

Phase Quantification and Frame Theory

In my contribution I will provide an outline of how two major strands in Löbner's work can be combined in a dynamic game theoretical semantics: phase quantification (PQ) and frame theory (FT). In a first step a formal analysis of PQ in (dynamic) arrow logic is presented. Based on this analysis it is shown in the second step that frame theory must not be understood as being an alternative to standard Tarskian semantics. Rather it must be seen as an *extension* of such a semantics. The extension developed in this contribution combines the formal frame theory developed in Petersen 2007 with the analysis of PQ in arrow logic developed in this paper. In contrast to other dynamic formalisms like Dynamic Predicate Logic, the dynamic aspect is already located in the lexicon. For example, although adjectives like 'late' are basically interpreted as properties of states, they admit in addition of an interpretation where they denote relations between states (or, to be precise, basic frames representing partial descriptions of objects in the sense of Löbner 2012, 2013 and Petersen 2007).

The paper is organized as follows: In sections (1) and (2) the empirical data used by Löbner for his account of PQ as well as counterexamples discussed in Mittwoch (1993) are presented. In section (3) the formal analysis of PQ in arrow logic is developed. In the final section it is outlined how this formal analysis can be used to arrive at a satisfying formal theory of frames.

1 *Phase quantification as a major module in natural language semantics*

According to Löbner (1987, 1989, 1999), quantification in natural language is not restricted to the semantics of noun phrases but applies to a wide range of semantic phenomena including for instance adverbs of quantification like 'already' or 'still', intensifiers like 'too' and 'enough', scalar adjectives like 'few' ('small') and 'many' ('big') and phasal verbs like 'begin', 'continue' and 'stop'. Some examples are given in (1).

- (1) a. He sometimes/always/never manages to be friendly.
- b. In China you can buy Coca-Cola somewhere/everywhere/nowhere.
- c. The dollar is already/still/not yet/no longer high.
- d. This house is big enough/too big for us.
- e. In the weather forecast they said it will continue to rain/start raining/stop raining.

Löbner is aware of the fact that traditionally the above examples are not normally covered by the term 'quantification'. However, according to him, this term nevertheless refers to a seemingly very comprehensive range of phenomena which are syntactically and grammatically rather diverse but semantically closely enough related to form a class of their own (Löbner 1987:53). Löbner refers to this broadened view of quantification as *phase quantification* (PQ). PQ is characterized by the following five constraints: (i) the interpretation of PQ expressions is always based on a (monotone) scale. This scale is either temporal (the time line in the case of the 'already'-group) or non-temporal, i.e. a dimension like width or height (scalar adjectives);¹ (ii) PQ-expressions contain an implicit parameter which models a particular perspective taken by the speaker; (iii) semantically, these expressions take two arguments: a predicate P which defines a positive phase or range of values on the scale and the parameter from (ii); (iv) sentences containing PQ-expressions are

¹ The use of 'schon' is not restricted to temporal uses, as shown by (i).

(i) Basel liegt schon in der Schweiz.

Löbner (1987:81) interprets (i) as follows: „Walk along any relevant path to Basel and you will cross the border of Switzerland“.

about admissible developments which are defined in terms of two adjacent phases (called a “double-phase”) on the underlying scale. The two phases differ with respect to the fact of whether the truth conditions imposed by the predicate are satisfied on them (positive) or not (negative); (v) the existence of an admissible development is a presupposition of sentences containing a PQ-expression.

These constraints are best explained by means of an example. Consider ‘schon’ (‘already’).²

- (2) a. Es ist schon spät.
It is already late.
b. Es ist spät.
It is late.

In this case the scale is the time line and the parameter point is a temporal reference point t . The logical expression is ‘schon(t ,late)’ and an admissible development (or interval) consists of a phase during which it is not late followed by a second phase during which it is late. The parameter point t is required to fall into the second (positive) phase. Figure 1 is a pictorial representation of an admissible interval. The truth conditions for ‘schon(t ,P)’ are given in (3).

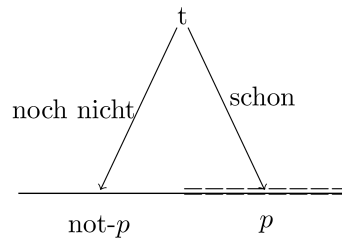


Figure 1

- (3) Truth conditions for ‘schon(t ,P)’
- ‘schon(t ,P)’ triggers the presupposition that there is a phase of not-P starting before t and that up to t at most one change between not-P and P occurred.
 - ‘schon(t ,P)’ is true iff the presupposition in (a) is satisfied and $P(t)$ is true.
 - ‘schon(t ,P)’ is false iff the presupposition in (a) is satisfied and $P(t)$ is false.
 - If the presupposition in (a) is not satisfied, ‘schon(t ,P)’ is undefined.

The presuppositions of the ‘already’-group are displayed in (4), where ‘ \Rightarrow ’ means presupposes.

- (4) a. schon p at $t \Rightarrow$ not p before t
b. noch p at $t \Rightarrow$ p before t
c. noch nicht p at $t \Rightarrow$ not p before t

In what exactly does the semantic contribution of expressions like ‘already’ consists? Or, more generally, what is the innovative feature of PQ? Let’s consider another example.

- (5) a. Das Licht ist an.
The light is on.
b. Das Licht ist schon an.
The light is already on.

² This use of ,schon’ is only one of three different uses of this expression distinguished by Löbner; see below for details.

According to Löbner (1999:51), modifying ‘spät’ with ‘schon’ “adds a sense of temporal dynamics”. He comments: “While (5a) is a stative predication about the implicit evaluation time *t*, sentence (5b) represents the same state as the result of a development from a previous state of affairs with the light not on to the present state with the light now on.” The notion of “temporal dynamics” is explained in terms of a presupposition. (5b) presupposes that the light was not on some time before *t*, i.e. (5a) was false on a relevant interval before *t*. By contrast, for (5a) no such presupposition is triggered. For Löbner, this has the effect that the meaning of a sentence involving a PQ-expression cannot be reduced to its truth conditions. In addition to the truth conditional dimension such expressions have both a procedural (or dynamic) and a cognitive dimension. Expressions involving phase quantification require information about the way in which the truth conditions came about. Löbner illustrates this view by the following procedural definition of ‘schon spät’: one starts from within the first (negative) semiphase, no matter where but, say, from its leftmost point. Next, one runs along the scale until one reaches the parameter point, which is required to lie in the double phase, and checks whether one is in the second (positive) semiphase. The conceptual dimension is described by Löbner as follows: in order to process and comprehend a sentence containing a PQ-expression, a speaker has to have the concept of the different admissible cases because otherwise (s)he is not able to mentally process its propositional content. As Löbner (1989:180) notes: “Making sense of any such sentence means constructing a specific alternative on the basis of the alternative cases as a first step, and only then, as a second step, checking (or registering, or asking, or whatever) which alternative applies.”

Empirical evidence for this analysis comes from data like the following (Löbner 1989:181f.).

- (6)
- a. Zwei plus zwei ist #schon/#noch vier.
 - b. Sie ist #schon/#noch nicht jung/Jungfrau.
 - c. Sie ist #noch/#nicht mehr alt.
 - d. Es ist schon/#noch spät.
 - e. Es ist #schon/noch früh.

Common to all examples in (6) is the fact that it is not possible to construct the required succession of two different phases, either a positive phase followed by a negative one or vice versa. (6a) is an example of an ‘eternal’ or timeless statement. (6b) and (6c) show that for temporally contingent statements, all irreversible states are incompatible with the perspective presupposed by ‘noch’, and conversely ‘schon’ excludes those states which cannot be preceded by a contrary state. (6d) and (6e) are not admissible because the underlying scale is ordered by ‘früh < spät’. From this it follows that there is no phase of lateness preceding ‘früh’ and no phase of earliness following ‘spät’. Thus, for both sentences the presupposition is not satisfied.

Löbner (1989:182) hypothesizes that the sentences in (6) are refuted *already at a level of conceptual analysis which precedes any reference to actual situations*. To quote Löbner: “To put it in terms of the analysis suggested: in these cases we know by the very conceptual content of the sentence that the set of admissible cases is degenerate.” (Löbner 1989:182)

1.1 Standard quantification and phase quantification

In contrast to Generalized Quantifier Theory (GQT), Löbner analyzes standard quantifiers like ‘all’ or ‘some’ not solely in terms of set-theoretic relations. For example, on the GQT view the meaning of a quantifier $Q(P)$ can be described as follows: $Q(P)$ is true just in case P is an element of the denotation of Q . This view is criticized by Löbner on the following ground: “Such a picture is natural in a semantic framework which has in view the truth conditions of sentences and does not consider the way truth or falsity comes about.” (Löbner

1987:79) The advantage of a procedural semantics is primarily seen in the fact that it provides criteria to choose among alternative formulations of truth conditions which are equivalent when viewed from their results but not from the way they come about. (Löbner 1987:79)

Standard quantifiers can be analyzed as an instance of phase quantification in the following way. Using the fact that quantifiers live on their domain of quantification, it follows that no other elements of the domain are relevant for the evaluation procedure. If one assumes in addition that the domain of quantification is finite, it is possible to define a linear order on the elements the quantifier lives on so that those elements which have a certain property (say the property of being human) come first. If defined in this way, the sentence ‘Some A are P’ is analyzed similarly to a sentence with ‘schon’: “start with elements of A for which P does not hold (if there are any), run through A, and you will eventually enter P, or, shorter, A reaches into P. (Löbner 1987:81) In Löbner (1987) this idea of relating standard quantification to phase quantification is made more precise in terms of semantic automata.

1.2 Phase quantification and semantic automata

In van Benthem (1986) the following two theorems are proved (see also Sevenster 2006 for details).

- (7) a. The first-order definable quantifiers are precisely those which can be recognized by permutation-invariant acyclic finite state machines.
- b. The first-order additively definable quantifiers are precisely those which can be recognized by push-down automata.

In the second theorem first-order additive logic is first-order logic extended with the ternary + relation and two constants *a* and *b*. The constant *a* is interpreted as the number of zeros and *b* as the number of ones. Formulas in this extension of FOL, then, are statements from standard arithmetic. According to these theorems, quantifiers like ‘all’, ‘some’ or ‘at least’ are recognized by acyclic finite state automata whereas quantifiers like ‘an even number of’ require for their recognition finite state automata with loops. The relation to natural language semantics is described by van Benthem (1986:151) as follows: “Viewed procedurally, the quantifier has to decide which truth value to give when presented with an enumeration of the individuals in the universe of discourse marked for their (non-)membership of A and B.” Below in Figures 2 and 3 the two automata for computing ‘all’ and ‘some’ are depicted.

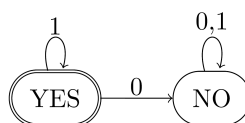


Figure 2

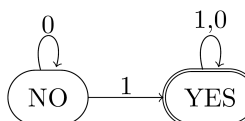


Figure 3

By unfolding such an automaton, one gets a tree (assuming that there is a unique initial state) of possible runs (or computations) (see e.g. Khoussainov and Nerode 2001 and Hollenberg 1998 for details). The automaton begins its run at the initial state, which is the root of the tree, and proceeds until it either reaches an accepting or a refuting state. Thus, edges of the tree correspond to possible moves (or behaviour) of the automaton while reading the input.

A relation between semantic automata and PQ is established by Löbner (1987:82) in the following way. The automata in Figures 2 and 3 can be considered as representing a

simple notion of border-crossing, either from p to not-p or from not-p to p. The automaton for ‘all’ can also be used for representing ‘noch’ (‘still’): “Start from a given, contextually determined point t’ where p holds (e.g. p = früh with YES the accepting state) and keep to it as long as you stay in p, but change irreversibly to the refuting state NO as soon as you encounter a time at which not-p holds, (e.g. at which it is no longer early).” However, as conceded by Löbner, these automata fail to capture the presupposition triggered by elements of the ‘schon’-group. He suggests that presuppositions can be modeled by indeterministic automata that are defined for the relevant input only, yielding no truth value (i.e. neither true nor false) if the presupposition is not satisfied. One formal possibility of defining this idea, alluded to by Löbner (1987:83), consists in defining presuppositions as additional automata that are ‘inserted’ as subroutines into automata like those in the two figures above, calculating the truth value of a corresponding sentence.

2 Some problems for phase quantification

There are a number of critical points that can be put forward against Löbner’s arguments for PQ. First, as observed by several authors, there are empirical counterexamples to central claims of PQ. On the theoretical side one has to mention that so far Löbner has never tried to formalize the above ideas, except for the short comparison to the concept of semantic automata explained above in section (1.2), and that the relation between SQ and PQ is not as neat as Löbner takes it.

2.1 Empirical adequacy

Mittwoch (1993) noted that there are obvious empirical counterexamples to the presuppositions in (4).

- (8)
- a. He/she is already rich.
 - b. The easy movement of the couplet is already there.
 - c. The Smiths have had a baby girl; they already have two sons.

These examples show that the existence of a negative phase is not a necessary condition for ‘schon’ (or ‘already’) to be admissible. (8a) can be used in a situation in which a baby is born who has come into an inheritance at birth. (8b) is appropriate to express a verdict about a poet’s very first work. Finally, what makes ‘already’ appropriate in (8c) is not the existence of a phase in which the Smiths had no children, but rather the contrast between the situation in which they only had two sons and the present situation (Mittwoch 1993:75). According to Mittwoch (1993:75), (6a) and (6b) are unacceptable solely due to the pragmatic meaning of ‘schon’ or ‘already’. This meaning involves temporal comparison of some kind. Whereas in (8a) it is comparison with some norm: one can be richer earlier than other people who attain riches, the state of being young referred to in (6b) starts at birth for everybody. Similar counterexamples can be found for the pair ‘noch nicht’ and ‘noch’.

- (9)
- a. Peters Augen waren noch nicht braun, als er geboren wurde.
 - b. #Peters Augen waren noch blau, als er geboren wurde.

As noted by Mittwoch (1993:76), there is a striking difference in acceptability between (9a) and (9b). If (9a) were simply a case of inner negation of ‘noch’, it should be as odd as (9b). However, (9b) is odd precisely because it suggests that Peter had blue eyes before his birth, which, though undoubtedly true, is irrelevant. This implication is rather due to the presupposition of ‘noch’. By contrast, (9a) lacks this implication. As a consequence, there is no need to speculate about the prenatal colour of Peter’s eyes. Mittwoch concludes that the combination of ‘noch nicht’ is not normally fully compositional and that it lacks the

presuppositional meaning component of 'noch'. Sentences with 'noch nicht' do not require a preceding phase of not-p. However, as noted by Mittwoch (1993:76), there are counterexamples to (9b).

- (10) a. Als Taschenrechner neu auf den Markt kamen, waren sie noch ziemlich teuer.
b. Als Taschenrechner neu auf den Markt kamen, kosteten sie noch 400DM.

A third set of counterexamples concerns the second and third type of uses of 'schon' distinguished by Löbner.

- (11) a. Peter hat schon drei Seiten gelesen.
b. Peter hat drei Seiten gelesen.

In this use 'schon' focuses on a time-dependent predicate. For example, in (11a) the predicate in focus is 'drei Seiten' and indicates the amount of text read so far by Peter (Löbner 1999:48). According to Löbner, the amount of material read at the parameter point t is a time-dependent function f . The meaning of (11a) can then be paraphrased as 'at t , f is already three pages'. When viewed as an instance of PQ, the predicate p is 'f is three pages'. Reading being a cumulative process (the amount of material read increases continuously with time), the negation of p , not- p , is equivalent to 'f is less than three pages'. Thus, if p is true, then there is a (negative) phase preceding the phase at which p is true at which not- p is true. However, exactly the same argument is true for the unmodified sentence (11b).

In the third type of use distinguished by Löbner, the time adverbial in focus specifies the normally implicit evaluation time t_n .

- (12) a. Peter war schon gestern da.
b. Peter war gestern da, ja er war (sogar) schon die ganze Woche da.

Similarly to the case of (11a), the admissibility of (12a) does not require a preceding phase of not- p , as shown by the example (12b).

2.2 The relation between standard quantification and phase quantification

Löbner's claim that standard quantification involving 'all' or 'some' is similar to proper PQ is open to criticism. First, there is an asymmetry between the standard (FOL) quantifiers 'all' and 'some' on the one hand and modifiers like 'schon' and 'noch' on the other. For example, whereas 'some' only requires there to be an element that is in the denotation of A and of B , 'schon' requires something stronger: in addition to 'late(t)', there must be a(n) initial preceding phase in the given admissible interval where 'late(t)' is false. Thus, for 'all' and 'some' there is only one way of how the truth conditions can be brought about. For 'all', one has to show for all elements of the domain (or a given context set) that they satisfy a particular condition (say, being mortal if being human). In the case of 'some' one gets: find some element which satisfies the property. In this respect the quantifiers are similar to unmodified 'late', which requires only a simple test to show either its truth or falsity at a given point in time. It is only by modifying this adjective with 'already' (or 'still' in the case of 'early') that one gets "a sense of dynamic development". By contrast, this component is absent in the case of the two quantifiers. Second, in contrast to cases involving elements of the 'schon' group there is in general no presupposition in the case of quantifiers. For both 'All humans are mortal' and 'Some students come from Italy', there are no corresponding sentences that can be said to 'add a sense of dynamics' triggering a presupposition similar to that of 'schon' in the case of 'schon spät'. Thus, there are no pairs corresponding to 'Es ist spät' und 'Es ist schon spät'. Rather, quantified sentences simply correspond to the

unmodified form of ‘spät’.³ Third, Löbner’s assumption that the domain of quantification can always be linearly ordered in such a way that elements having a certain property, say coming from Italy, are first in the ordering is artificial and, as admitted by Löbner, violates the condition of permutation invariance (i.e. the truth of a quantified sentence is not dependent on a particular order on the domain of quantification) for those quantifiers.⁴

An alternative interpretation of phase quantification

If there really is any concept of phase quantification, it must be possible to analyze standard quantification and phase quantification as instances of a general quantificational scheme. I will suggest, building on results from Van Benthem and Alechina (1997), that there is indeed such a general scheme.

3.1 Quantifiers as modal operators

In GQT, a monadic generalized quantifier Q is interpreted as a set of subsets of the domain in such a way that in a model M the formula $Qx\varphi$ is true just in case the set of elements which satisfy φ belongs to the interpretation of the quantifier. For the existential quantifier one gets that it is interpreted as the set of all non-empty subsets of the domain D underlying M . Similarly, the universal quantifier is interpreted as the singleton set containing only D . As shown in Van Benthem and Alechina (1997), quantifiers can also be interpreted as a special form of modal operators. Consider the Tarskian truth condition for the existential quantifier (for $\alpha = d$ or $\alpha = y$, α^{\rightarrow} is a sequence of objects or variables, respectively).

$$(13) \quad M, [d^{\rightarrow}/y^{\rightarrow}] \models \exists x\varphi(x, y^{\rightarrow}) \text{ iff there exists a } d \in D \text{ with } M, [d/x, d^{\rightarrow}/y^{\rightarrow}] \models \varphi(x, y^{\rightarrow})$$

(13) is an instance of the more general scheme (14).

$$(14) \quad M, [d^{\rightarrow}/y^{\rightarrow}] \models \diamond_x\varphi(x, y^{\rightarrow}) \text{ iff there is a } d \in D \text{ with } R(d, d^{\rightarrow}) \wedge M, [d/x, d^{\rightarrow}/y^{\rightarrow}] \models \varphi(x, y^{\rightarrow})$$

The difference between (13) and (14) is the following. In (14), the element d is required to stand in the relation R to the sequence d^{\rightarrow} , where R is an n -ary relation on the domain D so that D can be taken as structured. By contrast, in (13) one has the special case of a flat individual domain admitting of “random access”, where R is the universal relation. In view of this, (13) and (14) can also be formulated as (15a) and (15b), where R is a binary relation between elements of D and finite sequences from D .

- (15) a. $M, v \models \exists x\varphi(x)$ iff there exists a variable assignment v' which differs from v at most in its assignment of a value to x s.t. $M, v' \models \varphi(x)$.
- b. $M, v \models \diamond_x\varphi(x, y_1, \dots, y_n)$ iff there exists a variable assignment v' which differs from v at most in its assignment of a value to x s.t. $R(v'(x), v'(y_1), \dots, v'(y_n))$ and $M, v' \models \varphi(x, y_1, \dots, y_n)$ where y_1, \dots, y_n are all (and just the) free variables of $\diamond_x\varphi$ listed in alphabetic order.

Even (15b) can be generalized to (15c) where not only unary but n -ary modal operators are considered.

³ See below for details on this point.

⁴ From this it does not follow that the property of being ordered is cognitively unimportant, as shown by the following example. In a paper-and-pencil experiment Szymanik and Zajenkowski (2008) showed that on ordered domains processing sentences with the quantifier ‘most’ is easier than on unordered domains. The reaction times of people participating in the experiment were significantly faster if the domain was ordered compared to the same sentence on unordered domains.

(15) c. $M, v \models \hat{\diamond}_{x_1, \dots, x_m} \varphi(x_1, \dots, x_m; y_1, \dots, y_n)$ iff there exists a variable assignment v' which differs from v at most in its assignment of a value to x_1, \dots, x_m s.t. $R(v'(x), \dots, v'(x_m); v'(y_1), \dots, v'(y_n))$ and $M, v' \models \varphi(x_1, \dots, x_m, y_1, \dots, y_n)$ where y_1, \dots, y_n are all (and just the) free variables of $\hat{\diamond}_x \varphi$ listed in alphabetic order.

Van Benthem and Alechina (1997:1) comment: “When generalized quantifiers are viewed as first-order operators binding first-order variables, it becomes clear that a variable bound by a generalized quantifier cannot in general take any possible value. Its range is restricted, and this restriction can be defined using an accessibility relation.”

In (15a) the value of the variable x does not depend on the values of other variables, or, in terms of elements of the domain, the value of x can be chosen independently of the choice of the value of any other variable. In this respect ‘late’ is similar to the existential and the universal quantifier.

(16) $\exists t.late(t)$

However, in contrast to the two standard quantifiers, its interpretation is non-relational in the sense that no dependencies between or accessibility to other time points need to be taken into account. If ‘late’ is modified by ‘already’, the perspective changes. One is no longer interested in the property ‘late’ being simply true at a parameter point t_0 , say at speech time. Rather, the interest is restricted to those developments leading up to t_0 such that the truth value of ‘late’ is distributed on those developments in a particular way determined by ‘already’. Thus, one switches from a non-relational to a relational perspective on which not only single points but relations between points (or points and sequences of points) are taken into consideration. The first main thesis now is (17).

(17) Thesis I: The general format for PQ is the quantificational scheme in (15c).

(17) raises the question of what semantic and cognitive restrictions can be put on the accessibility relation R . From what has been said it follows that there are at least two different layers (or dimensions).

(18) a. non-relational : static
b. relational : dynamic

Consequently, there are basically three types of relations that can be relevant.

(19) a. relations at the static level
b. relations at the dynamic level
c. relations between the static and the dynamic level

Relations of type (19c) can be used for zooming in the sense of Blackburn and de Rijke (1997) and Finger and Gabbay (1992). On this perspective, the non-relational layer is used to provide information about the relational layer. At the relational level, objects can be seen as atomic objects with no internal structure, except for those structure that can be defined in terms of relations between those objects, i.e. in terms of relations of type (19b). By contrast, by using relations of type (19c), objects of the relational layer are described in a more fine-grained way by objects of a different sort.

In the context of ‘already’ and ‘still’ the domain D can be taken to be two-sorted, consisting of a sort of states and a sort of sequences of states.⁵ For temporal uses of ‘already’

⁵ See Balbiani et al. (2011) and the two appendices for details.

and ‘still’ the sort of states can be taken to be time points and the sort of sequences are intervals. For spatial uses, as in ‘Basel liegt schon in der Schweiz’, states are (spatial) points in the topological sense and the sort of sequences consists of paths. The two different sorts can be related in different ways. For example, in the temporal case, 13 possible relations, including equality, during and after, can be distinguished for two different intervals. For time points, there are three different relations: ‘before’, ‘equal’ and ‘after’. Finally, and most importantly, in the present context, there are five relations that can hold between a point and an interval: ‘before’, ‘beginning point’, ‘during’, ‘ending point’ and ‘after’. In the case of ‘already’ and ‘still’ there are three different types of relations one is interested in. At the level of intervals, the required relation is ‘meet’. For the relation between points and intervals the two relations are ‘during’ and ‘ending point’.

For each sort, there is a particular logic (or language) to talk about elements of the domain and relations holding between those elements. In addition, and most importantly, it must be possible to define relations between the two layers. The two layers are connected by two types of shifting operations (see e.g. de Rijke 1994), corresponding to the two possible types of relations in (19c).

- (20) a. static level \rightarrow dynamic level: modes (i.e. non-relational properties are analyzed in a ‘wider’ context, e.g. by describing how they ‘develop’ on a scale)
 b. dynamic level \rightarrow static level: projections (one passes from a relational view to the evaluation at a particular point)

A possible choice for the relational layer is Arrow Logic (van Benthem 1994)⁶. The basic operation of Arrow Logic is the following composition operation.

- (21) $C_{x,yz}$ x is a ‘composition’ of y and z (or, alternatively, x can be ‘decomposed’ into y and z)

The basic modal operator of an appropriate modal propositional language for expressing properties of (sets of) arrows is \bullet , whose satisfaction condition is (22).

- (22) $M, x \models \varphi \bullet \psi$ iff there exist y, z with $C_{x,yz}$ and $M, y \models \varphi$ and $M, z \models \psi$

Having the notion of an arrow together with the possibility of modeling in addition the internal structure of arrows in terms of other sorts of objects, makes it possible to use this notion as a generalization for different types of objects, in particular for Löbner’s notion of a phase. Three examples are given in (23).

- (23) temporal : intervals
 spatial (topological) : physical path
 conceptual : property

What type of second layer is used depends on the kind of objects that is modeled by an arrow. For example, in the case of properties only a beginning and an end point are distinguished without any internal structure. Next, I will illustrate this two-layered architecture by analyzing ‘already’.

The composition operation C can be used to decompose an arrow into two arrows which are sequentially related to each other (relation of type (19b)). Thus, x in (22) is an admissible interval as defined by Löbner, whereas y and z are the two adjacent phases into

⁶ See Appendix A for details.

which this interval can be split. Using C and the relation D (defined in the appendix), ‘already’ can be defined as (26a). If $p = \text{‘late’}$, one gets (24b).

- (24) a $M, s \models \text{already}(p)$ iff there are x, y, z s.t. (i) Cx, yz , (ii) $D(z,s)$, (iii) $M, y \models M, y \models \text{Int}(G\neg p)$ and (iv) $M, z \models \text{Int}(Gp)$
 b. $M, s \models \text{already}(\text{late})$ iff there are x, y, z s.t. (i) Cx, yz , (ii) $D(z,s)$, (iii) $M, y \models M, y \models \text{Int}(G\neg(\text{late}))$ and (iv) $M, z \models \text{Int}(G(\text{late}))$

According to (24b), ‘already late’ is true at a parameter point s just in case s belongs to an arrow (phase) z which is the right part of an arrow x s.t. during z ‘late’ is constantly true (with the possible exception of the left point) and during the left part y of x ‘late’ is constantly false (again with the possible exception of the left point). As it stands, (24) is not quite satisfactory. For example, it does not account for a sentence like (25), since in this case there is no phase before the parameter point during which ‘not rich’ holds.

- (25) Er war schon reich, als er geboren wurde.

This shortcoming can be remedied by using the weak Until-operator. This operator is compatible with the fact that its second argument, here p , constantly holds on the first phase. As a consequence, no border crossing needs to be involved.

- (26) $M, s \models \text{already}(p)$ iff there are x, y, z s.t. (i) Cx, yz , (ii) $D(z,s)$, (iii) $M, y \models M, y \models \text{Int}(\neg p \text{ W } p)$ and (iv) $M, z \models \text{Int}(Gp)$

(26) still makes an assertion about what holds after the parameter point so that ‘already’ has a futurate meaning, which is empirically not adequate (see Löbner 1989 for arguments and details). In this case one requires that the parameter point has to be the end point of the second phase.

- (27) $M, s \models \text{already}(p)$ iff there is an x s.t. (i) $RP(x,s)$ and (ii) $M, x \models \text{Int}(\neg p \text{ W } p) \bullet \text{Int}(Gp)$

Thus, on the present account, ‘already’ and ‘still’ semantically function as lifts (or shifts), i.e. they lift non-relational properties to relational ones. The semantic, or truth-conditional, effect of this lift consists in evaluating a static property not only with respect to a single state but with respect to a sequence of states of which this state is an element.

From what has been said so far it may seem that the relation between the two standard quantifiers and ‘already’ and ‘still’ has been lost. In order to show that this view is not correct, I will begin by considering the semantic automata from section (1.2) again.

3.2 Safety and liveness properties

There is another way of looking at the automata in Figure (2) and Figure (3). In the case of ‘all’ and ‘still’ the “border crossing” leads to a fail state or non-accepting state, i.e. the sentence is false. Thus, this state must not be attained after the automaton started at the initial state. By contrast, for ‘some’ and ‘already’ the border crossing is necessary in order to prove (the truth of) the sentence. Generalizing this observation, one gets:

- a property constantly holds (no “bad” thing happens) all, still
- a property which (possibly) fails to hold during an initial phase eventually comes to hold after some time (a “good” thing happens) some, already

No such border crossing is involved in the case of unmodified ‘late’ (or ‘early’) because it is a non-relational property. The above two kinds of properties can be defined in Temporal Logic.

A *safety* property is a property stating that “something bad does never happen.” These properties are expressed by formulas of the form (28).

$$(28) \quad \psi \rightarrow G\phi$$

In (28) ϕ is a propositional formula, i.e., a formula that does not contain any temporal operators. Intuitively, a safety property says that ϕ constantly or invariantly holds. If $\psi \equiv \text{true}$, (28) is reduced to (29).

$$(29) \quad G\phi$$

A *liveness* property states that “something good will happen”. These properties can be defined by (30).

$$(30) \quad \psi \rightarrow F\phi$$

If in (30) $\psi = \neg\phi$, a border crossing occurs. Similarly to safety properties, ϕ must not contain any temporal operators. Not all properties are safety or liveness properties. It is possible to combine the two kinds. An example is given in (31), assuming that W is taken as basic.

$$(31) \quad \psi \text{ U } \phi \equiv (\psi \text{ W } \phi) \wedge F\phi$$

In (31) $\psi \text{ W } \phi$ is a safety property whereas $F\phi$ is a liveness property.

Anticipating the discussion in section (3.4), one can say that combinations of safety and liveness properties, in particular if they involve the Until-operator, can be used to express dependence relations because they either say that a property is invariant (over a certain interval) or that its value has changed after some phase during which it didn’t hold. They therefore admit to view a property not only at a particular point (or state), i.e. in isolation, but to consider it in a broader context in which its relation to other the valuation at other states is taken into account as well.

3.3 Standard quantifiers as operations on scales

Recall that type $\langle 1 \rangle$ quantifiers in natural language live on a set A (Peters and Westerstahl 2006:89).

$$(32) \quad \text{If } Q \text{ is a type } \langle 1 \rangle \text{ quantifier, } M \text{ a universe and } A \text{ any set, then } Q_M \text{ lives on } A \text{ iff, for all } B \subseteq M, \text{ one has } Q_M(B) \leftrightarrow Q_M(A \cap B).$$

This property is a characteristic trait of restricted quantifiers. If Q_M lives on A , knowing for any subset B of M whether or not the quantifier holds of it reduces to looking at those elements of B which also belong to A . For type $\langle 1, 1 \rangle$ quantifiers like ‘all’ or ‘some’, it is possible to freeze the restriction argument as follows (Peters and Westerstahl 2006:110).

$$(33) \quad \text{If } Q \text{ is any type } \langle 1, 1 \rangle \text{ quantifier, and } A \text{ is any set, the type } \langle 1 \rangle \text{ quantifier } Q^A \text{ is defined, for all } M \text{ and all } B \subseteq M, \text{ by } (Q^A)_M(B) \leftrightarrow Q_{A \cup M}(A, B).$$

The effect of freezing is to reduce a type $\langle 1, 1 \rangle$ quantifier to a type $\langle 1 \rangle$ quantifier. By holding the restrictor argument constant (or frozen), it becomes possible to view it as a scale with respect to which elements of B (or $A \cap B$ due to the property of living on) can be checked,

whether they satisfy the required property or not. On this scale, either a ‘good’ thing happens or no ‘bad’ thing happens. In particular, one gets (34), where p is the property corresponding to the set B .

- (34) a. \forall : Gp (safety property : no border crossing)
 b. \exists : Fp (liveness property : border crossing)

We are now able to characterize the similarities and differences between the various forms of phase quantification.

(35) Thesis II: Common to all types of phase quantification is the fact that the truth conditions can be defined in terms of combinations of safety- and liveness properties of sequences or, more generally, arrows.

The various types differ in at least the following two respects.

- The standard quantifiers \forall and \exists are always defined.⁷
- The standard quantifiers \forall and \exists are permutation invariant.

As I will now show, these two differences are not independent of each other. As was shown in section (2.1), ‘already’ does not require that there be an initial phase during which $\neg p$ holds (36a). However, it is admissible only if this possibility exists, at least theoretically (36b,c). Similarly, ‘still’ imposes the condition that p eventually becomes false, otherwise it, too, is not admissible (37).

- (36) a. Er war schon reich, als er geboren wurde.
 b. # Das Auto ist schon neu.
 c. # Es ist schon früh.
 (37) a. # Er war noch alt, als er starb.
 b. # Es ist noch spät.

These constraints are not imposed by the two standard quantifiers. For example, \exists is compatible with the fact that all elements of the domain have the property expressed by p , i.e. one has Gp for all enumerations x . By contrast, if Gp holds for all x , then ‘already’ is not admissible. This difference can be explained if one considers the differences with respect to the cognitive significance of (combinations of) safety and liveness properties. This difference is the topic of the next section.

3.4 The cognitive significance of phase quantification

When viewed from the point of view of cognitive linguistics, the most important question with respect to modifiers like ‘already’ and ‘still’ is: what do they add in addition to the simple assertion that p holds at the parameter point t ? What are the consequences in processing the modified sentence in the brain? Only getting the information that it is late at the parameter point, solely conveys information about that particular point. It does neither give him/her information about what happened before nor about what is likely to happen afterwards with respect to the property of being late. Thus, there is an epistemic or informational uncertainty for the comprehender about what happened before t and about what is likely to happen after t with respect to the truth value of ‘late’. Such information is not provided by (unmodified) ‘late’.

⁷ Possible counterexamples are empty restrictor sets as in ‘All unicorns are tall’.

What is the cognitive relevance (significance) of resolving such epistemic uncertainties? First, there is a gain in the amount of information the comprehender gets. For example, (s)he not only knows that it is late at the parameter point but that during some interval (phase) before that point it was not late. Second, this gain in information can be used for strategic planning or to revise and adapt one's current projections (or expectations) of how a discourse (or a piece of communication) will continue. Simplifying somewhat, one can summarize the cognitive function of 'already' and 'still' as follows: resolving epistemic uncertainties allows a comprehender to eliminate certain possibilities of how a result came about or how it will continue to hold or develop. This helps reducing both processing and memory load during semantically parsing a sentence in the brain.

Let's consider the examples in (38), some of which have been discussed before.

- (38) a. Peter verfügt über ein Millionenvermögen. Er wurde schon reich geboren.
 b. The Smith have had a baby girl. They already have two sons.
 c. #Es ist schon früh.
 d. #Es ist noch spät.

When a comprehender comes to know that Peter is rich, he does not know how he acquired his riches.⁸ The second sentence in (40a) provides additional information. He might have inherited his money from his parents or some other source. A possible, though defeasible, conclusion that can be derived from this additional information is: probably, his riches are not due to his own achievements. An analogous argument applies to (40b). Upon learning that the Smiths have had a baby girl, I don't know how many children they have. Or, to put it in game-theoretical terms: I don't know the exact number of children in the "Smiths having children" game. In the case of (40c) and (40d) composition of the corresponding game with another game does not result in a gain of information or a reduction in epistemic uncertainty. For example, if it is early in the morning at the parameter point, then it has been early for all other points belonging to the interval denoted by 'this morning' preceding the parameter point. As a consequence, no new information is provided about how this state came about so that this information is redundant at the cognitive level.

Thus, after lifting 'late' to a relational property, its truth value at the parameter point t_0 depends on the truth value assigned to this property on a sequence (or arrow) the end point of which is t_0 , or, to put it differently, only those assignments of the value 'true' to the property at t_0 are admissible that also have $y_k = \text{false}$ for $1 \leq k \leq m$ and $y_j = \text{true}$ for $m + 1 \leq j \leq n$ for some m with n the length of the sequence and t_0 being one of the y_j . Thus, the value of t_0 (true in this case) is dependent on the values of the sequence y^{\rightarrow} : $R(x, y^{\rightarrow})$. On this perspective, elements of the 'already'-group not only lift a non-relational property to a relational one, but, in addition, they exclude some possible relations. As an effect, the relation R must not be the universal one, i.e. admit of random access, because in that case no (new) information would be added by triggering this lift.

By contrast, for the standard quantifiers, the cognitive relevance does not consist in eliminating epistemic uncertainty but rather in establishing (or building up) relations between the values of different attributes which have been learned in encounters with the world. They express relations between the values of *different* properties, whereas 'already' and 'still' express relations between the value of a *single* property at *different* states. Therefore, it does not matter whether the domain is ordered and whether some ways of bringing about the truth are excluded. These constraints only apply if a single property is viewed at different states. The above considerations are summarized in the table below.

⁸ Of course, one has to assume that the comprehender does not already have information about this point of Peter's life. Otherwise, he would get no new information.

	type of property	relation
late	atomic	non-relational
all	safety	relation between values of different properties
some	liveness	
still(late)	combination of safety and liveness	relation between the values of a single property at different states
already(late)	combination of safety and liveness	

Table 1

The property of border crossing refers to the type of property. Safety properties forbid such a border crossing, whereas (combinations of) safety and liveness properties require it. For atomic properties, the concept of border crossing does not apply.

4 Phase Quantification and Frame Theory

In this final section I will relate the analysis of section (3) to the frame theory that is being developed in the CRC 991 ‘The Structure of Representations’ at the University of Düsseldorf, the huge and international project led by Sebastian Löbner. This theory will henceforth be called the Düsseldorf Frame Model. Following Barsalou (1992), Löbner (2013) argues for the following two claims: (i) the human cognitive system operates with *one* general format of representations and (ii) if the human cognitive system operates with one general format of representations, this format is essentially a Barsalou frame. This Barsalou-Löbner Frame-Hypothesis (BLFH) requires a frame model that is sufficiently expressive to capture the diversity of representations and that is sufficiently precise and restrictive in order to be testable. Given these two constraints, it follows that the gap between cognitive linguistics, brain science and formal semantics has to be filled (Petersen 2013). A formalization of the BLFH was presented in Petersen (2007). In this formalization linguistic items, or the concepts expressed by them, like sortal nouns, say ‘bottle’ or ‘dog’, are modeled as typed feature structures (see Petersen 2007 for details). For example, a possible frame for the sortal noun ‘bottle’ is given in Figure 4 (this figure is taken from Gamerschlag et al. 2013, see Löbner 2013 for the reference).

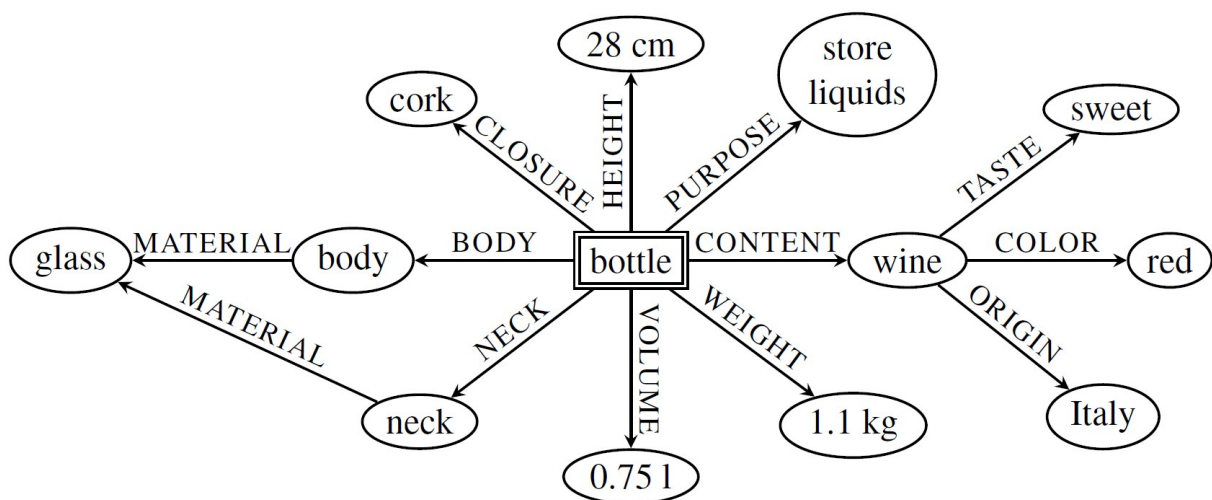


Figure 4

According to Löbner (2013), such a frame is a parametrized description of an object. The basic building blocks are attribute-value pairs. Attributes are functions which are defined for a certain type of possessor and which assign to every possessor of the appropriate type a unique value from a set of admissible values. For example, the attribute COLOR assigns possible colour values to the objects of type ‘visible (monochrome) object’. In the specific bottle example above, the attribute CONTENT specifies that the bottle contains wine whose origin is Italy (value of the attribute ORIGIN) and which tastes sweet (value of the attribute TASTE). Value specifications can be more or less specific, depending either on the amount of information that is available about the object or on the level of abstraction at which the object is described. Frames like that in Figure 4 can be taken as representing an atemporal or static (partial) snapshot of a bottle. What is not captured is the possibility that the value of an attribute may eventually be changed, say as the effect of an action or an event, or simply by the passing of time.

In Naumann (2013) frames in the Düsseldorf Frame Model were called Petersen Frames. They were formalized as pointed Kripke-models in the following way. Given a signature $\langle P, \text{Attr} \rangle$, a Petersen Frame Model (PFM) is a triple as given in (39).

- (39) $\langle S, \{R_a\}_{a \in \text{Attr}}, s_0, V \rangle$ with
- S a (non-empty) set of nodes (or states), the domain of the model,
 - each R_a is a (functional) binary relation on S ,
 - V is a valuation function that assigns to each $p \in P$ a subset of S ,
 - s_0 is the central node of the frame.

An example of a language for talking about PFMs is an extended modal language (see Naumann 2013 and Naumann and Petersen 2013 for details). However, this formalization of frames raises at least the following two serious issues: (i) if frames can be reduced to a particular type of feature structures, what is *specific* about a theory of frames, or, to put it differently, is there really a *genuine* theory of frames?, and (ii) in what exactly does the cognitive significance of frames lie?

From the perspective of the approach developed in section (3), the situation can be analyzed as follows. In previous formalizations of the BLFH only the truth-conditional dimension of frames has been taken into consideration. However, if truth conditions are taken as primary, the semantic value of a lexical item is reduced to (or is completely determined by) its contribution to the truth conditions of sentences.⁹ For example, in standard Tarskian semantics, the meaning of an expression in formal semantics is usually identified with the (constant) contribution it makes to the truth conditions of sentences in which it occurs. For example, intransitive verbs like ‘run’ or adjectives like ‘late’ or ‘cool’ denote sets of entities like persistent objects (‘run’) or time points (‘late’). By contrast, in a dynamic setting like Dynamic Predicate Logic (DPL), expressions are interpreted as (generalized) relations between (information) states.

$$(40) \quad \lambda x \lambda s \lambda s'. \|\text{Expr}\|(x)(s)(s')$$

However, in DPL atomic predicates like ‘run’ or ‘late’ are interpreted as tests, i.e. the input and the output state are identical so that their meaning can be reduced to that in a static (standard) Tarskian framework.

$$(41) \quad \lambda x \lambda s \lambda s'. \|\text{Expr}\|(x)(s)(s') \wedge s = s'$$

⁹ Thus, the additional meaning components do not consist in intersentential relations (anaphora) as in DRT or Dynamic Predicate Logic.

In the formal framework developed in section (3) adjectives like ‘late’ or ‘cool’ are basically analyzed as properties of states (or time points). As a consequence, their meaning can be identified with the contribution they make to the meaning (truth conditions) of a sentence in which they occur. However, given the way modifiers like ‘already’ and ‘still’ are analyzed in section (3), those adjectives are usually interpreted in a higher or lifted type. Intuitively, at this level the meaning corresponds to what Löbner calls an admissible interval. Using the distinction between a static and a dynamic component, the question becomes: how can a static component be integrated into a dynamic one?

One way of arriving at such an integration or combination consists in using the technique of combining systems (Finger and Gabbay 1992, de Rijke 1994). In such frameworks a global and a local component (layer) are distinguished. Models have the form $M = \langle S_g, \dots \rangle$ with the global component given by the ‘...’. The set S_g represents the local component. S_g is a set $\{m_i\}_{i \in I}$. Each m_i can itself be a model, and thus having a complex structure. In the setting of the Düsseldorf Frame Model the local layer, i.e. S_g , corresponds to the static dimension and therefore consists of a set of PFMs, which captures atemporal snapshots of an object or entity. The global component models the dynamic layer and is given by the arrow-models from section (3). The global-local distinction is paralleled by a (possible) distinction with respect to the languages (or logics) that are used to talk about the two layers (de Rijke 1994:174). First, there is a global language which talks about global aspects of the structure but not about local ones. Second, there is a local language which is used to describe elements of S_g .

For PQ, a combined model can be defined as follows.

- (42) A Dynamic Frame Model (DFM) is a triple $\langle \{P_f\}_{f \in F}, R, AS \rangle$ such that
- the elements of P_g are PFMs¹⁰,
 - AS is an arrow structure which is used to describe how the objects denoted by elements of P_g change,
 - R is a relation on $P_g \times A \times P_g$ which combines the local with the global layer. Intuitively, $(m, x, m') \in R$ if ‘executing’ an arrow in input m results in m' as output¹¹,
 - values of attributes in a PFM represent the values of a properties of a possessor before a change occurred. PFMs are static in the sense that only the contribution to the truth conditions of sentences are captured. A PFM is a partial description of an object and this description is true at a parameter point just in case the object exhibits the values of those properties expressed by attributes in the PFM.
 - a DMF represents the evolution of a property (or a set of properties) of an object with respect to a particular dimension (or a set of dimensions) and is therefore relational.

In the context of PQ, examples for arrow models are

- events or actions which change the value of a property (or the values of a set of properties) of an object.
- the flow (or passage) of time. On this perspective, arrows can be taken as time intervals.
- physical paths connecting regions (or points) in space (or space-time).

¹⁰ The domain of Petersen frame models will be ordered by a subsumption relation (see Carpenter 1992 for details).

¹¹ There will in general be constraints imposed on R. For example, for a pair (m, x) , $R(m, x)$ is required to be a singleton.

Arrows are a separate domain of the model and must therefore not be identified with binary relations on the domain S_g . Thus, one has $(m,x,m') \neq (m,x,m')$ if $x \neq x'$. By contrast, were R be defined as a binary relation on S_g , one would have $(m,m') = (m,m')$, which trivially holds.

So far no constraints have been imposed on the relation R , i.e. R can be an arbitrary relation on $P_g \times A \times P_g$. A first, and obvious, constraint imposes the correct core frame-semantic meaning. For example, in the case of ‘late’ any admissible transition must end in a PFM for which the value of the attribute TIME is ‘late’ (or ϕ_{late}). By itself, ‘late’ does not impose any further conditions. As a consequence, it is compatible with any transition that ends in a ‘late’-state. The contribution of modifiers like ‘already’ or ‘still’, then, consists in restricting this model to a submodel where each transition is admissible according to the constraints imposed by the modifier.

When taken together, one arrives at the following four hypotheses.

Hypothesis 1: The core frame-semantic meaning of an expression includes its (standard) static Tarskian meaning which is defined in terms of the contribution it makes to the truth conditions of sentences. This meaning component is captured in terms of PFMs.¹²

Hypothesis 2: The proper frame-semantic meaning of an expression is defined in terms of DFMs, which specify possible ways of how the core feature-semantic meaning, expressing its contribution to truth conditions of sentences, can be brought about. This is its dynamic meaning component and is part of its cognitive meaning.

Hypothesis 3: The frame-semantic meaning of an expression can be given in terms of DTM-formulas of the form

$$\pi \rightarrow \diamond_{RP}\phi$$

where π is a formula of the global language expressing how the truth came about and ϕ expresses the (static) truth conditions. Such a formula is satisfied by an arrow if it satisfies π and the truth conditions expressed by ϕ are true at its right (end) point (boundary). For example, in the case of ‘late’ one gets (43).

$$(43) \quad M, x \models \text{Int}(\neg \text{late } W \text{ late}) \bullet \text{Int}(G(\text{late})) \rightarrow \diamond_{RP} \text{late}$$

(43) expresses a relationship between the dynamic level of an arrow and its static component. (43) can be read as “if an arrow x can be decomposed into subarrows y and z satisfying π , then ϕ holds at the end point of the arrow”.

Hypothesis 4: Identical contributions to truth conditions modeled by the same PFM can correspond to different DFMs capturing the dynamic (cognitive) meaning of the expression (or concept).

Core frame-semantic meanings are expressed in terms of the language that is used to talk about PFMs, e.g. an extended modal language (see Naumann 2013 for details). The proper frame-semantic meaning is expressed in terms of the language used for talking about the global layer of a DFM. For phase quantification, this is the language defined in section (3).

Of course, phase quantification is an example of this fourth hypothesis: ‘It is already late’, ‘It is still late’ and ‘It is late’ have the same truth conditions (it must be late at the parameter point or at speech time), however the constraints they impose at the level of DFM

¹² This last claim need not necessarily hold for dynamic or action verbs like ‘eat’ or ‘hit’.

are different. Using Hypothesis (3) their difference consists in the DFM formula π while they all have the same formula $\diamond_{RP}\phi$ in the consequent, expressing the fact that the sentences have the same truth conditions. Depending on π , different types of information about the way the truth conditions have come about are conveyed by the sentences. As a consequence, different (additional) conclusions, like those discussed in section (3.4), can be inferred, reflecting the difference in their cognitive value (or in their cognitive meaning).

References

- Balbani, P, V. Goranko and G. Sciavicco (2011). Two-sorted point-interval temporal logics, *Electronic Notes in Theoretical Computer Science*, 278, 31-45.
- Barsalou, L. W. (1992). Frames, Concepts, and Conceptual Fields. In A. Lehrer and E. F. Kittay (eds.): *Frames, fields, and contrasts: New essays in semantic and lexical organization*. Hillsdale, NJ: Lawrence Erlbaum Associates, 21–74.
- Blackburn, Patrick and de Rijke, Maarten (1997). Zooming in, Zooming out, *Journal of Logic, Language and Information*, 6, 5–31.
- De Rijke, M. (1994). Meeting some neighbours, In Van Eijck, J. and A. Visser (Eds.), *Logic and Information Flow*, Cambridge, MIT Press, 170-195.
- Finger, Marcelo and Gabbay, Dov (1992). Adding a temporal dimension to a logic system, *Journal of Logic, Language and Information*, 2, 203–233.
- Hodges, W. and J. Väänänen (2010). Dependence of variables construed as an atomic formula, *Annals of Applied and Pure Logic*, 161, 817-828.
- Hollenberg, M. (1998). *Logic and Bisimulation*, PhD thesis, University of Utrecht.
- Khoussainov, B. and A. Nerode (2001). *Automata Theory and its Applications*, Birkhäuser, Basel.
- Kröger, F. and S. Merz (2008). *Temporal Logic and State Systems*, Heidelberg, Springer.
- Löbner, S. (1987). Quantification as a major module of natural language semantics' In J. Groenendijk, M. Stokhof, D. de Jongh (Eds.), *Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers*. Dordrecht: Foris, pp. 53-85.
- Löbner, S. (1989). German *schon – erst – noch*: An integrated Analysis, *LP*, 12, 167-212.
- Löbner, S. (1999). Why German *schon* and *noch* are still duals: a reply to van der Auwera, *LP*, 22, 45-107.
- Löbner, S. (2012). Functional concepts and frames, ms. University of Düsseldorf.
- Löbner, S. (2013). Evidence for frames from natural language. To appear in T. Gamerschlag, D. Gerland, W. Petersen, R. Osswald (eds.), *Frames and concept types: applications in language and philosophy*. Studies in Linguistics and Philosophy 94. Dordrecht: Springer.
- Mittwoch, A. (1993). The relationship between *schon/already* and *noch/still*: A reply to Löbner, *NLS*, 2, 71-82.
- Naumann, R. (2013). Outline of a Dynamic Theory of Frames. In Guram Bezhanishvili, Sebastian Löbner, Vincenzo Marra & Frank Richter (eds.): *Logic, Language, and Computation*, 115-137, LNCS Vol. 7758, Berlin, Heidelberg: Springer.
- Naumann, R. and W. Petersen (2013). Frames as strategies, paper presented at Tbilisi 2013.
- Peters, S. and D. Westerstahl (2006). *Quantifiers in Language and Logic*, Oxford, Oxford UP.
- Petersen, W. (2007): Representation of Concepts as Frames. *The Baltic International Yearbook of Cognition, Logic and Communication*, 2, 151-170.
- Sevenster, M. (2006). *Branches of Imperfect Information: Logic, Games, and Computation*. PhD Thesis, ILLC, Universiteit van Amsterdam.
- Szymanik J. and M. Zająkowski (2010). Comprehension of simple quantifiers - empirical evaluation of a computational model, *Cognitive Science*, 34, 521-532.
- Van Benthem, J. (1986). *Essays in Logical Semantics*. Reidel, Dordrecht.
- Van Benthem, J. (1994). A note on dynamic arrow logic, In Van Eijck, J. and A. Visser (Eds.), *Logic and Information Flow*, Cambridge, MIT Press, 15-29.

Van Benthem, J. (2002). Extensive games as process models, *Jolli*, 11, 289-313.

Van Benthem, J. (2012). In praise of strategies, In Van Eijck, J. and R. Verbrugge (Eds.), *Games, Actions, and Social Software*, LNCS Volume 7010, Springer, Heidelberg, 96-116.

Van Benthem, J. and N. Alechina (1997). Modal quantification over structured domains, In M. de Rijke (Ed.), *Advances in Intensional Logic*, Kluwer, Dordrecht, 1 - 27.

Appendix A: Arrow logic and two-layered systems

Arrow Logic¹³ is based on the intuition that binary relations can be interpreted as denoting sets of arrows. Examples are arcs in graphs, transitions in Labeled Transition Systems, attributes in attribute-value structures or even preferences if they are used as ranking relations. Arrows can have internal structure so that they need not be identified with ordered pairs because different arrows can have the same source (beginning point) and target (end point). Conversely, there may be points that are not related by arrows. An arrow frame is defined as follows.

- (1) Arrow frames are tuples (A,C,R,I) with
- | | | |
|----|----------|-----------------------------------------|
| a. | A | a (non-empty) set of objects ('arrows') |
| b. | C,x,yz | x is a 'composition' of y and z |
| c. | Rx,y | y is a 'reversal' of x |
| d. | Ix | x is an 'identity' arrow |

If a propositional valuation V is added to such a frame, one gets an arrow model with the following satisfaction relation.

- (2)
- | | | | |
|----|----------------------------------|-----|----------------------------------------------------------------------------|
| a. | $M, x \models p$ | iff | $x \in V(p)$ |
| b. | $M, x \models \neg\phi$ | iff | not $M, x \models \phi$ |
| c. | $M, x \models \phi \wedge \psi$ | iff | $M, x \models \phi$ and $M, x \models \psi$ |
| d. | $M, x \models \phi \bullet \psi$ | iff | there exist y,z with Cx,yz , $M,y \models \phi$ and $M,z \models \psi$ |
| e. | $M, x \models \phi^\circ$ | iff | there exists y with Rx,y and $M, y \models \phi$ |
| f. | $M, x \models Id$ | iff | Ix |

Arrow frames (models) are combined with state frames (models).

- (3) State frames are pairs (S, \leq) with
- | | |
|----|-------------------------------------------|
| a. | a (non-empty) set of states |
| b. | \leq a partial (or linear) order on S |

There are the following mechanisms of interaction (bridges) connecting the two components.

- (4)
- | | |
|----|--------------------------------------------------------------------------------|
| a. | $LP \subseteq A \times S$, mapping an arrow to its beginning (or left) point. |
| b. | $RP \subseteq A \times S$, mapping an arrow to its end (or right) point. |

Both LP and RP are required to be functional, i.e. both $LP(x)$ and $RP(x)$ are singletons. In terms of LP and RP the relation D ('during') between arrows and states is defined as follows.

- (5) $D(x,s)$ iff $LP(x) < s \leq RP(x)$

To LP and RP correspond the two modalities defined in (6).

¹³ See van Benthem (1994) and the references cited therein for details.

- (6) a. $M, x \models \diamond_{LP}\varphi$ iff $M, LP(x) \models \varphi$
 b. $M, x \models \diamond_{RP}\varphi$ iff $M, RP(x) \models \varphi$

Since one is mainly interested in the lifting of non-relational properties that can be expressed using one of the variants of the Until-operator, state formulas are evaluated on sequences $\gamma = s_0s_1\dots s_k$ in such a way that an atomic formula p is true on a sequence if it is true at its beginning point s_0 , (7a). In (7b)-(7c) the clauses for Int, corresponding to D, as well as for G and F are given.

- (7) a. $M, \gamma \models p$ iff $M, s_0 \models p$ for p a state propositional variable
 b. $M, x \models \text{Int}(\varphi)$ iff $M, D(x) \models \varphi$
 c. $M, \gamma \models G(\varphi)$ iff for all suffixes γ' of γ : $M, \gamma' \models \varphi$
 d. $M, \gamma \models F(\varphi)$ iff for some suffix γ' of γ : $M, \gamma' \models \varphi$

The definition of the Until-operator U is given in (8a). In (8b) the weak variant W of the Until-operator is defined. It is compatible with φ being constantly true.

- (8) a. $M, s \models \psi U \varphi$ iff there is an s' with $s < s'$ and $M, s' \models \varphi$ and for all s'' with $s < s'' < s'$: $M, s'' \models \psi$.
 b. $\psi W \varphi \equiv (\psi U \varphi) \vee G\varphi$

The intuitive meaning of the two variants are given in (9) (see Kröger and Merz 2008:66).

- (9) a. “There is a strictly subsequent state in which φ holds, and ψ holds until that state.”
 b. “ ψ does not become false before a state where φ holds is reached.” (“ ψ waiting for φ ”)