STRONG GENERATIVE CAPACITY
OF CLASSICAL CATEGORIAL GRAMMARS

Classical categorial grammars (CCG's) are the grammars introduced by Ajdukiewicz [1] (under the influence of Leśniewski's theory of semantical categories) and formalized by Bar-Hillel [2], Bar-Hillel et al. [3]. In [3] there is proved the weak equivalence of CCG's and context-free grammars (CFG's) [6]. In this note we characterize the strong generative capacity of finite and rigid CCG's, i.e. their capacity of structure generation. These results are more completely discussed in [4], [5].

Let $V$ denote a countable set whose members will be referred to as atoms. The set $FS(V)$, of functorial structures ($F$-structures) over $V$, is defined as follows:

1. $V \subseteq FS(V)$,
2. if $A_1, \ldots, A_n \in FS(V)$ ($n \geq 2$) then $(A_1 \ldots A_n)_i \in FS(V)$, for all $1 \leq i \leq n$.

Given an $F$-structure $(A_1, \ldots, A_n)A_i$ is called the functor, and each $A_j$, $j \neq i$, an argument of this $F$-structure. The notion of a substructure of some $F$-structure is defined in the natural way. $A_1, \ldots, A_n$ are called parts of $(A_1, \ldots, A_n)$. The size of $A \in FS(V)(s(A))$ is the maximal number of parts of substructures of $A$. A sequence $A_0, \ldots, A_n$ of substructures of $A \in FS(V)$ is called a branch in $A$ (of length $n$) if, for all $1 \leq i \leq n, A_i$ is a part of $A_{i-1}$. A branch $A_0, \ldots, A_n$, such that $A_i$ is the functor of $A_{i-1}$, for all $1 \leq i \leq n$, is called an $F$-branch. The external degree of $A \in FS(V)(d_e(A))$ equals the length of shortest branches in $A$ which lead from $A$ to some atom, and the degrees of $A(d(A))$ is the maximal $d_e(B)$, for
B ranging over substructures of A. The F-degree of \( A \in FS(V)(d_F(A)) \) equals the maximal length of F-branches in A.

Any set \( L \subseteq FS(V) \) is called a functorial language (F-language) over \( V \). By \( \text{sub}(L) \) we denote the set of all substructures of the F-structures from \( L \). We also set:

\[
\begin{align*}
(3) \quad &d(L) = \sup\{d(A) : A \in L\}, \\
(4) \quad &d_F(L) = \sup\{d_F(A) : A \in L\}, \\
(5) \quad &s(L) = \sup\{s(A) : A \in L\},
\end{align*}
\]

and we call these numbers the degree, F-degree and size of \( L \), respectively. \( FS(V) \) can be treated as the absolutely free algebra generated by \( V \) with operations \( (. . .)_i \). By \( \text{Int}_L \) we denote the largest congruence on the structure \( (FS(V), L) \) (treat \( L \) as a monadic predicate on \( FS(V) \)), and we call it the intersubstitutability relation for \( L \). We define the index of \( L \) \( (\text{ind}(L)) \) as the number of equivalence classes of \( \text{Int}_L \). Clearly, \( d(L), d_F(L), s(L) \) and \( \text{ind}(L) \) may be finite or countably infinite.

Phrase structures (P-structures) amount to F-structures which lack functor markers. By \( PS(V) \) we denote the set of all P-structures over \( V \). For \( A \in PS(V) \), we define \( s(A), d_e(A), d(A) \) as for F-structures. Similarly, for \( L \subseteq PS(V) \) (such a set \( L \) is called a P-language), we define \( d(L), s(L) \) and \( \text{ind}(L) \) as for the case of F-languages.

We fix, a denumerable set \( Pr \) of primitive types and we define \( Tp = FS(Pr) \). The members of \( Tp \) are called types. By a CCG we mean a triple \( G = (V_G, I_G, s_G) \), such that: \( V_G \) (vocabulary) is a countable set, \( I_G \) (initial type assignment) is a function from \( V_G \) into \( P(Tp) \), and \( s_G \) (principal type) \( \in Pr \). By \( Tp(G) \) we denote the union of all \( I_G(v), \) for \( v \in V_G \). A CCG \( G \) is said to be finite (resp. rigid) if \( Tp(G) \) is finite (resp. \( I_G(v) \) contains at most one type, for all \( v \in V_G \)).

Each CCG \( G \) determines the terminal type assignment \( T_G \), being defined as the smallest subset of \( FS(V_G) \times Tp \) fulfilling the conditions:

\[
\begin{align*}
(6) \quad &\text{if } x \in I_G(v) \text{ then } vT_Gx, \\
(7) \quad &\text{if } A_iT_G(x_1 \ldots x_n)_i \text{ and } A_jT_Gx_j, \text{ for } j \neq i, \text{ then } (A_1 \ldots A_n)_iT_Gx_i.
\end{align*}
\]

We identify \( T_G \) with a function \( T_G : FS(V_G) \to P(Tp) \).
Now, the $F$-language generated by a CCG $G(FL(G))$ consists of all $A \in FS(V_G)$ such that $s_G \in T_G(A)$. By dropping functor markers in the $F$-structures from $FL(G)$ we get the $P$-language generated by $G(PL(G))$, and by dropping the brackets in the $P$-structures from $PL(G)$ we get the language generated by $G(L(G))$. It is known [3] that the languages of finite CCG’s with a finite vocabulary coincide with those of CFG’s. Our notion of a CCG admits both $V_G$ and $I_G$ infinite (see [7] for some nice applications of CCG’s with an infinite initial type assignment).

An $F$-language is said to be (resp. finitely, rigidly) stratifiable if it equals $FL,(G)$, for some (resp. finite, rigid) CCG $G$. We prove:

**Theorem 1.** Each $F$-language is stratifiable.

**Theorem 2.** An $F$-language $L$ is finitely stratifiable iff each of the numbers $s(L), d_F(L)$ and $ind(L)$ is finite.

A $P$-language is said to be (resp. finitely) stratifiable if it equals $PL(G)$, for some (resp. finite) CCG $G$. By theorem 1, each $P$-language is stratifiable. As concerns finite stratifiablity, we obtain:

**Theorem 3.** A $P$-language $L$ is finitely stratifiable iff each of the numbers $s(L), d(L)$ and $ind(L)$ is finite.

To characterize rigid stratifiability we need an auxiliary notion. For $L \subseteq FS(V), A, B \in FS(V)$ we write $A \prec L B$ if, for some $C \in sub(L)$, $B$ is the functor of $C$ and, either $A$ is an argument of $C$, or $A \mathbin{Int}_L C$.

**Theorem 4.** An $F$-language $L$ is rigidly stratifiable iff there hold the following conditions:

(i) the relation $\prec_L$ is well-founded,
(ii) if $A, B \in L$ then $A \mathbin{Int}_LB$,
(iii) if $A \prec_L B$ then $B \notin L$,
(iv) if $(A_1 \ldots A_n)_i, (B_1 \ldots B_m)_j \in sub(L)$ and $A_i \mathbin{Int}_LB_j$, then $m = n$, $i = j$ and $A_k \mathbin{Int}_LB_k$, for all $1 \leq k \leq n$,
(v) if $(A_1 \ldots A_n)_i, Int_L(B_1 \ldots B_n)_j \in sub(L)$ and $A_j \mathbin{Int}_LB_j$, for all $j \neq i$, then $A_i \mathbin{Int}_LB_i$. 

References


Institute of Mathematics
Adam Mickiewicz University
Poznań, Poland