Principles as lexical methods

Jacques Jayez
EHESS-CELIITH
54 Boulevard Raspail
75006 Paris France
jjayez@dial.oleane.com

Danièle Godard
CNRS, Université Paris 7, UFRL
case 7003, 2 Place Jussieu
75005 Paris France
Godard@Univ-Lille3.fr

Abstract

Current research in lexical semantics has brought into focus (i) the existence of general tendencies ('principles') across languages or inside language families, (ii) the existence of numerous exceptions to these tendencies. To reconcile these two facts, it seems natural to endorse a general conception of 'principles-with-exceptions' or 'subregularities', usually cashed on a form of default logic. This leaves open the question of how to identify and manage the exceptions in a more precise way, i.e. not just as marginal by-products of a subregularity system. In this paper, we analyze different examples where complex cases can be handled explicitly, and show that principle-based approaches require, as a natural complement, to be controlled by more local information.

Introduction

It is widely acknowledged that some additional mechanisms are necessary to interpret even simple sentences, which elaborate on the more standard modes of combination between predicate and arguments. Such extra devices are illustrated by (pragmatic) metonymy (Nunberg 1978; 1993), coercion and (logical) metonymy (Pustejovsky 1991; Pustejovsky & Boguraev 1993; Hays & Bayer 1991; Stallard 1993), lexical extension rules (Copestake & Briscoe 1994), or interpolation (Pollard & Sag 1994; Godard & Jayez 1993). Since they exhibit some form of generality, those 'interpretation boosters' can be viewed as principles, that is, mutually independent resources which can be used freely to generate interpretations when needed. However, it has been suggested that cross-linguistic variation as well as language-internal exceptions make this strong view untenable in the long run (Nunberg & Taenen 1992; Copestake & Briscoe 1994). Our point in this paper is to show that some of these 'principles', those which are based on the sorting of linguistic entities, are locally controlled by (classes of) lexical items and constructs, which select (combinations of) operations to be activated, and possibly fix their parameters. Such general processes are in fact rather similar to methods, in a sense akin to that of object-oriented languages. While methods can be shared by several different objects, their application depends on the individual properties of these objects. In the same way, linguistic entities individually incorporate interpretive instructions. So-called principles lead no independent life. They are formed by abstracting away from these individual constraints. We will first point out some empirical problems for a principle-based approach. We then describe methods for handling sorts in the combination of lexical items, and show them at work through several illustrations.

Problems for a principle-based approach

There are general strategies which all involve a form of sort management. We show in this section that formulating them as general principles is not empirically sufficient, if possible at all. Their usefulness crucially depends on working out their precise conditions of application, which in turns depends on scrutinizing lexical properties. We illustrate here some problems encountered by general-purpose strategies.

Accessing the qualia structure

Pustejovsky's qualia structure and coercion (Pustejovsky 1991; Pustejovsky & Boguraev 1993) offer the opportunity to generate various interpretations without listing them up in a rigid way. E.g., the adjective fast can combine with nouns such as typist or book to give interpretations such as 'a person who types fast', or 'a book which can be read quickly'.

---

1In Proceedings of the AAAI workshop on representation and acquisition of lexical knowledge, March 27-29, Stanford University, pp. 57-68

2We leave aside problems connected with metaphor and pragmatic metonymy, although they also raise questions about sorting.
However, it is not possible to predict which role a
given A will access. In French, A’s like *lent, rapide*
can access the same role as ‘fast’ (viz. telic role),
if the noun denotes a person: the NP *une dactylo*
*lente/*rapide means exactly ‘a slow/fast typist’. But
these A’s do not access the telic role of *livre*. *Un livre*
lent/*rapide does not mean ‘a book which it takes a
long/short time to read’.

Turning now to various A’s belonging to the same
broad semantic class (temporal A’s), which modify the
same N ‘book’, we must recognize that there is no ob-
vious way to explain which roles each of these A’s will
preferentially access. *Un livre lent* does not mean ‘a
book which it takes a long time to read’, but ‘a book
whose plot proceeds at a slow pace’. So, the sort re-
quired by lent/*rapide is no longer telic-event in this
case: the A is predicated of the book content. In fact
the role which is accessed depends on the N.

We would expect rapide (‘quick’, ‘fast’) to access the
same role(s) as lent when used with the same noun; in
effect, they both modify the book content. However,
*un livre rapide* can also mean ‘a book which has been
written quickly’ (agentive role), a possibility which is
not easily open to lent. The adjective *interminable*
(‘endless’), when applied to book, can access the telic
role, the agentive role, and can be predicated of the
book content (the plot is endless). The question is
then and why these A’s can modify the book content
or the reading time.

It is intuitively satisfying to connect the plot and
the reading durations: an endless script, with a bor-
ing succession of episodes, is likely to cause an endless
reading. So, in this case, lexical access structure just
fits our naïve knowledge. But why does this seemingly
natural link not show up in the lexicon in the similar
cases of lent or rapide? Should not a slow plot induce a
slow reading, and a fast-paced plot favour a quick one?
A possible answer is simply that *lent* and *rapide* bear
on the *rhythm*, that is on the average number of salient
events. So, a short book with only a few interesting(episodes can be read quickly, and still be perceived as
*lent*.

Yet, how are we to explain the fact that *un livre*
*long/*court refers to the book length rather than to the
telic role? A long book should entail a long reading in
most cases. One cannot appeal to the book con-
tent to argue that an easy-to-read long book can be
read quickly. The same sort of argument goes for an
endless book: a book with a fascinating entangled sce-
nario can be read quickly. We conclude that there is
no obvious representational explanation for the scope
of those adjectives.

**Interpolation**

When the combination of a predicate with its argu-
ments does not supply an acceptable interpretation,
it is necessary to construct intermediate structures
which provide the missing interpretive link. This phe-
nomenon occurs in control structures (Pollard & Sag
1994, 7.5) as well as in simple V + complement NP
sequences. In the latter case, it has been proposed
(Pustejovsky 1991; Godard & Jayez 1993) that the
class of interpolated predicates be specified by general
abstract sort constraints. E.g. to *begin the room* may
be interpreted as ‘to build, paint, refurbish, clean the
room’, because these various predicates all inherit from
a sort modify (cf. (Godard & Jayez 1993) for details).
Unfortunately, no convincing generalization on these
constraints emerges from the inspection of a variety of
verbs.

First, such abstract constraints may be difficult to
substantiate. For instance, *attendre* (‘to wait for’) has
many interpolative uses, which we may divide into two
classes. In the first one, the interpolated predicate
seems to inherit from an abstract sort ‘come to be’, its
precise instantiation depending on the sort of the NP:
*j’attends l’arrivée du train* (event NP) means ‘I am
waiting for the arrival of the train to happen’,
*j’attends mon dernier livre* (product NP) means ‘I am wait-
ing for my last book to come out’, *j’attends la voiture*
(mobile-obj NP) means ‘I am waiting for the car to
arrive’. We will focus on the second one, exemplified
by another sense of *j’attends la voiture*, meaning for
example ‘I am waiting for my car to be cleaned, mend-
ed’, or ‘to be delivered’ in a situation in which I have
ordered a car. It seems that interpolated predicates
involve a notion of control, that is, X *attend Y roughly*
means ‘X has (temporarily) lost X’s control on Y,
because some Z has it, and X is waiting to resume X’s
control’.

The point here is that it is extremely difficult to flesh
out the notion of control in specific cases. For instance,
it seems that ‘control’ is relevant only for objects which
are privately possessed (not machines which are for
public use, such as cigarette machines), but which are
not available at a given moment because another per-
son retains them (for use, repair, etc.). Moreover, some
personal objects, whose use is suspended for repair, do
not provide good complements for *attendre*. E.g. it is
strange to say *Il fumait une cigarette en attendant la
porte* (‘He was smoking a cigarette while waiting for
the door’), in a context where a person is waiting for
a given door to be fixed in her house.

Second, it is not even clear in some cases that we
appearance to abstract sorts constraints rather than simple
lexical forms. For example, the verb *suggérer* + NP is
interpreted as ‘to suggest that one choose NP’: suggérer Pierre means that Pierre would be the right choice for some relevant purpose in the situation. Sort–driven interpretation would be of no help, since the interpolated predicate ‘choose’ is not recoverable from any intuitively convincing abstract sort for propositions (Asher 1993), nor from the object NP, since one can choose or suggest practically any kind of entity.

So, interpolation is represented in these cases as the insertion of predicates corresponding to lexicalized forms (verbs), rather than abstract predicates.

Pronominal anaphors and sort conflicts

The sort of an NP may be complex, that is, it may have a conjunctive form \( \sigma_1 \land \ldots \land \sigma_n \). For instance, the French NP in (1) has the sort mat–obj \( \land \) info–obj.

(1) Ce livre lourd est riche d’enseignements

‘This heavy book is full of interesting ideas’

A strong regularity is that the sorts event and mat–obj do not easily coexist, as shown in:

(2) ??La symphonie a été interrompue parce qu’elle a été emportée par l’orage

‘The symphony was stopped because it had been blown away by the storm’

While a pronominal anaphor does not have to be of the same sort as its antecedent, the two sorts must be compatible. Thus, (3) is acceptable:

(3) Ce livre est très lourd, mais il est riche d’enseignements

‘This book is very heavy, but it full of interesting ideas’

but not

(4) *La construction est très bien située [mat–obj]. Elle [event] a eu lieu l’année dernière.

‘The construction is in a very nice location. It took place last year’


‘The meal is on the table. It will take place in one hour’

Yet, some predicates such as disposer (‘to arrange’) reveal a complex sort for the noun, namely mat–obj \( \land \) arrangement, which smooths the sort transition between otherwise incompatible members of a NP-pronoun pair.

(6) *Le repas est disposé sur la table. Il aura lieu dans un heure

‘The meal is arranged on the table. It will take place in one hour’

(6) is much better than (5). This is only possible with certain nouns, e.g. with repas but not construction.

(7) *La construction est disposée sur la colline. Elle a eu lieu l’année dernière.

In this case again, a general regularity on sort compatibility is defeated by detailed lexical behaviour, whose description requires that small classes of items (not just broad sort–based classes) be taken into account.

The general framework: extended sorts + methods

We treat lexical structures as objects (in the sense of object–oriented languages) on which operations can be performed. This requires two basic extensions to current feature structures representations.

(a) We must give internal structure to our objects (e.g. qualia structure), as all general purpose KR formalisms do (cf. (Nebel 1990)). We start from standard HPSG descriptions (Carpenter 1992) augmented with relevant semantic indications, and a notion of sort–as–distribution.

For N’s\(^2\), we posit two new attributes dsorts and sdsorts. They contain information about the N’s distribution, i.e. the set of constructions in which it can occur.\(^4\) Such each set is abbreviated by an abstract label, called dsort, for distributional sort. E.g. the dsort event\(^5\) abbreviates a conjunction of constructions where a NP can be the complement of some temporal prepositions, or the subject of avoir lieu (‘to take place’). NP’s such as le tremblement de terre (‘the earthquake’) or la cérémonie (‘the ceremony’) are classified as event because they occur in such environments. In general, NP’s can be classified as possible complements of classes of verbs (Le Pesant 1994). In some cases, a dsort may be just a label for a list of items which are considered to be in the same class, e.g. the dsort arrangement below.

Since an N may get several dsorts, the attributes dsorts and sdsorts contain in fact boolean combinations of dsorts, which are called dsort formulas. When some N is endowed with more than one dsort\(^6\), we say

\(^2\)We will assume here that the relevant information for N’s is passed to NP’s, and will not distinguish N’s and NP’s in the examples to come.

\(^4\)Although selectional sorts enter into systematic relations with semantic sorts, we do not a priori assume that there is a one–to–one correspondence between the two.

\(^5\)For space reasons, we will not delve into the subsorts of event here. Cf. (Goddard & Jayez 1994).

\(^6\)In this case, a conjunction of the relevant dsorts is used if they must be present conjointly, while a disjunction is used if this is optional or forbidden (mutually inconsistent sorts).
it that is multisorted. E.g. the N livre (‘book’) is mat–obj and info–obj: intuitively, it can denote a material object as well as an informational structure.

The value of DSORTS expresses the intrinsic sorting of the N. So, it remains unchanged in the course of unification. The value of SDSORTS (selectional dsorts) is initially the same as that of DSORTS, but is updated to comply with the selectional restrictions of the verb. For instance, the verb poser (‘to put’) requires its complement to be of sort mat–obj. The N repas (‘meal’) has a sort formula mat–obj ∨ event, and mat–obj and event are incompatible. So, in the VP poser le repas, the value for SDSORTS will change to mat–obj.

In HPSG, information about the arguments of a lexical item in a construction are registered in a special list, the value of the ARG–s feature, whose members are the values of the SYNSEM features in the arguments. DSORTS and SDSORTS are located in the SYNSEM, in order to take advantage of the standard unification process of HPSG. More precisely, we will assume that DSORTS and SDSORTS are head features; this will ensure that they are propagated to their projections.

(b) We provide messages and methods which create new information states. In essence, we apply certain operations, echoing, in a unification–based framework, those usually defined in the context of record semantics (Cardelli & Mitchell 1991). Typically, these operations update some given description and return the result. Lexical items can of course carry with them several interpretation operations.

There are three reasons to prefer methods to standard typed unification: (i) One avoids to posit as many lexical entries as there are distinct semantic behaviour for an item. (ii) Dynamic effects, such as on–line generation or modification, are possible. (iii) Checking the presence of an attribute in a description will result in adding it to the description in a unification–based framework, but this is clearly unwanted in some cases. However, in the present treatment, we will extend unification only to a minimal extent.

A description D (Carpenter 1992) is a boolean combination of constraints and/or descriptions, where a constraint can be a sort assertion, a path–value declaration, or a path convergence declaration. We will consider only satisfiable (consistent) descriptions.

We extend the definition of (Carpenter 1992) by saying that the path π is in D (π ∈ D) if D contains a constraint π : σ or a constraint π′ : σ such that π″ ∈ σ and π = π′π″.

a ∪ b notes, as usual, the unification of terms a and b. List unification holds for lists of the same length whose corresponding elements unify.

Paths of labels ℓ₁ · · · ℓₙ are noted ℓ₁ | ℓ₂ · · · | ℓₙ.

We consider three basic methods φᵢ over the domain of satisfiable descriptions: sort checking srtch(), which checks the subsumption properties of an object, addition addif, which adds path–value declarations to an object, and replacement rpl() which replaces values inside an object. The result of applying φ to x will be noted φ(x). A method is any sequence φ₁(φ₂(···φₙ(x)···)) possibly with free variables. Disjunctions and conjunctions of methods can also be used. We define an atom for halting, terminate, such that, for any method µ, µ(terminate) = terminate.

• sort checking

Let D and D’ be two descriptions, we say that D subsumes D’, in symbol D’ ⊨ D, whenever any feature structure satisfying D’ satisfies D.

\[
srtch(D, D') = \begin{cases} \text{terminate}, & \text{if } D' \not\models D \\ D', & \text{otherwise} \end{cases}
\]

• Addition. The addif(D’, π, D) operator adds elements of information D’ at the location π in D. If a more precise information already exists at the same location, D is unchanged.

\[
\text{addif}(D', \pi, D) = \begin{cases} D & \text{if } srtch(\pi : D', D) = D \\ D \cup \{ \pi : D' \} & \text{otherwise} \end{cases}
\]

• Replacement replaces the value of a given path by another value.

\[
\text{rpl}(\pi, y, D) = (D - \{ \pi \}) \cup \{ y \}
\]

\[
\text{pathrep}(\pi, \sigma, D) = \left( D - \{ \pi : \sigma | \pi : \sigma' \in D \} \right) - \\
\left( \{ \pi' : \sigma' | \pi = \pi' \pi'' \land (\pi' : \sigma') \in D \land \pi'' \in \sigma' \} \right) \\
\cup \\
\{ \pi : \sigma \} \cup \\
\{ \pi' : \text{pathrep}(\pi'', \sigma'' : \sigma', \sigma') | \pi = \pi' \pi'' \land \\
(\pi' : \sigma') \in D \land \pi'' \in \sigma' \}
\]

Treatment of the examples

In what follows, to make the treatment uniform, we will suppose that methods are used in the course of or after a parsing process, which is sensitive to categorial syntactic information, in that it can group items into basic categories (S,VP, NP, etc.). This restriction is only for clarity, and does not preclude the application of methods to generation, or other tasks.

Adjective and noun combination

The general idea in this section is that A modifiers update the value of cont for the A (and thus for the N) when they combine with nouns. The standard HPSG representation for a N such as rapide + N is shown (in a simplified form) below:
We will make two basic assumptions. (i) CONT can contain a more complex information, that is, additional attributes and their values. For instance, an attribute QUALIA for the qualia structure will be found in CONT. (ii) The choice of a method depends on the value of the SDSORTS. Typically, an adjective must look for the sort formula of the noun it modifies, and resort to the appropriate methods.

We illustrate the method–basis approach with the case of rapide (‘fast’, ‘quick’), to which we associate a method which is in fact a disjunction of methods.

**Conventions**

The HPSG paths are labelled as follows:

- adjarg for \[ ADJ-DTR \] | \[ SYNSEM \] | LOCAL | CONT | RESTR
- nounind for \[ HEAD-DTR \] | \[ SYNSEM \] | LOCAL | CONT | IND
- nounhead for \[ HEAD-DTR \] | \[ SYNSEM \] | LOCAL | CAT | HEAD
- nouncont for \[ HEAD-DTR \] | \[ SYNSEM \] | LOCAL | CONT

\( D \) is a description for the above feature structure and \( D' \) is the description \( \text{adjarg} = \text{nounind} \). Note that \( D' \) is a term of \( D \).

We assume that RHYTHM and DURATION occur in the values of ACTIVITY, as well as in the CONT value of event nouns. More generally, types like activity or event are interpreted, in the language of descriptions, as descriptions containing minimally RHYTHM : \{ \{ \} \} ∧ DURATION : \{ \}.

**Methods**

We will deal with the following cases\(^8\) :

- \( \text{N} \) is an event (‘a quick construction’); the method is:
  \[
  \text{repl}(D' , \text{adjarg} = \text{nouncont} | \text{RHYTHM} , \text{srtrh} (\text{nounhead} \{ \text{DSORTS} : \text{event} , D \}))
  \]

A similar method is defined for DURATION. These methods simply declare that the relation rapide bears on the value of RHYTHM or DURATION.

- For animal and machine (‘a fast horse / typist / printer’), the A will add ACTIVITY. We define:
  \[
  \text{add}((\text{ACTIVITY} : \text{activity}) , \text{nouncont} , \text{srtrh} (\text{nounhead} \{ \text{DSORTS} : \text{animal} ∨ \text{machine} , D \})) = \mu (D)
  \]

and use the simple embedding (shown only for DURATION):

\[
\text{repl}(D' , \text{adjarg} = \text{nouncont} | \text{ACTIVITY} | \text{RHYTHM} , \mu (D)).
\]

For break–event (‘a fast resigning’\(^9\)), we replace the totality of nouncont, by a method of form

\[
\text{repl}(\text{nouncont} , \text{nouncont} : x , D) \quad \text{where} \quad x \quad \text{is:}
\]

\[
\text{REL} : \text{and} ∧ \text{ARG1} : (\text{REL} : \text{rapide} ∧ \text{ARG} : \{ \}) ∧ \text{ARG2} : (\text{REL} : \text{before} ∧ \text{ARG1} : (\text{DURATION} : \{ \}) ∧ \text{ARG2} : \text{nomin} ∧ \text{ARG1} | \text{ARG} = \text{ARG2} | \text{ARG1} | \text{DURATION}
\]

The intended interpretation is that the duration of the interval which precedes the event denoted by the noun is short.

- For path (ways of communication: ‘a fast road’), we just use the telic role, which is supposed to be of sort activity. The method is (for RHYTHM):

\[
\text{repl}(D' , \text{adjarg} = \text{nouncont} | \text{QUALIA} | \text{TEVIC} | \text{RHYTHM} , D).
\]

- For info–obj (books, texts, documents, etc.), we will assume that the adjective can modify the value of the agentive role in the qualia, or an INFO–CONT role whose value must contain at least one of the two attributes USE–INFO and SCRIPT–INFO. The first one is appropriate for dictionaries, technical treatises, and corresponds to the presence of an information independent of any plot or script.\(^10\) The second one is appropriate for any form of story and its value contains RHYTHM. Rapide can access the two, which produces the two readings ‘fast plot’ and ‘insufficient information’. The method is analogous to the previous ones.

Since these methods are independent they can be activated in parallel, and several cumulative results can be produced. E.g. a break–event can itself have a duration, so that the corresponding construction gets at least two readings.

Although methods resemble each other in some respects, their control remains ‘local’. E.g. leni looks for a RHYTHM attribute, producing a clumsy combination if it does not find it. This is why it cannot be applied to a dictionary, or in general, to any form of documentation. Moreover, not just any RHYTHM attribute will do for leni: telic and agentive role, which provide activity values with a RHYTHM attribute, are not systematically accessed.

**Interpolation**

When the interpolated predicate is restricted, as in the suggérer example, interpolation is a straightforward replacement. If suggérer + NP has a content of the following form, where \text{encont} abbreviates the path lead-

\(^8\)There are other cases not described here, e.g. those of the sorts state, feeling, function.

\(^9\)This sort labels events which constitute the borderline between two opposite situations. E.g. the event démission (‘resigning’) separates a situation in which one is in charge of some professional or official activity from a situation in which one has refused this responsibility.

\(^10\)It is an open question whether there exist clear distributional criteria for identifying those classes (Le Pesant 1994).
ing to the complement’s content:

\[
\text{(REL : suggest \& suggest \rightarrow } - | \text{SUBJ} | - | \text{IND} \&
\text{suggested = compcont | IND} = D
\]

can one define a method:

\[
\text{repl(suggested = compcont | IND, x, D), where x is}
\]

\[
\text{(proposition \& REL : choose \& Chooser : person \&}
\text{chosen = compcont | IND).}
\]

For attendre, we have to compensate for the relative opacity of the ‘control’ notion. Instead of using a set of abstract conditions on the possible sorts of the predicate, we adopt a hybrid approach. We first specify the interpolated predicate as follows. It can be:

- any verb of sort happen for NP’s of sort event,
- any verb of sort produce, e.g. fabriquer (‘to make’), préparer (‘to prepare’), construire (‘to build’),
- any synonym of arriver (‘to arrive’),
- any synonym of donner (‘to give’) or rendre (‘to give back’).

The indices of the arguments vary according to the predicate. For instance, for a form X attend Y, the interpolated form with produce is X attend Z produce Y. It is X attend Z give Y to X for donner.

The method for donner is given as an example:

\[
\text{repl(waited-for = compcont | IND, x, D), where x is}
\]

\[
\text{(proposition \& REL : give \& Giver : person \&}
\text{beneficiary = HEAD-DTR-} - | \text{SUBJ} | - | \text{IND} \& \text{given =}
\text{compcont | IND)}
\]

Most often, the choice over predicates depends on the context. E.g. J’attends le livre de Pierre (‘I am waiting for Peter’s book’) can mean that the speaker is waiting for Peter’s book to be given to her, to be written, or even to arrive by mail. Contrary to what appears at first sight (see above), it should be noted that the choice of a predicate does not coincide with the sorts of the NP complement. The important point is that we do not resort systematically to abstract constraints on predicates, because they are difficult to flesh out in a coherent way. The differences between commencer (‘to begin’), attendre, suggester, and other interpolative verbs show clearly that a general mechanism for interpolation construction is out of reach, and that interpolation must be conceived as locally determined by classes of verbs and arguments.

Sort drifting

We want to account both for the general compatibility constraint and the possibility of overriding it. The verbs like disposer, which reveal that the sort formula of some NP’s is compatible with a pronoun of SDSORTS of type event, constitute a small class, which can be listed.\(^{11}\) We call it order. A sort formula for the NP’s of the repas class will be mat-obj \& arrangement. Because of the selectional frames, the SDSORTS of le repas is mat-obj when it is the complement of poser, mat-obj \& arrangement when it is the complement of disposer. Normally, a pronoun must get a sort which is unfiable both with the value of SDSORTS and of DSORTS for its antecedent. Thus, (5) is strange because the SDSORTS values mat-obj and event do not unify. But, if a strong notion of inconsistency is used, the unification of mat-obj \& arrangement with event, which is found in (6), will also fail, which is not what we want.

An obvious solution is to substitute a nonmonotonic relation term unification for the traditional term unification\(^{12}\). Using ⊨ for strict consequence and ⊢ for nonmonotonic one, we can posit:

\[
\text{event ⊨ mat-obj ▷ undefined}
\]

\[
\text{event ⊨ mat-obj ▷ arrangement ⊨ event ▷ mat-obj ▷ arrangement}
\]

Although there exist reasonably simple frames for dealing with such dependence systems (Schulz 1991), they also raise questions in the framework of unification, which we leave aside here.

**Conclusion**

Contrary to a principle-based approach, the interpretations we have sketched are not driven by some principle(s) which can apply to the case (success) or not (exception). In object-oriented parluce, we would say that lexical objects can trigger a given (combination of) method(s) whenever they satisfy explicit control conditions (Martin & Odell 1995, chap. 17), which range from simple presence in a list to abstract constraints. The effective triggering may still depend on factors such as default ordering of methods, pragmatic cues, or probabilistic hierarchy, which leave room for indeterminacy. Describing a core of lexical preferences in a given idiom does not exclude other interpretive processes. It is required for constructing a layered approach based on an explicit (partial) hierarchy of strictness/relaxation, which puts principles in their place, not just anywhere.

**References**


\(^{11}\) *agencer, arranger, disposer, installer, mettre en place, préparer,* etc.

\(^{12}\) Actually, unification by ⊨ returns a pair whose first element is a conjunction, and second element an evaluation of the ‘goodness’ of the result, e.g. (event ▷ mat-obj ▷ arrangement, tolerable).


