Type-Coercion in Contexts of Desire

Johan Bos
Bologna, Italy
bos@meaningfactory.com

Abstract

A computational lexicon should have generative mechanisms in order to deal with the creativity and flexibility known from natural language. I present a large-scale semantic lexicon based on combinatory categorial grammar (CCG) with a model-theoretic interpretation based on discourse representation theory (DRT), and discuss how to incorporate type-coercion for verbs expressing contexts of desire, such as *want*. I compare functor coercion with strong and weak argument coercion, and also show how to incorporate information on tense and presupposition.

1 Introduction

The statistical revolution in computational linguistics has given us wide-coverage grammars with an open view on the lexicon, in contrast to the classic, manually crafted grammars, that have a closed view on the lexicon. Even though the lexical categories and their semantic content in a statistically induced grammar are fixed, their instantiations aren't, and can be created on the fly during parsing, ensuring robustness. Hence, an issue of importance is to find out whether this view of natural language engineering is compatible with the generative power of the lexicon (Pustejovsky, 1995). After all, statistical parsing and the generative lexicon share the goal of being able to deal with creativity and flexibility found in natural language. But in practice, there are a number of stumbling blocks. There isn't as much linguistic detail in statistically derived grammars as in handcrafted ones required for generative lexica. And mechanisms such as type coercion are likely to make the process of syntactic parsing and semantic interpretation harder.

In this article, I will take a wide-coverage grammar and a statistical parser, and concentrate on how a well-known generative device, type coercion, could be integrated in the interpretation process. I will focus on a class of verbs that show different syntactic subcategorisation frames but nevertheless express similar meanings, as illustrated by *want* in (1)–(3) below:

(1) No politician wants to say what I just said.

(2) They want a 24-hour library.

(3) Mr. Bush wants perestroika to succeed.

Like Pustejovsky (1995), I would like the semantics to be similar in the above examples, as they all introduce "contexts of desire" of propositional type, but can occur with VPs or NPs as complement too. In particular, providing a proper analysis for the NP-complement case of *want* in (2) is challenging, and I discuss how coercion mechanisms can account for it. Unlike Pustejovsky (1995), I also discuss three further subcategorisation frames of *want*, of which the latter is significantly different from the three cases listed above: (4), with a passive construction as complement, (5), with an adjectival phrase as complement, and (6) where *want* seems to have an NP-PP pair complement.

(4) The officials wanted him replaced.

(5) They want psyllium prices low for their purchases next year.

(6) He wants $44 million in the budget next year.

Even though it is true that these cases aren't common in corpora such as the Wall Street Journal, the current accounts nonetheless fail to provide an adequate analysis for them. My aim is to account for these phenomena with the help of a categorial grammar (CCG), with a lexicon based on CCG-bank. As semantic theory I take Discourse Representation Theory. I will first introduce the theoretical framework and its practical implementation.

2 Background

2.1 Combinatory Categorial Grammar, CCG

CCG is a lexicalised theory of grammar in which all syntactic dependencies are coded in the lexical categories (Steedman, 2001). The version of CCG that I adopt is based on CCGbank (Hockenmaier, 2003), a set of CCG derivations derived from the Wall Street Journal texts from the Penn Treebank (Marcus et al., 1993). The basic categories are `S` (sentence), `N` (noun), `PP` (prepositional phrase), `NP` (noun phrase) and `T` (text).

Functor categories are composed out of the basic categories with the help of slashes indicating order of and position of arguments: a functor category $\alpha\backslash\beta$ yields a category α when it finds an argument of category β on its left, and a functor category α/β yields a category α when it finds an argument of category β on its right. An example of a functor category is `(N\N)/NP` for the preposition *in*, as shown in Figure 1.

Following a convention introduced in CCGbank, the category `S` is associated with a feature indicating sentence mood or the different forms of verb phrases. I use a colon to combine a category with a feature, and so we have `S:dcl` for declarative sentences, `S:whq` and `S:ynq` for questions, `S:pss\np` for passive verb phrases, `S:b\np` for base clauses, `S:to\np` for infinitival clauses, and so on (see Hockenmaier (2003)).

To combine categories deriving new categories, CCG is equipped with a set of combinatory rules and a set of non-combinatory rules. The combinatory rules combine two categories producing a new one. At our disposal we have forward application (>), backward application (<), forward and backward composition (>B and <B), forward and backward substitution (>S and <S), their crossing variants, and generalised versions of the composition rules. All of these rules have a direct semantic interpretation, and give expressive power that go beyond context free grammars (Steedman, 2001).

The non-combinatory rules consist of the type-raising and type-changing rules. They are unary rules, mapping a single category into another single category. Here we have forward and backward type-raising (>T and <T), which are rules that change an argument in a functor in a systematic way. For example, a category of `NP` can be raised to `S\(S/NP)` using the >T rule. The type-raising rules, too, have a direct semantic interpretation.

The type-changing rules, however, are irregular and mostly motivated in CCGbank from a practical point of view, because they cut down the number of lexical categories. As a matter of fact, all type-changing rules can be eliminated from the system by inflating the number of lexical categories. One of the most frequent type-changing rules is `NP`$\rightarrow$`N`, changing a noun into a noun phrase; most other rules change verbal clauses into modifiers (Hockenmaier, 2003). I will indicate type-changing rules in derivations using L, and a first example of a type-changing rule in action can be seen in Figure 1. From the perspective of interpretation, type-changing rules aren't convenient, as each particular instance of a type-changing rule has it own semantic interpretation.

2.2 Discourse Representation Theory (DRT)

DRT is a formal theory of text meaning based on model-theoretic semantics (Kamp and Reyle, 1993), with three components: (1) a formal language of Discourse Representation Structure (DRS), the meaning representations for texts; (2) a component that deals with the semantic interpretation of DRSs; and (3) a component that translates text into DRS, i.e., the syntax-semantics interface.

A key idea of DRT is that a DRS plays both the role of content (giving a precise model-theoretic interpretation of the text processed so far) and context (assisting in interpreting anaphoric expression occurring in subsequent text). A DRS consists of a set of discourse referents and a set of DRS-conditions. DRS-conditions can either be basic, storing information about the discouse referents or relations between them, or complex, containing embedded DRSs. Hence DRSs are recursively defined, and the way they are nested predicts which discourse referents are accessible for future anaphoric reference and which are not. The core of the DRS langauge can be translated into first-order logic; alternatively it can be given a direct model-theoretic interpretation.

The syntax-semantics interface employed in my version of DRT is based on combining the DRS language with machinery of type-theory. This enables us to combine it almost straightforwardly with a CCG grammar and gives us a convenient compositional semantics. The basic semantic types in our inventory are e (individuals) and t (truth value). The set of all types is recursively defined in the usual way: if τ_1 and τ_2 are types, then so is $\langle\tau_1, \tau_2\rangle$, and nothing except the basic types or what can be constructed via the recursive rule are types. Expressions of type e are either discourse referents, or variables. Expressions of type t are either basic DRSs, DRSs composed with the

```
Mr. Stronach wants                to                 resume      an   influential role in       the  company
--- -------- -------------------- ------------------ ----------- ---- ----------- ---- -------- ---- -------
N/N N        (S:dcl\NP)/(S:to\NP) (S:to\NP)/(S:b\NP) (S:b\NP)/NP NP/N N/N         N    (N\N)/NP NP/N N
----------->                                                          --------------->          ----------->
N                                                                     N                         NP
-----------L                                                                           -------------------->
NP                                                                                     N\N
                                                                      -------------------------------------<
                                                                      N
                                                                 ------------------------------------------>
                                                                 NP
                                                    ------------------------------------------------------->
                                                    S:b\NP
                                    ----------------------------------------------------------------------->
                                    S:to\NP
             ---------------------------------------------------------------------------------------------->
             S:dcl\NP
-----------------------------------------------------------------------------------------------------------<
S:dcl
```

Figure 1: CCG derivation (featuring *want*, subject control).

merge (;), or DRSs formed by function application (@), but not variables. The complex types correspond either to variables, λ-abstracted partial DRSs, or function application. An example of a partial DRS is shown in Figure 2.

Following CCG's type transparency principle, the CCG categories maps systematically onto the semantic types: $\mathtt{N} \mapsto \langle e,t \rangle$, $\mathtt{PP} \mapsto \langle e,t \rangle$, $\mathtt{NP} \mapsto \langle\langle e,t \rangle,t\rangle$, $\mathtt{S} \mapsto \langle\langle e,t \rangle,t\rangle$, and $\mathtt{T} \mapsto t$. In order to incorporate a neo-Davidsonian approach to events in a compositional setting, I advocate the use of the *continuation* approach (Bos, 2009), motivating the type $\langle\langle e,t \rangle,t\rangle$ for the category $\mathtt{S}$. The continuation approach introduces a discourse referent for an event in the lexical entry for a verb, and sub-categorises for a potential modifier that could occur in a sentence by providing a "free slot" in the form of a λ-bound variable functionally applied to event discourse referent. Any actual modifier can absorb this slot and introduce a new "free slot". When the end of the sentence is reached, or when a sentential complement is completed, the free slot is filled with an empty DRS (as in Figure 2) or with a DRS containing tense information (as in Figure 9).

I will present DRSs in their familiar box notation and visualise discourse referents reflecting their place in the underlying sortal hierarchy: x_i for individuals, e_i for events, c_i for contexts, and t_i for temporal intervals, where i is an index (sometimes omitted). An example of a DRS is (8).

2.3 Implementation

The theortical framework introduced comes with a practical implementation. It consists of a statistical parser trained on CCGbank, which is integrated with a set of taggers (Curran et al., 2007). These tools produce CCG derivations decorated with information on part of speech and named entities, the basis of which the Boxer system for semantic interpretation computes DRSs in the manner described in the previous section (Bos, 2008).

The large-scale semantic lexicon of Boxer pairs each lexical CCG category with a generic proper partial DRS, obeying the semantic type restrictions. A generic partial DRS can be overwritten by a more specific one if information is made available by the parser referring to part of speech, named entity tags, or the tokens (words) themselves. For instance: a token that is assigned $\mathtt{N/N}$ is mapped to a generic partial DRS for noun modifiers, but when the token has been assigned a part of speech tag for superlative adjectives, it will be mapped to a partial DRS with appropriate semantics for modelling superlatives. If the token equals the string *other*, it will be mapped to a partial DRS introducing a presupposition, and so on.

The overall system achieves high coverage on newswire texts (about 99% on the Wall Street Journal texts of the Penn Treebank) and is therefore appropriate to use in practical, open-domain applications (Curran et al., 2007).

3 Modelling Contexts of Desire

I will motivate the need for type coercion in contexts of desire, based on variants of the proposal by Pustejovsky (1995). The focus of my analysis are verbs such as *want*, *demand*, *expect*, and *need*. These verbs have in common that they show several different syntactic subcategorisation frames, but don't seem to differ semantically across these different patterns (Partee, 1974). In order to account for this discrepancy between syntax and semantics, I discuss how type-coercion techniques as proposed by Pustejovsky (1995) can be introduced in my combined framework of CCG and DRT, contrasting two different approaches: argument and function coercion. For explanatory purposes I will provide a detailed discussion only of the verb *want*.

Table 1: CCG categories for *want* CCGbank.

Comp.	CCG Category	Freq	%
VP	`(S:X\NP)/(S:to\NP)`	390	62.70
NP	`(S:X\NP)/NP`	143	22.99
NP+VP	`((S:X\NP)/(S:to\NP))/NP`	51	8.20
	`((S:X\NP)/(S:pss\NP))/NP`	9	1.45
	`((S:X\NP)/(S:adj\NP))/NP`	3	0.48
NP+PP	`((S:X\NP)/PP)/NP`	10	1.61

In CCGbank the tokens *want*, *wants*, *wanted* and *wanting* are associated with various different lexical categories. Table 1 lists the four most common complement classes with their CCG categories, each of them corresponding to the examples (1)–(6) respectively. I will discuss the lexical semantics for these four complement classes below.

3.1 VP-Complement

By far the most frequent lexical category is where *want* subcategorises for a infinitival verb phrase in a subject control configuration, making up for almost two-third of the uses of *want* in CCGbank (Table 1). An example of a sentence illustrating this use of *want* is the earlier (1), or (7) below, which derivation in CCG is shown in Figure 1, and the DRS produced for this derivation in (8).

(7) Mr. Stronach wants to resume an influential role in the company.

(8) ([x_1 x_2 | named(x_1,mr) named(x_1,stronach) company(x_2)] ; [e_3 c_4 | want(e_3) agent(e_3,x_1) theme(e_3,c_4) c_4: [x_5 e_6 | influential(x_5) role(x_5) in(x_5,x_2) retain(e_6) agent(e_6,x_1) theme(e_6,x_5)]])

Here, and in the following DRSs, all presupposed information is presented in a separate DRS conjoined with the DRS containing the asserted content. In this particular example the presuppositions are triggered by the proper name *Mr Stronach* and the definite description *the company*, and are globally accommodated, following the algorithm proposed by van der Sandt (1992).

We see here that *wants* is a subject control verb, which is reflected in the DRS analysis because the agent of the "want" event, x_1 (denoting *Mr. Stronach*), is identical to the agent of the "resume" event. The theme of the "want" event is the context described by discourse referent c_4, which associates the DRS with a meaning paraphrased as "Mr Stronach retaining the influential role in the company". The lexical entry for *wants* with this CCG category is specified in Figure 2.

TOK	*wants*
CAT	`(S:dcl\NP)/(S:to\NP)`
SEM	⟨⟨⟨⟨e,t⟩,t⟩,⟨⟨e,t⟩,t⟩⟩,⟨⟨⟨e,t⟩,t⟩,⟨⟨e,t⟩,t⟩⟩⟩
DRS	λv.λn.λp.(n@λx.([e c | want(e) agent(e,x) theme(e,c) c:((v@λp.(p@x))@λe. [|])] ;(p@e)))

Figure 2: Lexical entry for *wants* (subject control).

Note that, even though the VP complement for this subcategorisation frame of *want* isn't of propositional type, no type-coercion is needed. The mechanism of subject-control already yields the proper type.

3.2 NP-complement

The syntactic configuration of this lexical category for *want* corresponds to an ordinary transitive verb, and it is assigned to almost a quarter of all categories for *want* in CCGbank (Table 1). Some Wall Street Journal examples are (2) and (9); Pustejovsky (1995) discusses (10) at length.

(9) The U.S. wants quick results.

(10) John wants a beer.

Obviously, applying the standard semantic analysis of transitive verbs to an intensional verb such as *want* will yield inadequate interpretations, as there is no account for the elided predicate that seems to occur in this use of *want* to produce a proper context of desire. The evidence for the existence of such an elided predicate is two-fold (Partee, 1974). First, the NP-complement case of *want*, as do the other subcategorisation frames, introduces an opaque context, in which indefinite noun phrases can receive a non-referential reading. The preferred interpretation of (10) is where John doesn't have a particular beer in mind — he is just thirsty and couldn't care less which specific beer he could get his hands on. Second, temporal modifiers seem to be able to interact with the elided predicate, as (11) shows (Partee, 1974; Pustejovsky, 1995).

(11) Bill wanted a car until next week.

One approach to account for the elided predicate is to assume a lexically-governed transformation which deletes an embedded *to have*, as has been argued for by generative semanticists (Partee, 1974). In contrast, Pustejovsky (1995) argues that the interpretation of this elided predicate is context-dependent, and that an example such as *want a beer* has a strong default interpretation that amounts to *want to drink a beer*, where *drink* is produced by the telic aspect of the qualia structure of *beer*. He proposes the machinery of complement type coercion to account for this phenomenon.

I will recast and compare these approaches in my combined framework of CCG and DRT, and propose *type-coercion rules* as a new class of non-combinatory rules in CCG. There are two main types of coercion rules: argument coercion and functor coercion. In the case of *want*, we simply discard its lexical category for transitive verbs. To derive a complete analysis for sentence such as (10) we have the following two possibilities:

1. **Argument Coercion**: transform the argument `NP` into the matching `S:to\NP`.
2. **Functor Coercion**: transform the functor `(S:dcl\NP)/(S:to\NP` into the matching `(S:dcl\NP)/NP`.

Applied to example (10), the CCG derivations shown in Figure 3 are generated, in which the coercion rules are indicated by the symbol C.

Let's have a closer look at these proposals. First consider argument coercion. Here the coercion rule changes the argument of category `NP` into `S:to\NP`, enabling us to give a spanning analysis of (10). The semantic interpretation of this coercion rule, as indeed any coercion rule, is not fixed and depends on the context. Hence, in a way, the coercion rules behave similar to the type-changing rules in CCG. Given a semantic interpretation ψ for the noun phrase, we can characterise its coercion as

```
NP            ↦   ψ
------C
S:to\NP       ↦   (φ@ψ)
```

where ϕ denotes the semantic interpretation of a transitive verb with category `(S:to\NP)/NP`. I assume that the coercion process has somehow access to the lexicon to derive the required semantic interpretation of a transitive verb. We could either pick a general transitive verb interpretation such as *have* that could apply to any argument (*weak argument coercion*), or one that is specifically triggered by the argument (*strong argument coercion*). The latter option conforms to Pustejovsky (1995)'s type coercion involving qualia structure from the noun phrase complement.

Let's now consider functor coercion. It is a process that transforms the functor, which has been assigned category `(S:dcl\NP)/(S:to\NP)`, into the category `(S:dcl\NP)/NP`. This enables us to give a spanning analysis for example (10). Given the semantic interpretation ψ for *want* in Figure 2, we can interpret functor coercion as

```
(S:dcl\NP)/(S:to\NP)    ↦   ψ
--------------------C
(S:dcl\NP)/NP           ↦   λn.(ψ@(φ@n))
```

where ϕ denotes the semantic interpretation of a transitive verb with category `(S:to\NP)/NP`. Interestingly, it seems that functor coercion can be eliminated from the grammar and stated in the lexicon, as a process of *lexical coercion*. It would be identical to functor coercion, but no coercion rules are needed, at the cost of an additional CCG category for *want* in the lexicon. This approach is demonstrated in Figure 4. Note that argument coercion cannot be replaced by lexical coercion, because arguments can have compound structures.

$$
\left[\begin{array}{ll}
\text{TOK} & \textit{wants} \\
\text{CAT} & \texttt{(S:dcl\textbackslash NP)/NP} \\
\text{SEM} & \langle\langle\langle e,t\rangle,t\rangle,\langle\langle\langle e,t\rangle,t\rangle,\langle\langle e,t\rangle,t\rangle\rangle\rangle \\
\text{DRS} & \lambda n_2.\lambda n_1.\lambda p.(n_1@\lambda x_1.(\begin{array}{|l|}\hline \text{e c} \\ \hline \text{want(e)} \\ \text{agent(e,}x_1\text{)} \\ \text{theme(e,c)} \\ \text{c:}(n_2@\lambda x_2.\begin{array}{|l|}\hline \text{e} \\ \hline \text{have(e)} \\ \text{agent(e,}x_1\text{)} \\ \text{theme(e,}x_2\text{)} \\ \hline \end{array}) \\ \hline \end{array};(p@e))))
\end{array}\right]
$$

Figure 4: Lexical entry for *wants* (transitive).

What are the differences between functor and argument coercion? They are, first of all, clearly not just notational variants: functor coercion can be lexicalised, argument coercion can't. Weak argument coercion yields the same results as functor coercion, if the coerced relation is a general one. For *want* and *need* the verb *have* seems a good candidate, but for other verbs, such as *ex-*

```
John wants                 a    beer      John wants                 a    beer
---- -------------------- ---- ----      ---- -------------------- ---- ----
N    (S:dcl\NP)/(S:to\NP) NP/N N         N    (S:dcl\NP)/(S:to\NP) NP/N N
---L                      -------->      ---L -------------------C -------->
NP                        NP             NP   (S:dcl\NP)/NP        NP
                          --------C           ---------------------------->
                          S:to\NP             S:dcl\NP
     ---------------------------->       ---------------------------------<
     S:dcl\NP                            S:dcl
---------------------------------<
S:dcl
```

Figure 3: Comparing argument coercion (left) with functor coercion (right).

pect, this isn't always appropriate (Partee, 1974). Strong argument coercion would produce a default interpretation, relying on a device such as qualia structure (Pustejovsky, 1995). It would however also require a device that can offer an alternative interpretation in examples such as (12), which certainly doesn't mean that John wants to drink his beer quickly.

(12) John wants a beer quickly.

An argument that seems to be in favour of argument coercion (both the stong and weak version), at least with the machinery proposed in this article, is its potential to deal with modifiers that interact with the coerced predicate in examples such as (11) and (13).

(13) John wants nice weather tomorrow.

Here the temporal adverb *tomorrow* is modifying (at least under the preferred interpretation) the elided predicate, whether this is *have* obtained with functor coercion or weak argument coercion, or something like *enjoy* or *experience*, obtained with strong argument coercion. Given the CCG machinery at our disposal, we can account for this reading with argument coercion, but not with functor coercion.

3.3 NP+VP complement

The three lexical category for the third complement class of *want* in Table (1) have both an NP and a VP as complement, and together cover ca. 10% of the occurrences of *want* in CCGbank. For Pustejovsky (1995), this is actually the canonical case, not involving any kind of coercion. The case with an infinitival VP, as shown in (3) and (14), is however far more common than the case with a passive verb phrase, as in (4) or (15), or adjectival phrase (5).

(14) Czechoslovakia wants the dam to be built.

(15) The country wants half the debt forgiven.

Note that in Government and Binding theory this phenomenon is referred to as "exceptional case marking", because *want* having a category typically known from control verbs such as *promise* or *persuade*, it doesn't introduce a thematic role between itself and the subject of the embedded clause, despite the fact that the embedded subject is assigned accusative case. This observation is also reflected in the partial DRS given in Figure 5, where the only roles assigned is the agent role between the subject of the main verb and the *want*-event, and the theme role between the context of desire and the *want*-event.

$$\left[\begin{array}{ll} \text{TOK} & \textit{wants} \\ \text{CAT} & ((\text{S:dcl}\backslash\text{NP})/(\text{S:to}\backslash\text{NP}))/\text{NP} \\ & ((\text{S:dcl}\backslash\text{NP})/(\text{S:pss}\backslash\text{NP}))/\text{NP} \\ & ((\text{S:dcl}\backslash\text{NP})/(\text{S:adj}\backslash\text{NP}))/\text{NP} \\ \text{SEM} & \langle\langle\langle e,t\rangle,t\rangle,\langle\langle\langle\langle e,t\rangle,t\rangle,\langle\langle e,t\rangle,t\rangle\rangle,\langle\langle\langle e,t\rangle,t\rangle,\langle\langle e,t\rangle,t\rangle\rangle\rangle\rangle \\ \text{DRS} & \lambda n_1.\lambda v.\lambda n_2.\lambda p.(n_2@\lambda x.(\begin{array}{|l|}\hline e\ c \\ \hline \text{want}(e) \\ \text{agent}(e,x) \\ \text{theme}(e,c) \\ c{:}((v@n_1)@\lambda e.\begin{array}{|c|}\hline \\ \hline \\ \hline \end{array}) \\ \hline \end{array};(p@e))) \end{array}\right]$$

Figure 5: Lexical entry for *wants* (NP+VP).

3.4 NP+PP complement

The fourth lexical category in Table (1) has both an NP and PP as complement. There is only a handful of cases found in CCGbank where *want* is assigned this category, as in example (6). As far as I am aware this case hasn't been discussed in the literature before.

The problems associated with this use of *want* is exemplified by first looking at an NP-complement case as in (16a), an example which can be paraphrased as (16b), and we can account for by functor coercion. As we have seen before, Pustejovsky (1995) goes a step further, and claims that a default interpretation of (16a) is more specific, and using the telic role present in the qualia structure of the complement and strong argument coercion, produces an interpretation expressed by (16c).

(16) a. I want bread.

b. I want to have bread.

c. I want to eat bread.

Now consider example (17a), which can be paraphrased as (17b), but in contrast, cannot be expressed as meaning (17c), eliminating the possibility of strong argument coercion. (Hence advocates of strong argument coercion need a story here to explain why the default interpretation for (17a) isn't available. A possible explanation is that the purpose of the speaker overwrites the telic role of bread.)

(17) a. I want bread on the table.

b. I want to have bread on the table.

c. ?? I want to eat bread on the table.

d. I want bread to be on the table.

Interestingly, (17a) does not entail (16a), nor does (17b) entail (16b). Also, it seems the occurrences of the reconstructed *have* in (16b) and (17b) show different senses, and the meaning of (17b) is close to that expressed in (17d). Thus, I claim that we're dealing with yet another subcategorisation frame for *want*, and one that requires a special treatment in the lexicon. Further evidence for this claim comes from examples constructed from the Wall Street Journal are (18) and (19).

(18) We want Nelson Mandela out of prison.

(19) No one wants stock on their books.

I argue that no coercion is required in these examples, but that combining the NP and PP complements results in a state that expresses the context of desire. From the viewpoint of compositional semantics, this is sound: an NP is of type $\langle\langle e,t\rangle,t\rangle$, and a PP of type $\langle e,t\rangle$, and applying the former to the latter yields an expression of type t, corresponding to a DRS, as Figure 6 shows. This analysis gives an interpretation for a sentence such as (17a) that can be paraphrased as "I desire a state in which the bread is on the table".

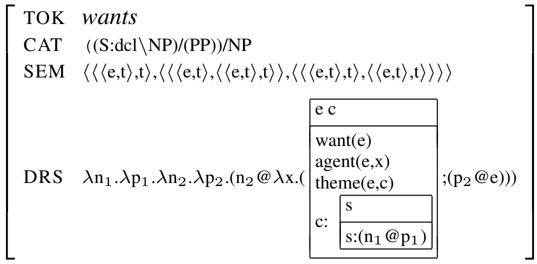

Figure 6: Lexical entry for *wants* (NP+PP).

4 Loose Ends

4.1 Coercion as Rewriting

A different way of viewing this process is to see the coercion rule as short-hand notation for a larger piece of CCG derivation, as if an elided structure has been resolved on the level of syntax. Under this view, the argument coercion can be resolved by a simple rewriting rule (Figure 7). Similarly, we can formulate functor coercion with the help of a rewriting rule (Figure 8). I assume that such rewriting rules have somehow access to the lexicon to retrieve an appropriate verb and substitute it for the slot marked by the question marks.

```
                  to                  ???          X
X                 ------------------  -----------  --
--                (S:to\NP)/(S:b\NP)  (S:b\NP)/NP  NP
NP         ⇒                          ------------->
------C                               S:b\NP
S:to\NP           -------------------------------->
                  S:to\NP
```

Figure 7: Rewriting rule for argument coercion.

```
                          X
                          ------------------
                          (S:X\NP)/(S:to\NP)
                          ------------------C
                          (S:X\NP)/NP

                               ⇓

X                   to                  ???
------------------  ------------------  -----------
(S:X\NP)/(S:to\NP)  (S:to\NP)/(S:b\NP)  (S:b\NP)/NP
                    ------------------------------>B
                    (S:to\NP)/NP
-------------------------------------------------->B
(S:X\NP)/NP
```

Figure 8: Rewriting rule for functor coercion.

4.2 Tense

The event described by the desired context introduced by *want* takes place after the event of expression the desire. Following Kamp and Reyle (1993) in representing tense, we can reformulate the partial DRS and include conditions on tense, as done for the subject-control case in Figure 9 for present tense of *want*. Here now is a constant denoting the utterance time, and the condition $e \subseteq t$ reads as "event e is included in period t". For the past tense it suffices to replace the DRS-condition $t_1 = now$ by $t_1 < now$.

TOK *wants*
CAT (S:dcl\NP)/(S:to\NP)
SEM $\langle\langle\langle\langle e,t\rangle,t\rangle,\langle\langle e,t\rangle,t\rangle\rangle,\langle\langle\langle e,t\rangle,t\rangle,\langle\langle e,t\rangle,t\rangle\rangle\rangle$
DRS $\lambda v.\lambda n.\lambda p.(n@\lambda x.($ [e c t_1 | want(e), $e \subseteq t_1$, $t_1 = now$, agent(e,x), theme(e,c), c:((v@$\lambda p.(p@x))@\lambda e.$ [t_2 | $e \subseteq t_2$, $t_1 < t_2$])] ;(p@e)))

Figure 9: Lexical entry for *wants* (tensed)

4.3 Presupposition

The verb *want* seems to trigger a complex presupposition, one that interacts with tense and negation. This presupposition can be characterised as a state in which the main event of the context of desire isn't realised. The evidence for this presupposition is backed up by the observation that (10) entails that John is in a situation where he has no beer, and (17a) entails that there is no bread on the table. These entailments seems to survive the usual tests for identifying presupposition triggers (van der Sandt, 1992). If we encode the presupposition trigger in the lexical entry for *want*, we will obtain DRSs such as (20) for a sentence like (14).

(20) ([x_1 x_2 | named(x_1,czechoslovakia), dam(x_2), ¬[s_5 t_1 | build(s_5), product(s_5,x_2), s_5 o t_1, $t_1 = now$]] ; [e_3 c_4 t_6 | want(e_3), agent(e_3,x_1), theme(e_3,c_4), $e_3 \subseteq t_6$, $t_6 = now$, c_4: [e_5 t_7 | build(e_5), product(e_5,x_2), $e_5 \subseteq t_7$, $t_6 < t_7$]])

4.4 Scope

Opaque contexts which host indefinite descriptions are known to give rise to scope ambiguities. This is also the case for contexts of desire. For instance, (7) is ambiguous with respect to the respective scopes assinged to *an influential role* and the context of desire introduced by *want*. There is a narrow scope interpretation (also known as the *de dicto* reading or *non-referential* reading), where Mr Stronach has no specific role in mind. There is also a wide scope interpretation (known as the *de re* reading or *referential* reading), where Mr Stronach has a particular role in mind. The ambiguity also appears for the NP-complement case of *want*: (10) has besides the non-referential reading also a referential reading where John has a particular beer in mind that he desires. The current framework doesn't deal with scope ambiguities, and always produces the non-referential reading. But note that the partial DRS in Figure 4 could be reformulated to get the referential reading.

References

Johan Bos. 2008. Wide-Coverage Semantic Analysis with Boxer. In J. Bos and R. Delmonte, editors, *Semantics in Text Processing. STEP 2008 Conference Proceedings*, volume 1 of *Research in Computational Semantics*, pages 277–286. College Publications.

Johan Bos. 2009. Towards a large-scale formal semantic lexicon for text processing. In M. Stede, C. Chiarcos, and R. Eckart de Castilho, editors, *Proceedings of GCSL 2009*. Gunter Narr Verlag.

James Curran, Stephen Clark, and Johan Bos. 2007. Linguistically Motivated Large-Scale NLP with C&C and Boxer. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 33–36, Prague, Czech Republic.

Julia Hockenmaier. 2003. *Data and Models for Statistical Parsing with Combinatory Categorial Grammar*. Ph.D. thesis, University of Edinburgh.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Kluwer, Dordrecht.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

Barbara H. Partee. 1974. Opacity and scope. In Munitz and Unger, editors, *Semantics and Philosophy*, pages 81–101. New York: NYU Press.

James Pustejovsky. 1995. *The Generative Lexicon*. The MIT Press, Cambridge, Massachusetts/London.

Mark Steedman. 2001. *The Syntactic Process*. The MIT Press.

R.A. van der Sandt. 1992. Presupposition Projection as Anaphora Resolution. *Journal of Semantics*, 9:333–377.