Constraints: an operational framework for Constructions Grammars
Philippe Blache

To cite this version:
hal-00135434

HAL Id: hal-00135434
https://hal.archives-ouvertes.fr/hal-00135434
Submitted on 7 Mar 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Constraints: an operational framework for Construction Grammars

Usually, linguistic theories make use of a hierarchical representation of syntactic information. Even when no relation with tree-like structures is explicit, as for HPSG ([Sag99]) or Construction Grammars (see for example [Kay99]), a structure is described in terms of a hierarchy. This aspect is reinforced by lexicalization: representing syntactic information at the lexical level comes with feature localization in terms of position into a hierarchy, feature propagation being controlled through the head/mother structure. This conception, close to the generative way of representing information, is very rigid in the sense that building a hierarchical structure is a pre-requisite and one cannot say anything about syntactic properties of non-canonical (not to say non-grammatical) utterances. Construction grammars also suffer from this problem. Moreover, for the same reasons, the only way to represent local constraints is sub-constructions inheriting from higher level frames. Let’s take the example of the SAI construction (see [Fillmore98]):

1. Did you learn your lesson?
2. Did you really learn your lesson?
3. Didn’t you learn your lesson?
4. *Did really you learn your lesson?
5. *Did you not learn your lesson?
6. *Did you learn your lesson?

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linearity (→)</td>
<td>Linear precedence constraints.</td>
<td>Det &lt; N</td>
</tr>
<tr>
<td>Dependency (→)</td>
<td>Dependency relations between categories.</td>
<td>AP → N</td>
</tr>
<tr>
<td>Obligation (→)</td>
<td>Set of compulsory and unique categories. One of these categories (and only one) has to be realized in a phrase.</td>
<td>N → NP</td>
</tr>
<tr>
<td>Exclusion (≠)</td>
<td>Restriction of co-occurrence between sets of categories.</td>
<td>N[pro] ≠ Det</td>
</tr>
<tr>
<td>Uniqueness (Uniq)</td>
<td>Set of categories which cannot be repeated.</td>
<td>U(req NP) = { Det, N, AP, PP, Pm }</td>
</tr>
</tbody>
</table>

On the basis of such properties, it is possible to specify for an utterance, whatever its form, its syntactic specificities. The process is simple: given a set of categories corresponding to the lexical entries, properties of the grammar can be evaluated. Some of them are satisfied, some other can be violated (this is the case of non-canonical inputs). The result is a syntactic characterization of the input, formed by these two sets of properties. A construction can be detailed as a set of such constraints:

1. \( V[aux] ⊏ NP[subj] \)
2. \( NP[subj] ⊏ V[-fin] \)
3. \( V[aux] ⊏ NP[subj] \)
4. \( V[-fin] ⊏ XP[-sub] \)
5. \( Adv[\neg] ⊏ NP[subj] \)
6. \( NP[subj] ⊏ Adv[-\neg] \)
7. \( NP \sim V \)
8. \( Adv \sim V \)

This subset of constraints \( \{1,2,4,7,8\} \) represents the information of the SAI construction represented above and we can say that these two notations are equivalent. However, adding new
information such the one concerning the adverbs simply consists in adding new constraints to the
set describing the construction, at the same level, without needing any inheritance mechanism.

The interest of such a constraint-based approach is that constraints are used as properties
making it possible to identify a construction. In our example, when the subset of constraints
\{1,2,4,7,8\} is satisfied, we know that we have a SAI construction. Then other properties (for
example semantic ones) specific to this construction can be applied. Moreover, it is possible to
identify constructions even for ill-formed inputs by means of constraint relaxation techniques.
Using constraints allows in conclusion to represent all information contained in constructions in a
more flexible way.

References

Natural Language Processing, D. Christodoulakis (ed), Lecture Notes in Artificial Intelligence 1835,
Springer-Verlag


Aspects of Linguistic Explanation, Stanford University.

ture, Chicago University Press.

what’s x doing y construction”, Language.


Press.

[Sag99] Sag I. & T. Wasow (1999), Syntactic Theory, A Formal Introduction, CSLI.