Dependencies, semantic constraints and conceptual closeness in a dynamic frame theory

Ralf Naumann

A neglected, if not almost ignored topic in formal semantic theories of natural language are semantic (or meaning) relations between lexical items. Results from psycholinguistics and neuroscience, in particular based on the N400, provide ample evidence that such relations play indeed an important and prominent role during the (semantic) processing of sentences in the brain. For example, although neither \textit{John squeezed an orange} nor \textit{John squeezed an apple} contain a semantic anomaly, they are processed differently in the brain, because \textit{orange} is more expected as the direct object of \textit{squeeze} than \textit{apple}. Similarly, \textit{This melon sounds ripe} is acceptable whereas \textit{This melon sounds oval} is not, although in both cases an adjective is used that is semantically not directly related to the sound dimension of a melon.

In this article we propose a dynamic and probabilistic extension of frame theory (Löbner 2014, Naumann 2013) in which data like the above can be analyzed. In order to capture both semantic relations and constraints (or expectancies), we use techniques from Dependence Logic (Väänänen 2007) and theories of belief revision and belief update (Spohn 1988, Goldszmidt & Pearl 1992, Boutilier 1998, Gärdenfors 1988). Using frames makes it possible to use a decompositional analysis: an object is related to a set of properties that can be changed by events. As a consequence, a lexical item like ‘orange’ can be taken as a table in a database or knowledge base consisting of attribute-value pairs. This way of interpreting lexical items makes it possible to apply the strategies from Dependence Logic and theories of belief revision and belief update mentioned above. In particular, it is possible (i) to define dependency relations between different properties of an object and (ii) to...
define quantitative plausibility relations (κ-rankings) on a frame that determine how this frame is revised or updated with new information.

1 Introduction

According to many, if not most, current formal semantic theories, common nouns like ‘orange’ or ‘paper’ are basically analyzed as sets of objects. For example, ‘orange’ is first translated as the lambda-term $\lambda x. orange(x)$, which, in a second step, is interpreted as a subset of the domain, or, more precisely, as a function from this domain to the set of truth values (1a). Similarly, using an event-based approach, verbs like ‘run’ are interpreted as sets of events or the corresponding characteristic function (1b).

(1) a. $[[orange]]^M = \lambda x \in D_{\langle e_{\text{object}}\rangle} \cdot f_{orange}(x) = 1$
   
   b. $[[run]]^M = \lambda e \in D_{\langle e_{\text{event}}\rangle} \cdot f_{run}(e) = 1$

In recent years, such approaches to defining the semantics of basic lexical items like common nouns and verbs have been criticized from neuroscience. According to Baggio & Hagoort (2011), those theories are ‘by design insensitive to differences between words of the same syntactic category denoting objects of the same type’ (Baggio & Hagoort 2011, 1343). As a consequence, they are inappropriate as a theory of semantic processing in the brain. This criticism is based on empirical results from neurophysiological and neuroimaging phenomena like the N400\(^1\), which is a component of event-related potentials (ERP’s), whose amplitude is modulated by semantic complexity.

N400.

Consider the examples in (2) and (3).

(2) a. Jenny put the sweet in her mouth after the lesson.
   
   b. Jenny put the sweet in her pocket after the lesson.

(3) Every morning John makes himself a glass of freshly squeezed juice. He keeps his refrigerator stocked with (oranges/apples/carrots).

---

\(^1\) For details on this component, see Baggio & Hagoort (2011).
A formal semantic analysis of the sentences in (2) differs only in the sort of object assigned to the locative argument of the verb put: mouth versus pocket. Yet, when this sentence is uttered in a context where Jenny leaves the classroom after a lesson, Hagoort & Brown (1994) found a difference in the N400 between mouth and pocket, showing that there is a difference during processing in the brain that needs to be accounted for by formal semantic theories.

Sentences like (3) were used by Federmeier & Kutas (1999) in an ERP experiment also targeting the N400. The authors found an increasing N400 effect with the ordering ‘oranges’ < ‘apples’ < ‘carrots’. According to one interpretation of the N400, this effect is closely related to predicting upcoming words in a sentence which is based on semantic relations between words in the memory component of the brain. For example, in (3) both ‘apple’ and ‘carrot’ trigger a larger N400 compared to ‘orange’ because the former are semantically less related to an event of squeezing a fruit than the latter (Kutas & Federmeier 2011). As an effect, the cost of semantically integrating ‘apple’ or ‘carrot’ in the given semantic context (say ‘John squeezed’) is higher than in the case of ‘orange’. Thus, the N400 is an effect that is directly related to semantic relations between concepts expressed by words in the lexicon, in particular by relations between nominal and verbal concepts.

Stimulus subject perception verbs.
Perception-based verbs (henceforth PBVs) refer to sensory properties of objects like ‘taste’ or ‘sound’. Correlated to each sense modality is a set of values that this property can take and which are specific to it. For example, for the property ‘sight’ appropriate values are ‘square’, ‘oblong’ and ‘oval’. PBVs admit of a direct-sensory use in which a predicative complement is added. Semantically, this complement specifies an intra-dimensional value, i.e. an element of the set of values appropriate for the property expressed by the verb.

(4) This melon sounds muffled/tastes sweet/smells fruity.

In addition to the direct perception use, PBVs can be used inferentially. In this case the predicative complement does not determine a value of the scale corresponding to the modality expressed by the verb, but a value belonging to another modality.

(5) a. This melon sounds ripe/old/*oval.
   b. This melon looks oval/*muffled.
The examples in (5) show that the inferential use is not always admissible but depends on the types of sense modalities expressed by the verb and the predicative complement. Thus, similar to the examples of simple sentences, semantic processing of this use of PBVs involves semantic relations. In this case, these are relations between different properties of objects that can be changed by actions or events.

**The the ... the construction.**

The third and final construction involving semantic relations discussed in this article is the *the* . . . *the*-construction.

(6) a. The older a stamp, the more expensive it is.
   b. The more alcohol you drink, the higher is your blood alcohol concentration.
   c. The more residents are affected, the sooner noise abatement measures will be implemented.

Similar to the inferential use of PBVs, this construction expresses a dependency relation between the values of two properties over time. The relation between the values of the two properties need not be strict, as shown by the following example: ‘The older a stamp, the more likely it is that it gets more expensive’.

What is common to all the examples discussed in this section are the following points: (i) there is an explicit or implicit reference to properties of objects or events. Events of squeezing are semantically more related to objects of sort ‘orange’ than to, say, objects of sort ‘carrots’; this reference is explicit in the case of the inferential use of PBVs and the *the* . . . *the*-construction. It is always implicit in the case of verbs, at least if they are formally analyzed as given in (1b). (ii) this reference to properties of objects and events is not used in isolation but rather in the context of a semantic relation between various properties. So what is at stake are semantic relations and, even more importantly, the degree to which those properties are semantically related to each other. This latter aspect will be called *semantic closeness* between properties (or between the concepts related to those properties) and (iii) since all examples involve non-stative verbs like ‘squeeze’, or analogous constructions like ‘getting older’, which denote events that bring about changes with respect to objects, one also has to consider semantic relations between properties *over time* when the values of these properties are changed by actions or events.
2 Outline of the theory

According to Baggio and Hagoort (2011:1342), formal semantic theories which describe how words belonging to different syntactic categories or denoting different sorts of objects combine to more complex units are ‘by design insensitive to differences between words of the same syntactic category denoting objects of the same type’. The authors put the blame for this ‘insensitivity’ on the fact that such theories focus on truth conditions, i.e. how language relates to the world, and not on considering natural language as a psychological phenomenon. What is required, instead, is to provide a theory of semantic processing that is both combinatoric and able to track usage-based semantic expectations of the kind involved in the data from section 1. It is important to note that this criticism not only applies to static formal semantic theories in which the notion of truth conditions is taken as central but equally to dynamic variants of formal semantics if the dynamic component is restricted to account for inter-sentential anaphoric relations, which are analyzed in terms of discourse referents or pegs. Consider (3) again. The dynamic aspect of the update operation triggered by the direct object (say ‘an orange’) is independent of the particular sort of object, but only depends on the context change potential of the existential quantifier associated with the translation of ‘a’ in the formal language since this constituent introduces a new discourse referent. By contrast, the head noun receives a completely static interpretation. It is analyzed as a test. The sortal information provided by the common noun is used to eliminate all possibilities in which the object assigned to the discourse referent fails to satisfy this condition. As a result, there is no further difference between, say, ‘an orange’ and ‘an apple’ at this level of information.

What is completely missing from this view of information encapsulated both in static and dynamic approaches to meaning in natural language is the aspect that (declarative) sentences describe situations in the world. Such a description can either concern the fact that some property of an object holds (or fails to hold) or that an event (action) occurs which changes some property of an object. One area in which this type of information is dealt with are theories of belief revision and belief update. Belief revision is usually taken as dealing with incorporating new information about a static, unchanging world. By contrast, belief update is about incorporating information about changes in the world that are triggered by actions or events. New information about a static world is incorporated into a ranked belief
set (often called an epistemic state). As a consequence, the way such an epistemic state is changed not only depends on the formulas that currently form the belief set (or knowledge base) of an agent but also on the way those formulas (or the possible worlds used to interpreted those formulas) are ranked. Such information cannot be inferred if the meaning is restricted to sortal information, say it is an orange or a running, and if the dynamics only captures discourse information.

The conclusion that we draw from this failure of current formal semantic theories is that semantic processing cannot solely be based on (i) truth-conditional content and (ii) discourse information in form of information about anaphoric relations which leads to the notion of a context change potential in terms of discourse referents or pegs and (iii) (possibly) world knowledge and context information. In addition, there are at least three further types of information: (i) information about the semantic closeness between nominal and verbal concepts, which expresses degrees of expectancy or plausibility between these two types of concepts. This type of information corresponds to ranking functions in theories of belief revision and belief update; (ii) dependency relations between the values of two properties of an object which can be expressed in Dependence Logic and (iii) information about the way such dependencies are related over time if the values of the corresponding properties are changed by events. Such information requires the use of various ranking functions that not only consider static semantic relations but also the way of how such relations can be defined in the context in which not only a static world but a world in which events bring about changes is taken into consideration (belief update).

Consider the following example. When processing a common noun like 'orange', a language user only gets sortal information: it is an object of sort 'orange' belonging to a particular subset of the universe (or the domain of the model). This kind of information is exactly what is usually captured in an (extensional, type-theoretic) truth-conditional semantics and which is formalized by the meaning or satisfaction clauses in (1). This aspect of meaning will be called the proper or lexical meaning of a common noun or an intransitive verb. Thus, as in model-theoretic semantics, the lexical meaning of common nouns and verbs is defined in terms of only sortal information and (possibly) its arity.

Another way of looking at the difference between current dynamic approaches and our approach is the following: whereas the former defines the dynamics with respect to words belonging to closed word classes like determiners ('a' or 'some') or modal expressions like 'might' (Veltman’s Update Semantics), our approach locates the dynamics in open word classes like common nouns and verbs that are used to express changes occurring in the world.
Dependencies, semantic constraints and conceptual closeness in a dynamic frame theory

Given only this information, no information about non-sortal properties is supplied. In order to get such information, a language user applies both local contextual information and global world knowledge to extend this lexical information, e.g. by information about properties of objects.\(^3\)

<table>
<thead>
<tr>
<th>object</th>
<th>color</th>
<th>form</th>
<th>origin</th>
<th>ripeness</th>
<th>taste</th>
</tr>
</thead>
<tbody>
<tr>
<td>orange</td>
<td>green</td>
<td>oval</td>
<td>spain</td>
<td>ripe</td>
<td>sweet</td>
</tr>
</tbody>
</table>

Table 1: Tabular representation of the lexical meaning of the common noun ‘orange’ enriched with contextual information and world knowledge.

From a linguistic point of view, the representation in Table 1 provides a decompositional analysis of a common noun.

\[ \lambda x (\text{orange}(x) \land \text{color}(x) = \text{green} \land \text{form}(x) = \text{oval} \land \text{origin}(x) = \text{spain} \land \text{ripeness}(x) = \text{ripe} \land \text{taste}(x) = \text{smooth} \ldots ) \]

However, such a decompositional representation of the meaning of a lexical item is still both a flat and completely static structure in the following sense. First, no distinction is made between admissible values for a particular property. Although these values can be ordered (e.g. say in form a scale, i.e. a partially or linearly ordered set), there is no relation that orders them with respect to plausibility or expectancy. Second, no distinction is made between admissible values for objects to which this object can be related. For example, for the denotation of common nouns: what are the most plausible (expected) events that bring about a change w.r.t. one of its properties? Conversely, for events denoted by verbs: with respect to which sorts of objects does the event most likely bring about a change? Third, ‘Does the event have more than one outcome, i.e. it is deterministic or non-deterministic?’ Fourth, no information about dependencies between (the values of) properties is expressed. Thus, the problem is not only related to getting more information, but also to the question of how this information is ranked and what dependencies exist between different properties. However, in order to impose both expectancy and

\[^3\] Thus, our analysis follows Hagoort’s ‘Immediacy Hypothesis’: all sorts of information available to the comprehender is immediately used in parallel in order to arrive at a meaningful interpretation. According to this hypothesis, a language user not only uses the information provided by the lexical meaning of an expression, say it is an orange or a port, but also information from the context in which an expression is processed and his world and/or background knowledge.
dependency constraints a decompositional analysis of the denotations of common nouns and verbs is needed because only then is it possible to explicitly refer to the properties with respect to which those constraints are defined.

Another way of looking at this problem is in terms of the information state of a language user. We follow dynamic approaches and define an information state as a set of possibilities consisting of the alternatives that are still open according to the information available to the language user. In this paper information sets are defined in terms of variable assignments. Following standard practice, sets of assignments will be called possible worlds. Consider (7) again. The information state of a language user w.r.t. to this information is given by a set of possible worlds capturing his epistemic uncertainty, which is due to the fact that his knowledge about the values of properties of an orange is only partial and incomplete. As an effect, his knowledge consists of all those possibilities that are compatible with his current knowledge. In the present case the alternatives concern possible expansions of his knowledge about the orange. He then assumes that the actual (correct) description is some subset $U$ of the set $W$ of possible worlds. However, since all possible worlds are assumed to have equal status for the language user, no world is preferred or more expected than any other in the set of all possibilities. As a consequence, updating amounts to intersecting. A further problem concerns the information that a language user can infer from his current information state provided, say, by applying the information given by the lexical meaning plus context information together with world knowledge. If his information state is a flat structure in the sense that all worlds are taken as equal, no information about the values of properties about which no information is provided can (defeasibly) be drawn. By contrast, if a language user has information both about dependency and expectancy relations, he can use this type of information to (defeasibly or non-defeasibly) infer other pieces of information about the situation described by the sentences he is currently processing. For example, knowing only that a melon is ripe, he defeasibly infers a value for its Sound attribute: muffled. Thus, the cognitive significance of dependency and expectancy relations consists in the fact that given part of a sentence, a language user will defeasibly infer as much additional information about the situation described by the sentence as possible. Consider the following example. Suppose there is an input state representing mostly ducks (say, because the topic of a conversation are ducks). Then an event of

4 Or, as sets of sets of assignments. See section 3 for details.
swimming is more expected than events of jumping or walking. By contrast, if the topic is about deers, swimming is less expected than jumping.\footnote{See van Elk et al. (2010) for empirical evidence based on an EEG study and references cited therein.}

3 Outline of the formalization

3.1 Structures for events, objects and their properties

We start by fixing models for objects and events that capture sortal information which is used in defining the lexical meanings of common nouns and verbs.

**Definition 1 (Object structure)** Let $CN$ be a set of object sort symbols like ‘orange’. An object structure $O$ is a quadruple $\langle O, \{P_{cn}\}_{cn \in CN}, \sqsubseteq_o, \sqcup_o \rangle$ s.t. (i) $O$ is a non-empty set of objects like trees and dogs; (ii) each $P_{cn}$ is a subset of $O$; (iii) $\sqsubseteq_o$ is the material part-of relation on $O$, which is required to be a partial order and (iv) $\sqcup_o$ is the join operation on $O$, which is required to be a join-semilattice.

**Definition 2 (Event structure)** Let $VERB$ be a set of event sort symbols like e.g. ‘squeeze’. An event structure $E$ is a quadruple $\langle E, \{P_v\}_{v \in VERB}, \sqsubseteq_e, \sqcup_e \rangle$ s.t. (i) $E$ is a non-empty set of actions and/or events like runnings or readings; (ii) each $P_v$ is a subset of $E$; (iii) $\sqsubseteq_e$ is the material part-of relation on $E$, which is required to be a partial order and (iv) $\sqcup_e$ is the join operation on $E$, which is required to be a join-semilattice.

Elements of $E$ and $O$ will be called entities. At the level of $O$ and $E$, entities are taken as elements of the underlying domain of some fixed global model $M$, which can have parts. Examples are the leg of a table or the tail of a dog for the domain $O$ of persistent objects and a subevent of eating half an apple for the eating of the whole apple in the domain $E$. Such relations are represented by a part-of relations $\sqsubseteq_o$ and $\sqsubseteq_e$, respectively. In addition, they can be ‘summed’ to form plural entities. For example, if $o$, say Fred, and $o'$, say Mary, are elements of the object domain $O$, then $o \sqcup_o o'$ is also an element of $O$. This is modeled by the join operations $\sqcup_o$ and $\sqcup_e$, respectively.

What is missing at this level is the view of an entity as a ‘bundle’ of properties, corresponding to a decompositional analysis at the linguistic and/or conceptual level. Such a view makes it possible to impose constraints on (the values of) properties of entities denoted by common nouns and events. Properties of objects
like ‘Ripeness’, ‘Sound’ or ‘Age’ are represented by partially or linearly ordered sets, called scale structures.

**Definition 3 (Scale structure)** A scale structure \( D \) is a pair \( \langle \Delta, \leq \rangle \) s.t. \( \Delta \) is a non-empty set of degrees, the set of admissible values for the scale, and \( \leq \) is an ordering on \( \Delta \), usually either a partial or a linear order. Scales are required to have a least element. Intuitively, the least element means that no information about the value is known or provided.

Let \( PROP \) be a set of property symbols like ‘sort’ or ‘ripeness’ and let \( \{ D_p \}_{p \in PROP} \) be a family of scale structures indexed by elements from \( PROP \). Elements of \( O \) are assigned a subset of \( \{ D_p \}_{p \in PROP} \) by a (subset of a) family of partial functions \( \{ \gamma_p \}_{p \in PROP} \), which assign to an \( o \in O \) the scale structure \( D_p \), if defined. The following condition is imposed on this assignment. If \( o, o' \in P_{cn} \), then \( \gamma_p(o) \) is defined iff \( \gamma_p(o') \) is defined and one has \( \gamma_p(o) = \gamma_p(o') \), i.e. objects belonging to the same sort are assigned the same scale structures. If \( \gamma_p(o) \) is defined for an object of sort \( cn \), the property \( p \) is said to be admissible for objects of sort \( cn \).

While processing a common noun, context information and world knowledge provide a language user with the current values of some of the properties assigned to the object denoted by the common noun. This decomposition can be represented as a (finite) conjunction of the form (8).\(^6\)

\[
\phi_\sigma \land \phi_1 \land \ldots \land \phi_n \quad (= \phi)
\]

In (8), \( \phi_\sigma \) expresses sortal information (lexical meaning), i.e. information about the property ‘Sort’ and the \( \phi_i \) non-sortal information (context information and world knowledge), e.g. information about properties like ‘Ripeness’. Since in general a language user doesn’t know the values of all properties of the object, he is epistemically uncertain about the exact ‘status’ of the object. For example, suppose that w.r.t. a particular melon the values of the properties ‘Form’ and ‘Origin’ are known to be ‘oblong’ and ‘spain’ by a language user and that there are exactly two other properties ‘Sound’ and ‘Ripeness’, whose possible values are ‘dull’ or ‘muffled’ and ‘not ripe’ or ‘ripe’, respectively. The set of possibilities can be represented by the following set of assignments. The ‘real’ melon could be any of the four melons, each corresponding to a variable assignment.

\(^6\) Alternatively, the conjunction \( \phi \) can be taken as a set of formulas, i.e. as a knowledge base in theories of belief revision.
3.2 Dependence logic

One way of looking at Table 2 is as a table in a database. In Dependence Logic (Väänänen 2007), such tables are an instance of a team. A team is a set of agents, with an agent being defined as a function from finite sets (or tuples) of variables, called the domain of the agent, into an arbitrary set, called the codomain of the agent. In the present context, agents are objects, i.e. elements of the domain $O$, viewed as bundles of properties.

**Definition 4 (Team Dependence Logic)** Let $\langle x_1, \ldots, x_n \rangle$ be a finite tuple of property variables such that no two variables are of the same property sort (i.e. each variable has associated with it a sort $p \in \text{PROP}$). Let $M$ be the union of the domains $\Delta$ from $\{D_p\}_{p \in \text{PROP}}$. An agent is any function from $\langle x_1, \ldots, x_n \rangle$ to $M$. A team $S$ is a set of agents. A team $S$ is admissible for objects of sort $cn$ if $\text{dom}(S) = \langle x_1, \ldots, x_n \rangle$ and for $x_i$, $1 \leq i \leq n$ the sort of $x_1 v(i)$ is admissible for objects of sort $cn$.

Each row in Table 2 is an assignment, or, when viewed from the point of view of an application, a possible description of an object (an agent). Properties of objects (agents) are represented by attributes which are variables in the formal representation. Thus, teams are directly related to the view of an object as a ‘bundle’ of properties.

An operation on teams is the *supplement* operation, which adds a new attribute to the objects in a team, or alternatively changes the value of an existing attribute.

**Definition 5 (Supplement of a team; Väänänen 2007)** If $M$ is a set, $S$ is a team with $M$ as its codomain and $F : S \rightarrow M$, $S(F/x_n)$ is the supplement team $\{s(F(s)/x_n) : s \in S\}$, where $s(a/x_n)$ is the assignment which agrees with $s$ everywhere except that it maps $x_n$ to $a$: $\text{dom}(s/x_n) = \text{dom}(s) \cup \{x_n\}$, $s(a/x_n)(x_i) = s(x_i)$ when $x_i \in \text{dom}(s) \setminus \{x_n\}$ and $s(a/x_n)(x_n) = a$. 

<table>
<thead>
<tr>
<th>object</th>
<th>sort</th>
<th>form</th>
<th>origin</th>
<th>sound</th>
<th>ripeness</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_1$</td>
<td>melon</td>
<td>oblong</td>
<td>Spain</td>
<td>dull</td>
<td>ripe</td>
</tr>
<tr>
<td>$m_2$</td>
<td>melon</td>
<td>oblong</td>
<td>Spain</td>
<td>dull</td>
<td>not ripe</td>
</tr>
<tr>
<td>$m_3$</td>
<td>melon</td>
<td>oblong</td>
<td>Spain</td>
<td>muffled</td>
<td>ripe</td>
</tr>
<tr>
<td>$m_4$</td>
<td>melon</td>
<td>oblong</td>
<td>Spain</td>
<td>muffled</td>
<td>not ripe</td>
</tr>
</tbody>
</table>

Table 2: A set of possibilities for an object denoted by the common noun ‘melon’
The supplement operation is used to model the combination of the lexical meaning of a common noun with context information and world knowledge about the referent of this noun in a given context. Let $x_n, \ldots, x_m$, $n < m$, be the attributes about which the context and world knowledge provide information.

If $S$ is the team corresponding to the lexical meaning of a common noun, then $S(F/x_n)(F/x_{n+1}) \ldots (F/x_m)$ is the team resulting from adding the information about the attributes $x_n, \ldots, x_m$.

In Dependence Logic, formulas are interpreted with respect to sets of assignments (teams) and not w.r.t. to single assignments as in first-order logic. In Dynamic Dependence Logic, formulas are interpreted as relations between sets of assignments (Galliani 2014). This shift makes it possible to define dependency relations between attributes. For example, functional dependency between a sequence $\vec{x}$ of variables and a variable $y$ is expressed by the atomic formula $= (\vec{x}, y)$, with the intuitive meaning 'the $\vec{x}$ totally determine $y$'. The satisfaction clause for this dependence atom is (9a). The constancy atom $= (x)$ requires the value of the attribute $x$ to be constant in a team, (9b). This formula is used to express that a language user knows the value of an attribute.\footnote{For formulas that do not contain a dependence atom, one has: $\mathcal{M} \models_S \phi$ iff for all $s \in S : \mathcal{M} \models_s \phi$, where $\models_s$ is the usual Tarskian satisfaction relation.}

(9) a. $\mathcal{M} \models_S = (\vec{x}, y)$ iff $\forall s, s' \in S(s(\vec{x}) = s'(\vec{x}) \rightarrow s(y) = s'(y))$

b. $\mathcal{M} \models_S = (y)$ iff $\forall s, s' \in S(s(y) = s'(y))$

c. $\mathcal{M} \models_X \exists x \phi$ iff there is a function $F : X \rightarrow \exists_M$ such that $\mathcal{M} \models_X[F/x] \phi$, where $\exists_M$ is the local existential quantifier defined by $\{A \subseteq M \mid A \neq \emptyset\}$ and $S[F/x]$ is the team $\{s[a/x] \mid s \in S, a \in F(s)\}$.

The interpretation of the existential quantifier is based on the supplement operation, i.e. it either adds a new attribute to all agents in the current team, or alternatively it changes the value of an existing attribute. Thus, the existential quantifier is inherently dynamic in the sense that it changes the current team w.r.t. which it is interpreted (see Galliani 2014, for details on a dynamic interpretation of Dependence Logic).

Using the dependence formula $= (\vec{x}, y)$, it is possible to express dependencies between properties like ‘Age’ and ‘Price’ for stamps and ‘Ripeness’ and ‘Sound’ for melons.
(10) a. \(=(age, price)\)
   b. \(=(ripeness, sound)\)

For example, (10b) says that the value of the attribute Sound is functionally dependent on that of the attribute Ripeness. Both examples in (10) are not quite correct because they do not take into consideration that for example (10a) holds for stamps but not for other artefacts or human beings. Second, the value of the price depends in general not only on its age but also on other factors like availability or demand. These shortcomings can be remedied as follows.

(11) a. \(x_{sort} = stamp \rightarrow = (age, price)\)
   b. \(x_{sort} = stamp \rightarrow = (age, availability, demand, \ldots, price)\)

A team represents the set of possibilities of a language user in the following sense: \(g \in S\) if and only if the language user believes \(g\) to be a possible (and complete) description of the object. As noted in Galliani & Väänänen (2014), moving from assignments to teams (or sets of assignments), makes it possible to assign to each formula \(\phi\) and model \(M\) the family of teams \(S = \{ S | M \models_S \phi \}\). As a consequence, formulas can be interpreted as conditions over belief sets. Knowledge of the value of a property in the sense that this property is assigned the same value in all information states can be expressed by a constancy atom \(=(x)\). In Table 2 above, this holds for the attributes ‘sort’, ‘form’ and ‘origin’.

**Definition 6 (Information state w.r.t. to an object)** Given a decompositional formula \(\phi\) representing the beliefs of a language user about an object \(o \in O\), his epistemic uncertainty (or his set of possibilities) w.r.t. to \(o\) is given by the family of teams \(S\) of teams satisfying \(\phi\), i.e. \(S = \{ S | M \models_S \phi \}\).

Note that information states are defined w.r.t. the domain \(O\) of objects. The domain \(E\) of events plays no role. Rather, this domain functions as a state transformer: elements of this domain trigger changes in information states.

### 3.3 Ranking functions

So far, the information state about an object of a language user is flat in the sense that all teams in this information state are taken as equally plausible. However, a language user also has expectancies about (i) the values of properties about which he so far doesn’t have any information and (ii) sorts of events in which an object of
the given sort is most plausibly involved. These expectations lead to a ranking of the teams in his information state. Such expectancies are defined in terms of $\kappa$-rankings, which are based on the notion of surprise.

**Definition 7 ($\kappa$-ranking function; Goldszmidt & Pearl 1992, Spohn 1988)**

A ranking is a function $\kappa : \Omega \to \mathbb{N}^*$ with $\Omega$ a non-empty set such that $\kappa(\omega) = 0$ for at least one $\omega \in \Omega$ and $\mathbb{N}^* = \mathbb{N} \cup \{\infty\}$.

In the present context, $\Omega$ is either a set of teams (or possible worlds with each world representing a team in $\Omega$) or the domain $E$. The numbers can be thought of as denoting degrees of surprise (Halpern 2005, p.43). For example, $\kappa(w)$ is the degree of surprise a language user attributes to team $w$ to be the ‘correct’ team representing the interpretation of a sentence or a part of it. The higher the number, the greater the degree of surprise. For example, '0' means ‘completely unsurprising’ whereas greater numbers express increasingly higher degrees of surprise. The value $\infty$ means ‘impossible’ or ‘so surprising as to be impossible’. In terms of plausibility or expectancy, the value 0 means ‘most plausible’ or ‘most expected’. Though degrees of surprise are assigned to elements of $\Omega$, they can also be defined to formulas $\phi$. The rank or degree of surprise of $\phi$ is the least rank of the set of worlds in which $\phi$ is true, (12).

\[
(12) \quad \kappa(\phi) = \min_w \{\kappa(w) \mid M \models_w \phi\}
\]

One has $\kappa(\phi) < \kappa(\psi)$ if $\phi$ is less surprising than $\psi$. For example, given an information state about a melon in which it is known that this melon is ripe it is less surprising to find that its taste is sweet than to find that the taste is not sweet but sour. One therefore has $\kappa(\text{sweet}) < \kappa(\text{sour})$.

$\kappa$-rankings are not only used for ranking information states and formulas but also to define defeasible conditionals that allow to defeasibly infer information $\psi$ from information $\phi$. These conditionals have the form $\phi \xrightarrow{\delta} \psi$ and mean ‘Typically, if $\phi$ then expect $\psi$ with strength $\delta$’ (Goldszmidt & Pearl 1992). If $\phi \xrightarrow{\delta} \psi$ holds w.r.t. a ranking $\kappa$, $\psi$ must be true in all most expected (or least surprising) worlds in which $\phi$ is true. This condition imposed on $\xrightarrow{}$ can be expressed in terms of conditional ranks, which have the form $\psi \| \phi$. Intuitively, $\psi \| \phi$ expresses the degree of surprise of finding $\psi$ given that $\phi$ is known to be true. The definition of $\psi \| \phi$ is given in (13).
Dependencies, semantic constraints and conceptual closeness in a dynamic frame theory

(13) \( \kappa(\psi|\phi) = \kappa(\psi \land \phi) - \kappa(\phi) \)

The inequality \( \kappa(\neg \psi|\phi) > \delta \) means that given \( \phi \) it would be surprising by at least \( \delta + 1 \) ranks to find \( \neg \psi \) and is equivalent to \( \kappa(\psi \land \phi) + \delta < \kappa(\neg \psi \land \phi) \) (Goldszmidt & Pearl 1992). Now \( \phi \xrightarrow{\delta} \psi \) is defined by (14).

(14) \( \phi \xrightarrow{\delta} \psi \text{ iff } \kappa(\psi \land \phi) + \delta < \kappa(\neg \psi \land \phi) \).

A ranking function \( \kappa \) is said to be admissible with respect to a set \( \Delta \) of defeasible conditionals if (15) holds.

(15) \( \kappa(\phi_i \land \psi_i) + \delta_i < \kappa(\phi_i \land \neg \psi_i) \) for all \( \phi \xrightarrow{\delta} \psi \in \Delta. \)

Each type of (defeasible) inference discussed in the introduction is analyzed in terms of a mapping whose range is a ranking function of a particular kind. These mappings differ w.r.t. (i) the sort of the domain, which reflects what is known by the language user, and (ii) the domain of the ranking function, which reflects the type or sort of knowledge that is defeasibly inferred from this knowledge. The following cases have to be distinguished.

1. **given:** information about the sort of an object;
   **inferred:** information about the action (event) in which this object is involved.
   Example: given: `duck`; inferred: `swim`.

2. **given:** information about the sort of an action (event);
   **inferred:** information about the sort of participants (modulo a thematic role)
   Example: given: `swim`; inferred: `duck`, `dolphin` or `fish`.

3. **given:** information about the value of an attribute \( a \);
   **inferred:** information about the value of a different attribute \( a' \).
   Example `The melon is ripe`: given: ‘Ripeness:ripe’; inferred: ‘Sound:muffled’.

4. **given:** the sort of an action (event);
   **inferred:** the values of attributes of objects involved in the event that hold in the consequent state of the event.
   A first example is ‘The melon ripened’. There are at least two inferences that can be drawn about the consequent state of the ripening: (i) it is ripe, i.e. the value of the Ripeness attribute is ‘ripe’ and (ii) the value of the Sound attribute is ‘muffled’. Inference (i) is non-defeasible because ‘ripen’ is a degree achievement that requires the maximum value on the underlying scale to
hold in the end state of the event. By contrast, inference (ii) is defeasible. A second, more complex, example is given by PBV’s: ‘The melon sounds ripe’. This sentence is based on experiencing a muffled sound of the melon (‘The melon sounds muffled’). Next, a most plausible sort of event is looked for such that Sound:muffled holds in its consequent state. In this case an event of ripening is the most expected candidate. Since in the consequent state of such an event Ripeness:ripe holds, ‘The melon sounds ripe’ follows.

5. given: a change in the value of an attribute \( a \);
   inferred: a change in another attribute \( a' \).
   An example is given by ‘The older a stamp, the more expensive it is’.

3.4 Rankings on information states

In a first step, the set of teams \( W \) satisfying (part of) a decompositional formula \( \phi \) is ranked.

**Definition 8 (Ranking on information states)** A ranking on an information state corresponding to a decompositional formula \( \phi = \phi_1 \ldots \phi_n \) is a ranking function \( \kappa : W \rightarrow \mathbb{N}^* \) s.t. \( \kappa^{-1}(0) \subseteq [\phi_i] \) iff \( M |\_w = (\phi_i) \) for all \( w \in W \).

The condition \( \kappa^{-1}(0) \subseteq [\phi_i] \) iff \( M |\_w = (\phi_i) \) expresses the requirement that a language user knows the value of a property if it is constant in all teams belonging to the information state.\(^8\) Note that it is not required that the whole decompositional formula \( \phi \) be known. For example, if a language user only knows that the object is of sort \( P_{cn} \), only \( \phi_\sigma \) satisfies the condition \( \kappa^{-1}(0) \subseteq [\phi_\sigma] \).

The ranking function \( \kappa \) can naturally be interpreted as characterizing the degree to which a language user is willing (i) to predict possible continuations of a sentences with respect to properties of objects and (ii) to accept alternative descriptions which are not in accordance with his current information about the object. For example, in the case of a melon or an orange, the most plausible or least surprising values for the attribute ‘Taste’ is ‘fruity’, whereas ‘salty’ will most likely get the value \( \infty \) because it is deemed to be impossible.

\(^8\) It is assumed that an attribute (or a property) has exactly one value, i.e. attributes are functional relations.
3.5 Rankings of information states on events

The first mapping to be defined captures case (1). Since this case concerns the expectancy of particular sorts of events given the sort of an object, it has to be a mapping from the domain $W$ of teams in an information state to a ranking function $\kappa$ with domain $E$.

**Definition 9 (Event ordering; Boutilier 1998)** An event ordering is a mapping $\mu : W \rightarrow (E \rightarrow \mathbb{N}^*)$ that maps each $w \in W$ to a $\kappa$-ranking $E \rightarrow \mathbb{N}^*$ on the domain of events $E$. Instead of $\mu(w)$, we will write $\kappa_w$. It is required that $\kappa_w(e) = 0$ for some event $e \in E$, i.e., there is at least one most plausible event to occur in a world $w$. If $\kappa_w(e) = \infty$, this means that an occurrence of $e$ at $w$ is taken to be impossible. In addition, we require $\kappa_w(e) = \kappa_w(e')$ for two events $e, e'$ belonging to the same sort $P_v$, i.e., events of the same sort are assigned identical plausibility for a given $w$.

Given $\mu$, each world $w$ has associated with it an event ordering $\mu(w)$ that determines the plausibility of event occurrences at that world. For example, if $W$ is a family of teams of sort ‘duck’, events of sort ‘swim’ will be assigned the value 0. By contrast, if the sort is ‘deer’, events of sort ‘jump’ are most plausible and hence get value 0. For human beings, the set of most plausible events is in general rather large due to the fact that they can be correlated to a large number of different sorts of events (see van Elk et al. 2010, for details).

Since $W$ represents information about objects, the mapping $\mu$ establishes a relation between the domain $O$ and the domain $E$. The cognitive significance of this mapping is the following. Given an information state $w$, a language user uses $\kappa_w$ to defeasibly infer the most plausible events that are likely to occur with an information state of this sort and, in an additional step, expects particular verbs (or verbs stems) to occur farther down the sentence which denote events of those sorts.

3.6 Rankings of information states w.r.t. events

The mapping $\mu$ only captures the expectancy of the occurrence of an event given objects of a particular sort. Next we define an analogous mapping that determines the expectancy of a particular sort of object, given information about an event of some sort. It maps elements of the domain $E$ to ranking functions with a domain of teams. This corresponds to case (2).

---

9 Intuitively, $\kappa_w(e)$ captures the plausibility of the occurrence of event $e$ at $w$. 
**Definition 10 (Information state ranking for events)** An information state ranking for events is a mapping \( \mu^* : E \rightarrow (W \rightarrow \mathbb{N}^*) \) that is defined by \( \mu^*(e)(w) = \mu(w)(e) \).

Intuitively, \( \mu^* \) captures the fact of what types of information states are ‘preferred’ by events of a given sort. The cognitive significance of this mapping is similar to that of \( \mu \). If a verb is encountered denoting events of type \( \sigma \), a language user uses this mapping to predict the most plausible sorts of objects to fill in a role in the event. In contrast to the mapping \( \mu \), there is not a single mapping but rather a family of such mappings because this type of mapping must be defined relative to a particular thematic role like actor or patient.

### 3.7 Event outcome ranking

In a final step, we define the relation between an event and its possible outcomes. This mapping maps an event and a team (the input state) to a ranking function on teams such that an element of this domain functions as the consequent state which results when the event occurs in the input state.

**Definition 11 (Event outcome ranking)** An event outcome ranking is a mapping \( \tau : E \rightarrow (W \rightarrow (W \rightarrow \mathbb{N}^*)) \) that assigns to an event \( e \in E \) and an (input) information state \( w \) a ranking function on the set of information states. It is required that \( \forall e, e' \in P_v : \tau(e)(w) = \tau(e')(w) \) hold, i.e. events of the same sort have the same outcome ranking functions relative to a given world \( w \). Since \( \tau(e)(w) \) is a ranking function, one must have \( \tau(e)(w)(w) = 0 \) for at least one event \( w' \) so that one outcome of \( e \) is most plausible.

Intuitively, \( \tau(e)(w)(w') \) describes the plausibility that the world \( w' \) results when event \( e \) occurs in \( w \) (Boutilier 1998:292). For example, an event denoted by ‘ripen’ results in a state in which the object that undergoes the change, say a melon, is ripe.\(^{10}\) The cognitive significance of \( \tau \) is the following. If a language user knows the sort of the event, say after having processed the predicate, he can defeasibly infer possible outcomes. Thus the set of possible outcomes (teams) is not a flat set but a ranked set of alternatives. Let \( S_{e,w} = \{ v \mid \tau(e)(w)(v) \neq \infty \} \) be the set of outcomes that possibly result for a given \( e \) and \( w \). The mapping \( \tau \) then induces a ranking \( \kappa \) on this information state as follows: \( \kappa(v) < \kappa(v') \) iff \( \tau(e)(w)(v) < \tau(e)(w)(v') \).

---

\(^{10}\) ‘The melon ripened’ implies that the melon was ripe at the end of the event since ‘ripen’ is a so-called degree achievement.
This mapping is used for case (iv). For example, if the sentence "The melon ripened" is processed, a language user not only knows that an event of sort ‘ripen’ occurred but in addition, by using $\tau$, he infers that (i) the melon is ripe and (ii) that it sounds muffled. Again, (i) is a non-defeasible inference whereas (ii) is defeasible.

The mapping $\tau$ is extended to sequences (or histories) of events in the following way.

\[(16) \ w \xrightarrow{e^n} v \text{ iff there are } u_0, \ldots, u_n \text{ s.t. } w = u_0 \text{ and } v = u_n, \ e_1 \ldots e_n \text{ s.t. } e^n = e_1 \sqcup \ldots \sqcup e_n \text{ and for each } (u_i, u_{i+1}) \text{ with } 0 \leq i < n \text{ one has } u_i \xrightarrow{e_{i+1}} u_{i+1}.\]

For $w \xrightarrow{e^n} v$, the rank is defined as the sum of the ranks of its component (atomic) transitions.

\[(17) \ \tau^*(e)(w)(v) = \sum \tau(e_{i+1})(u_i)(u_{i+1})\]

This rank expresses the degree to which a language user thinks that this history might occur (or has been occurred, using an abductive argument). It is used for case (5). Each history represents a possible evolution of how an outcome $\phi$ can be brought about by a sequence of events $e = e_1 \ldots e_n = e^n$. For example, if the sort of the events $e_i$ is restricted to events of sort ‘ageing’, all histories have an outcome in which the object undergoing the change is older than in the input state. The output states can differ w.r.t. other properties, like ‘Price’ for example, that can also be changed by an event of sort ‘ageing’.

4 Applying the formalism to the data from section 1

When processing a sentence, a language users knows that his current information state will be changed to a new one. Using his world knowledge, he also knows that this sentence either describes a change in the world or the persistence of a property of an object. In the former case the event described can either be deterministic or non-deterministic and the sentence can describe a relation between two properties over time that are linked by a dependency relation.

The cognitive significance of ranking functions and dependency relations is grounded in the fact that they allow a language user to anticipate as much information as possible about the potential output information state that results from processing the next upcoming sentence. Using the mappings $\kappa$, $\mu$, $\mu^*$ and $\tau$, he
can already calculate the plausibility of a transition \( w \xrightarrow{e} v \) as follows (Boutilier 1998:292).\(^{11}\)

\[
\kappa(w \xrightarrow{e} v) = \tau(e)(w)(v) + \mu(w)(e) + \kappa(w).
\]

According to (18), the plausibility of a transition \( w \xrightarrow{e} v \) depends on the plausibility of \( w \), the degree to which an event \( e \) is expected to occur in \( w \) and the degree to which event \( e \) can bring about an outcome \( v \) given input \( w \).\(^{12}\) Given a condition \( \phi \) that has to hold in the output state \( v \), the set of possible \( \phi \)-transitions is defined by (19) (Boutilier 1998:293).

\[
(19) \quad Tr(\phi) = \{ w \xrightarrow{e} v \mid v \models \phi \land \kappa(w \xrightarrow{e} v) \neq \infty \}.
\]

The most plausible transitions resulting in an outcome state satisfying \( \phi \) are (20).

\[
(20) \quad \text{mpt}(\phi) = \{ v \mid w \xrightarrow{e} v \in \min(Tr(\phi)) \}.
\]

In our application to natural language, the interpretation of a sentence need not involve all three mappings. Consider ‘The melon ripened’ and ‘The duck swam’. After processing the subject a language user is given a ranked set of teams, i.e. an information state, of a particular sort: ‘melon’ and ‘duck’, respectively. He uses the mappings \( \tau \) and \( \mu^* \) to predict a most expected action and/or event together with most expected results in which a melon or a duck are involved. Thus, in this particular case he will calculate \( \kappa_w(w \xrightarrow{e} v) = \tau(e)(w)(v) + \mu(w)(e) \) for a given \( w \) or, equivalently, \( \kappa(w \xrightarrow{e} v) = \tau(e)(w)(v) + \mu(w)(e) + \kappa(w) \) with \( \kappa(w) = 0 \). For ‘melon’, an event of ripening has a particular non-defeasible outcome: the melon is ripe. Given this most expected event, there is in addition a most expected (least surprising) outcome: it will sound muffled. As a consequence, the most expected (least surprising) elements of the set \( S(w) = \{ (e, v) \mid w \xrightarrow{e} v \} \) will be events of sort ripening with a consequent state \( v \) in which the melon is both ripe and sounds muffled. By contrast, for ‘duck’ and ‘swim’, the corresponding set is larger because a swimming event usually has no single, most expected outcome. For example, there can be a change of location or a loss of energy as well as a combination of such results. However, none of those changes need be salient in a

\(^{11}\) As noted by Boutilier (1998:292), this formula is the qualitative analogue of the probabilistic equation \( \Pr(w \xrightarrow{e} v) = \Pr(v|w,e) \cdot \Pr(e|w) \cdot \Pr(w) \).

\(^{12}\) For expectations that involve the passing of time like for instance in ‘The older a stamp, the more expensive it is’, the mapping \( \tau^* \) instead of the mapping \( \tau \) is used.
given context. One way of modelling this lack of salience of particular results is to assume that nothing changes with respect to properties in the input state \( w \) by setting \( v = w \). Below we will see how this idea can be made precise by using defeasible conditionals.

A drawback of using ranking functions directly is that they involve teams. However, in many cases an expectation only involves two particular properties and not all properties denoted by a complete decompositional formula. For this reason, defeasible inferences based on expectations are better directly expressed in terms of those properties. One way of doing this which is still based on ranking functions, is to formulate defeasible inferences in terms of defeasible conditionals. We follow Goldszmidt & Pearl (1992), who define a consequence relation on a set \( \Delta \) of defeasible conditionals and a distinguished \( \kappa \)-ranking \( \kappa^+ \). This ranking is defined as a ranking function that is minimal in the sense that any other admissible ranking function must be assigned a higher ranking to at least one world and a lower ranking to none.\(^{13}\) As a consequence, \( \kappa^+ \) assigns to each world the lowest possible rank permitted by the admissible constraint. The exact definition is given below. The parameter \( \delta \) stands for the strength with which the consequent follows from the antecedent. This takes care of the fact that the inference is defeasible and, at least in general, not strict. The greater \( \delta \), the greater the strength with which \( \sigma \) follows from \( \phi \). In the limiting case if \( \delta = \infty \), the defeasible inference is strict. In the sequel, when giving examples of defeasible conditionals, the exact value of \( \delta \) will be left open since this value has to be determined empirically.

**Definition 12 (Plausible inference; Goldszmidt & Pearl 1992)** \( \sigma \) is a plausible conclusion of \( \phi \) relative to a set \( \Delta \) of defeasible conditionals, written \( \phi \overset{\delta}{\longrightarrow} \sigma \), iff

\[
\kappa^+ (\phi \land \sigma) + \delta < \kappa^+ (\phi \land \neg \sigma).
\]

The antecedent contains information about the input information state which expresses information that is known to the language user, i.e. it is required that one has \( M \models_S = (\phi) \) for each team \( S \) in the input information state. The difference between the two types of information consists in their strength. Whereas the antecedent has to be known, for the information in the consequent this need not be the case.\(^{14}\) Thus, the consequent contains information that can defeasibly be added to this input information state, resulting in a new (output) information state. This information state not only consists of the information that results

\(^{13}\) Goldszmidt & Pearl (1992) show that any consistent set \( \Delta \) has a minimal ranking.

\(^{14}\) How this difference is modelled in the output information state must be left to another occasion.
when semantically processing the (surface) constituents of a sentence but also by adding the information in the consequent of defeasible conditionals whose antecedent matches information that is given by processing a particular constituent or, more generally, by information that is given by the context. The addition of the information in the consequent is modeled by using the supplement operation from Dynamic Logic, expressed by the existential quantifier. In the context of a dynamic semantics defeasible conditionals are used to construct the output information state. For example, if the consequent has the form \( \phi_1 \land \ldots \land \phi_n \) with \( \phi_i \equiv x_\sigma \), For 'The melon ripened', one gets the defeasible conditional (21a), and for 'The duck swam' the defeasible conditional (21b).

\[ \begin{align*} 
(21) \quad a. \quad x_{\text{object}} &= \text{melon} \models^\delta x_{\text{event}} = \text{ripen} \land x_{\text{ripeness}} = \text{ripe} \land x_{\text{sound}} = \text{muffled} \\
& \quad b. \quad x_{\text{object}} &= \text{duck} \models^\delta x_{\text{event}} = \text{swim} 
\end{align*} \]

The difference between an event of ripening, which brings about particular results and which therefore has a consequent state, and an event of swimming is captured by the fact that for the former but not for the latter there are defeasible conditionals specifying inferences about what holds in the consequent state.

**Simple sentences and the N400.**

For simple sentences like 'John squeezed an orange', only the mappings \( \mu \) and \( \mu^* \) are important. Outcomes play no role because only the expectancy relations between sorts are involved. In (22), two examples of plausible inferences are given

\[ \begin{align*} 
(22) \quad a. \quad x_{\text{sort}} &= \text{squeeze} \models^\delta x_{\text{theme}} = \text{orange} \\
& \quad b. \quad x_{\text{sort}} &= \text{orange} \models^\delta x_{\text{event}} = \text{squeeze} \lor x_{\text{event}} = \text{buy}. 
\end{align*} \]

Example (22b) is used to augment the current state with the information that the eventuality is of sort 'squeeze' or of sort 'buy'. If \( \delta > 0 \) holds, this means that a language user is more reluctant to draw the plausible inference. However, in the present context it is assumed that a language user only uses plausible inferences where \( \delta = 0 \). In the consequent, the existential quantifier is used, in order to capture the dynamic character of this defeasible inference since a new attribute, here \( x_{\text{event}} \) has to be introduced.
**The the . . . the-construction and the inferential user of PBVs.**

In contrast to simple sentences like ‘John squeezed an orange’, which can be analyzed in terms of only using $\kappa$, $\mu$ and $\mu^*$, both the the . . . the-construction and the inferential use of PBVs involve in addition the outcome mapping $\tau$. This is a direct consequence of the fact that they involve dependencies of (the values of) properties over time.

**The the . . . the-construction.**

Consider again example (6a), repeated here as (23).

\[(23)\] The older a stamp, the more expensive it is.

The price of a stamp is in general not only dependent on its age but also on other factors such as availability and demand. In Dependence Logic, this dependency can be expressed by (24), where $\vec{x}$ is a sequence of variables (attributes) containing ‘age’.

\[(24) = (\vec{x}, \text{price})\]

Such a functional dependency is a necessary condition for the truth of a the . . . the-construction. In addition, a stamp can get older without becoming more expensive at the same time. Thus, one only has ‘Typically (normally), a stamp gets more expensive if it gets older’. Therefore, an event of sort ‘ageing’ (or ‘getting older’) for a stamp can have at least two different outcomes. In one output only the age of the stamp has increased and in a second output both its age and its price have increased (relative to the input state). As a consequence, events of ageing for stamps are non-deterministic. Since the the . . . the-construction involves the comparative construction the . . . the, it is necessary to not only consider single transitions but sequences of such transitions defined in terms of the mapping $\tau^*$.

The output states $u_i, 1 \leq i \leq n$, of the (atomic) transitions differ in the value assigned to the outcome mapping $\tau$.\(^{15}\) Assuming $\tau(e)(w)(v) = 0$, just in case $v$ satisfies both the condition that the value of ‘Age’ has increased and that the value of ‘Price’ has increased, the most plausible histories involving a sequence of ageing events for a stamp are those in which both the stamp not only gets older but also gets more expensive.\(^{16}\)

---

\(^{15}\) They do not differ w.r.t. $\kappa$ and $\mu$ because an event of ageing leaves these rankings unchanged.

\(^{16}\) An analogous argument for other sorts of objects need not go through as in this case $\tau(e)(w)(v)$ need not be 0.
In defining a defeasible conditional based on $\tau^*$, one abstracts from the temporal development. Rather, one only uses the information that the value of the given attribute, say Age, in the input information state has increased (decreased) and that there is a corresponding change in the dependent attribute, say Price. Thus, the general form of a defeasible conditional based on $\tau^*$ has the form (25a). For (23), one gets (25b). Using this defeasible conditional, one infers that the value of the Price attribute has increased too.

(25) a. $x_{e_{sort}} = \sigma_e \land x_{o_{sort}} = \sigma_o \land x_{o_{attr}} = \beta \land x_{o_{attr}'} = \alpha \mid x_{o_{attr}} \neq \beta \land x_{o_{attr}'} \neq \alpha$.

b. $x_{e_{sort}} = \text{ageing} \land x_{o_{sort}} = \text{stamp} \land x_{o_{age}} = \alpha \land x_{o_{price}} = \beta \mid x_{o_{age}} > \alpha \land x_{o_{price}} > \beta$.

**Inferential use of PBVs.**

On its inferential use, the interpretation of a PBV involves a change. We will argue that the interpretation process is similar to an abductive argument (see Boutilier 1998) involving three steps. Consider the example ‘The melon sounds ripe’. First, there is an observation (perception): the melon emits a particular sound that is classified as ‘muffled’. Second, an explanation for this particular sound value is given by postulating some (most) plausible event or events that could have brought about the observed change in the property expressed by the verb (‘sound’ in this case). Besides a ripening event, the sound of the melon could have been manipulated mechanically. But the former event is assumed to be more plausible, say due to experience and general world knowledge. Finally, the outcomes of this event are calculated. In this case one gets that the melon is ripe. The defeasible element is the postulation of a (most) plausible event. In the case of PBVs, this is an event related to the property expressed by the predicative complement, e.g. a ripening in the case of ‘The melon sounds ripe’ where the predicative complement is ‘ripe’.

Similar to the *the . . . the*-construction, there are two constraints that must be satisfied. First, there must be a functional dependency between the two properties. For example, the value of the ‘Sound’ property must be determined by the value of the ‘Ripeness’ property. Second, this condition need only hold in the normal or typical case.\(^\text{17}\) Consider (26).

(26) *The melon sounds oval.

\(^{17}\) Gamerschlag & Petersen (2012) and Petersen & Gamerschlag (2014) formulate related constraints in their type-based frame analysis of PBVs.
In this case there is no functional dependency between the value of the property ‘Sound’ and the property ‘Form’. In a team of sort ‘melon’, the value of the ‘Form’ property can arbitrarily vary while the ‘Sound’ property remains constant, say ‘muffled’. For ‘The melon sounds muffled’, the information in the input information state is (27a). The first defeasible inference is based on the mapping $\mu$, (27b). The second step involves the non-defeasible inference that an event denoted by the verb ‘ripe’ brings about a state in which the object undergoing the ripening is ripe at the end of the event (27c). When taken together, one gets (27d).

(27)  
   a. $x_{\text{sound}} = \text{muffled} \land x_{\text{sort}} = \text{melon}$.  
   b. $x_{\text{sound}} = \text{muffled} \land x_{\text{sort}} = \text{melon} \sim x_{\text{e.sort}} = \text{ripen}$.  
   c. $x_{\text{e.sort}} = \text{ripen} \land x_{\text{sort}} = \text{melon} \vdash x_{\text{ripeness}} = \text{ripe}$.  
   d. $x_{\text{sound}} = \text{muffled} \land x_{\text{sort}} = \text{melon} \land x_{\text{e.sort}} = \text{ripen} \sim x_{\text{ripeness}} = \text{ripe}$.

5 Summary

In this paper we developed an extension of a dynamic semantic theory for natural language which makes it possible to express both dependency relations between properties of objects and expectancies between nominal and verbal concepts. The theory is based on a decompositional analysis of common nouns in which they are interpreted as ‘bundles’ of properties, similar to the way objects are represented in database theories. The ranking functions defining those expectancies are used to draw defeasible inferences from information that is provided by the lexical meaning of words in a sentence that have already been processed.

Needless to say, the theory has to be worked out in greater formal detail: (i) The relation between Dependence Logic and $\kappa$-rankings must be further explored. E.g., is it possible to define ranking functions directly in Dependence Logic?; (ii) The dynamic component must be made more explicit. In particular, how are information states for various objects modeled and how is it possible to explicitly talk about changes?; (iii) How are the rankings empirically determined? Possible approaches are strategies from n-gram models and techniques used in neuroscience based on the concept of cloze probability; and (iv) Defeasible inferences are simply added to the output information state. As a consequence, there is no distinction between ‘hard’ and ‘soft’ (defeasible) information.
References


Author

Ralf Naumann
University of Düsseldorf
nauman@phil.uni-duesseldorf.de