(\mathsf{INTRODUCE}_{pass})\\
&= opt(\lambda x_{np_{to}}.OPT(\mathsf{INTRODUCE}_{pass}(x)))\\
&= opt(\lambda x_{np_{to}}.opt(\lambda y_{np_{by}}.OPT(\mathsf{INTRODUCE}_{pass}(x)(y))))\\
&= opt(\lambda x_{np_{to}}.opt(\lambda y_{np_{by}}.\mathsf{INTRODUCE}_{pass}(x)(y)))
\end{aligned}$$

In the morpho-syntactic level, this amounts to:

$$\begin{aligned}
&opt(\lambda x_f.opt(\lambda y_f.(\lambda t_f.\lambda b_f.\lambda s_f.s \bullet was\text{-}introduced_f \bullet b \bullet t)(x)(y)))\\
&= opt(\lambda x_f.opt(\lambda y_f.\lambda s_f.s \bullet was\text{-}introduced_f \bullet y \bullet x))\\
&= \lambda w_f.\mathbf{option}(w\ , \lambda x_f.\ \lambda z_f.\mathbf{option}(z, \lambda y_f.\ \lambda s_f.s \bullet was\text{-}introduced_f \bullet y \bullet x\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad ,\quad \lambda s_f.s \bullet was\text{-}introduced_f \bullet \epsilon \bullet x\ )\\
&\qquad\qquad\quad ,\quad \lambda z_f.\mathbf{option}(z, \lambda y_f.\ \lambda s_f.s \bullet was\text{-}introduced_f \bullet y \bullet \epsilon\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad ,\quad \lambda s_f.s \bullet was\text{-}introduced_f \bullet \epsilon \bullet \epsilon\ ))
\end{aligned}$$

This covers the four combinations of present/missing morpho-syntactic arguments. Similarly, in the semantic level we have:

$$\begin{aligned}
&opt(\lambda x.opt(\lambda y_e.(\lambda r_e.\lambda a_e.\lambda p_e.\mathbf{introduce}_{e(e(et))}(r)(p)(a))(x)(y)))\\
&= opt(\lambda x_e.\lambda z.\mathbf{option}(z\ , \lambda y_e.\lambda p_e.\mathbf{introduce}_{e(e(et))}(x)(p)(y)\\
&\qquad\qquad\qquad\qquad , CLOS(\lambda y_e.\lambda p_e.\mathbf{introduce}_{e(e(et))}(x)(p)(y))))
\end{aligned}$$

And applying *CLOS* gives:

$$\begin{aligned}
&opt(\lambda x_e.\lambda z.\mathbf{option}(z\,, \lambda y_e.\lambda p_e.\mathbf{introduce}_{e(e(et))}(x)(p)(y)\\
&\qquad\qquad\qquad\quad , \lambda p_e.\exists y_e.\mathbf{introduce}_{e(e(et))}(x)(p)(y)))
\end{aligned}$$

$$\begin{aligned} = \lambda w_{e^o}.\mathbf{option}(w\, , \lambda x_e.\lambda z_{e^o}.\mathbf{option}(z\, , &\lambda y_e.\lambda p_e.\mathbf{introduce}_{e(e(et))}(x)(p)(y) \\ &, \lambda p_e.\exists y_e.\mathbf{introduce}_{e(e(et))}(x)(p)(y)) \\ , \lambda z_{e^o}.\exists x_e.\mathbf{option}(z\, , &\lambda y_e.\lambda p_e.\mathbf{introduce}_{e(e(et))}(x)(p)(y) \\ &, \lambda p_e.\exists y_e.\mathbf{introduce}_{e(e(et))}(x)(p)(y))) \end{aligned}$$

This covers the four combinations of the two present/existentially-closed semantic arguments. The same procedure also correctly derives the other specific entries discussed in section 3.

6 Conclusions

Optionality with verbal arguments is a well-known phenomenon that is treated in one way or another by most syntactic theories. We started out by pointing out the narrow-scope behavior of existential quantifiers over empty argument slots, which, although familiar, is not treated systematically by syntactic frameworks we are aware of. The behavior of unaccusative verbs in contrast to their passive forms has led us to conclude that *by* phrases in passives should be treated similarly to optional arguments, and not using event-modifiers. The Optional Argument Principle aims to describe the relations between the syntactic optionality of arguments and their semantic properties. Within the framework of Abstract Categorial Grammar (ACG), we introduced a standard interpretation of option types, which led to a general transformation of grammars without argument optionality to grammars that encode it in a uniform way.

Acknowledgements. We thank Marijana Marelj for helpful discussions. The work was partially supported by a Van Gogh grant to the 2nd and 3rd author by the French-Dutch Academy. The work of the 3rd author was partially supported by a VICI grant number 277-80-002 by the Netherlands Organisation for Scientific Research (NWO). The work of the 4th author was supported by an NWO grant number 360-70-340.

References

1. Bach, E.: In defense of passive. Linguistics and Philosophy 3, 297–341 (1980)
2. Carlson, G.: Thematic roles and their role in semantic interpretation. Linguistics 22, 259–279 (1984)
3. Dekker, P.: Existential disclosure. Linguistics and Philosophy 16, 561–588 (1993)
4. Dowty, D.: Grammatical relations and Montague Grammar. In: Jacobson, P., Pullum, G. (eds.) The Nature of Syntactic Representation. Kluwer, Dordrecht (1982)
5. Fodor, J.A., Fodor, J.D.: Functional structure, quantifiers, and meaning postulates. Linguistic Inquiry 11, 759–770 (1980)
6. de Groote, P.: Towards abstract categorial grammars. In: Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, ACL (2001)
7. Gunter, C.: Semantics of programming languages: structures and techniques. The MIT Press (1992)
8. Landau, I.: The explicit syntax of implicit arguments. Linguistic Inquiry 41, 357–388 (2010)

9. Landman, F.: Events and Plurality: the Jerusalem lectures. Kluwer, Dordrecht (2000)
10. Larson, R.K.: Events and modification in nominals. In: Proceedings of Semantics and Linguistic Theory, SALT8, pp. 145–168 (1998)
11. Lasersohn, P.: Lexical distributivity and implicit arguments. In: Proceedings of Semantics and Linguistic Theory, SALT3 (1993)
12. Levin, B.: English Verb Classes and Alternations. The University of Chicago Press, Chicago (1993)
13. Muskens, R.: Language, Lambdas, and Logic. In: Kruijff, G.J., Oehrle, R. (eds.) Resource Sensitivity in Binding and Anaphora, pp. 23–54. Kluwer (2003)
14. Winter, Y., Zwarts, J.: Event Semantics and Abstract Categorial Grammar. In: Kanazawa, M., Kornai, A., Kracht, M., Seki, H. (eds.) MOL 12. LNCS, vol. 6878, pp. 174–191. Springer, Heidelberg (2011)