An Algebra for Tracing Categories of Social Processes: From a Surprising Fact to Middle-Range Theory using Categorical-Generative Analysis[1]

Bruno da Rocha Braga[a,b]

[a] Central Bank of Brazil, SBS Quadra 3 Bloco B, 70.074-900, Brasília, Brazil
[b] Federal Institute of Education, Science and Technology, 610 N., 70.860-100, Brasília, Brazil
bruno.rocha.braga@ifb.edu.br

Abstract. This paper describes a method for the analysis of the evolutionary path of a complex, dynamic, and contingent social phenomenon in an empirical setting. Given empirical evidence of a surprising or anomalous fact, which contradicts the prediction of the wide-acknowledged theory, the goal is to formulate a plausible explanation based on the context of occurrence, taking a holistic and historical point of view. The procedure begins by translating theoretical propositions into grammar rules to describe patterns of either individual action or interaction that may occur within the hypothesized social system. The result is a category of social process in which the objects are types of decision-making events carried by a stable community of actors over time, and the relationships between them are state transitions revealed in the sequences of event outcomes. Therefore, structural comparison between pairs of representative instances result in an extension of the category of social processes, relying on the configurations of contextual conditions that enable the occurrence of the new event outcome in specific empirical settings.

Keywords: Computational Linguistics; Critical Realism; Generative Social Science; Pragmatism; Theory of the Firm.

[1] The first draft of this paper dates back to 2019. This second version (2022), which included the example of the Theory of The Firm, is also available at https://dx.doi.org/10.2139/ssrn.4332535.

1 Introduction

The difference between the objects of study in the natural sciences and the social sciences arises from the observation that people interpret the objective reality in many subjective ways. Specifically, the human mind perceives and interprets external signals using a categorical framework of its own. Additionally, due to the high uncertainty that results from subjectivity, explanation prevails over prediction in social research, and attempts to forecast future event outcomes are limited to contingent tendencies that may eventually become reality.

Despite their distinct natures, the transposition of methods of scientific research from the natural sciences into the social sciences without prior consideration of their ontological assumptions is common practice under the auspices of the epistemological guidance of the mainstream paradigm known as social positivism. This problem is particularly notable in economics, where the capability of prediction remains the main criterion for scientific assessment of works in this discipline. There is nothing inherently wrong with the mainstream research approach in economics and other social sciences, as they reflect both the goals and beliefs of their communities, which are often divergent. However, there are efforts of reconciliation among many fields of the social sciences towards some common tendencies.

The most promising trend in the social sciences seems to be towards formulating middle-range theories, which involve modeling a real-world phenomenon based on empirically testable propositions deduced from universal laws, rather than formulating such an abstract, grand theory (Merton, 1968). The diffusion of this approach to theorizing among the social sciences is due to the acknowledgement of the complex nature of social phenomena, which requires empirically grounded methods. Nonetheless, complexity is not fully intelligible using the methods of the so-called normal science, which rely on the assumptions of Social Positivism, particularly the so-called näive-realism.

Ironically but not surprisingly, the alternative to normal science still involves the transposition of recent results from some fields of mathematics and physics, as shown in the book "A New Kind of Science" (Wolfram, 2002). The hard sciences are already familiar with complexity as a justification for emergent phenomena, which are patterns of higher-level properties and behaviors that result from lower-level interactions among their constituent elements. It is ambitious to assume that the physical world operates using automata, but it is not so far-fetched to assume that the social world, particularly organizations, operate in this way since these systems often result from our deliberate rational actions.

Nonetheless, mathematical models such as complex adaptive systems and dynamical systems still fail to predict future event outcomes in the emergent level of social phenomena because of the structure and agency relation, which is absent in natural phenomena. The agents are not static reactive objects, but they evidently learn from social situations, such that they may still change their own behavior and that of the surrounding social structures in unpredictable ways. Consequently, social systems are epiphenomena of networks of individuals acting under the guidance of consensual rules, such that the result is not only collective, orderly behavior but also emergent macrostructures that seem to exhibit properties and behavior of their own. Even under the assumption of perfect knowledge of the individual decision rules, the evolutionary path of the emergent social forms and processes appear to remain largely independent from lower-level developments and is highly sensitive to any small differences in the initial conditions. The computational power of modern computers already enables agent-based modeling and simulation of complex social systems (e.g., markets and organizations) relying on the assumption of the ontological equivalence of structures and agents to computational devices. Nevertheless, there is a need for a proper definition of social science, which implicates its own goals and methods, not necessarily the same as those of the natural sciences.

Generative Social Science is such an interdisciplinary effort towards the understanding of complex, dynamic, and contingent behaviors as the emergent result from interacting agents using computational models and simulation techniques (Epstein, 2006). The term "generative" still comes from Noam Chomsky's Generative Grammar Theory (1957), which is a linguistic theory that relies upon rule systems to produce infinite valid constructions based upon a finite set of elements. Clearly, the elements and constructions that are the objects of studying in Generative Social Science are not alphabet symbols and sentences of a natural language, but individual actions and social interactions instead.

Generative Grammar Theory has influenced many fields of science, including computer science in the specification of the syntax of programming languages, medicine in the study of the immune system (Jerne, 1985), and economics in the study of institutions for the governance of common goods (Crawford & Ostrom, 1995). Nonetheless, most works still rely on the assumptions of social positivism, rejecting any relevant distinction between science and social science. Some of these assumptions include: (i) methodological monism, which argues for the existence of one scientific method; (ii) epistemological objectivism, which holds that true knowledge comes from perception, independent of subjective interpretations or biases; (iii) the demarcation criterion that discriminates between scientific and non-scientific statements relying on the logic of falsification rather than the logic of verification; (iv) causal explanation as a provisional statement that is hypothetically deduced from universal laws of nature, including of human nature, which is true until being refuted by an empirical test; and (v) methodological individualism, which holds that all scientific explanations must refer to evidence about individuals and their interactions, rather than to emergent social forms, properties, or behaviors.

In this way, Braga (2017a) proposed a generative approach for modeling social processes as patterns of sequences of decision-making events that support the qualitative assessment of complex, dynamic, and contingent behavior in specific empirical settings using a grammar. The goal is the refinement of a theory in respect to a surprising or anomalous fact, which is not a prediction nor a logical consequence of its statements. In this methodology, known as Categorical-Generative Analysis (CGA), the grammar is a tool to translate theories into a set of rules of behavior of a relatively stable group of actors within a hypothetical social system. The derivation steps of an instance of any category of social process is still description of the sequence of individual actions and interactions that take place in an empirical setting. The emergent process is analytically decomposable in terms of actions and interactions among elements of the system, but the emergent form is not analytically decomposable in the same way. The researcher can only enumerate and describe the social processes necessary to generate the hypothetical social system.

The CGA method relies on a post-positivist epistemology, Pragmatist Critical Realism (Nellhaus, 1998), which provides its assumptions about meaning, truth, and the nature of reality. These assumptions include: (i) methodological dualism, which requires that the methods used to study human action be distinct from those used in natural sciences; (ii) epistemological relativism, which suggests that truth often derives from conventions and frameworks of assessment that are specific to particular social contexts; (iii) the demarcation criterion that relies on the logic of retroduction; (iv) causal explanation as an unobservable generative mechanism that is activated under a specific configuration of contextual conditions, which results in a socially and historically situated tendency rather than an universal law; and (v) methodological holism, which assumes that an emergent reality exists independently of lower-level entities, relationships, and behaviors.

Categorical-Generative Analysis relies on grammars for modeling and analyzing categories of social processes because grammars represent the discrete-space, discrete-time version of dynamical systems. Just as the design and qualitative assessment of the behavior of these systems is possible in mathematics, CGA seeks to enable a similar assessment for process-like social phenomena. In addition, most of qualitative research methodologies lacks a model and systematic analytical procedures to handle empirical evidence under the assumptions of a post-positivist epistemology. There is also a tendency for rigorous research quality standards like what exists for quantitative methods. Although modern computational linguistics techniques have eliminated the reliance on Chomsky's formalism for grammar specification, CGA is still necessary and further improvements as those detailed in this work will certainly be welcome among its potential practitioners. This is the goal of the present paper.

2 The Problem with Generative Grammars

Generative Grammar Theory, or simply Generativism, is the branch of linguistics concerned with a hypothesized innate grammatical structure, that is, a biological capacity, which is built in the human brain (Everaert et al., 2015). Consequently, the theory suggests there is innate constraints on what grammar that a human language could exhibit, which is the assumption of the existence of a *universal grammar*.

The generative approach to studying of language involves developing a formal grammar, which is a system of grammatical rules that produce valid sentences in the target language. It is an extension of the paradigm of Linguistic Structuralism, but which argues that language is an object of study within a branch of cognitive psychology, such as it is reasoning and problem solving (Chomsky, 2016).

Grammar is a system of rules for producing sentences classified in relation to the computational limits imposed by their production rules on the possible patterns of sequences of symbols in a language. *Chomsky hierarchy* (1956) is a typology or containment arrangement of the classes of grammars (and languages) based on the set of all sentences that each of them can produce (or comprise). This hierarchy defines a containment relation ($\subseteq$) between each pair of classes of grammars. The levels of Chomsky's hierarchy divide the set of all languages into distinct classes based on the computational complexity of their system of production rules. Complexity refers to the available resources required to generate their set of grammatically valid sentences (i.e., in essence, recursion and context-sensitivity), but in terms of computation time and memory storage. Consequently, this hierarchy asserts that regular languages are a subset of context-free languages, but with no pattern based on recursion. Additionally, context-free languages are a subset of context-sensitive.

Despite Formal Language Theory becoming a key branch of computer science, several linguists still reject Generativism for various reasons. Even generative linguists argue that Chomsky's generative grammar formalism has certain inadequacies when it comes to analyzing natural languages. Firstly, terminal and nonterminal symbols are structureless objects, meaning that syntactic relations rely only on the grammar rules. Furthermore, given a fixed set of alphabet symbols, language variation results from grammar variation only. Finally, concatenation over the alphabet set is the unique admissible syntactic operation.

In this way, there is a linguistic technique to eliminate these drawbacks called *lexicalization*, which involves creating a controlled vocabulary in a *lexicalized*, or *type logical grammar*. In these grammars, the *lexicon*, or vocabulary of the language, contains lexical items for words, set phrases, and word patterns, which represent the units of meaning, or *lexemes*. Each syntactic structure regarded as a *type* or *category* has a lexical item in the alphabet set (Σ). This approach has two advantages: (i) most syntactic relations between words derive from the syntactically typed lexical items assigned to them; and (ii) the remaining syntactic relations derive from *type inference rules*, which are language-independent logical operations on types using deductive syllogism only.

In a lexicon, types (or categories) are syntactic structures, representing sets of strings. In turn, terminal symbols become informative syntactic structures mapping to either other terminals or non-terminals as complex types, waiting for evidence of the pattern of syntactic relations that they predict to occur in valid sentences of the language. Last of all, language variation now results from lexicon variation, which is language specific, while the fixed set of type inference rules, which extends the lexicon by assigning types to strings, becomes a kind of universal grammar since it is common to all languages.

In addition, lexicalized grammars are *constituency grammars*: a class of grammar formalisms that rely on a subject-predicate term logic, based on binary division of non-terminals that result in a one-to-one-or-more correspondence between nodes in the derivation tree structure, known as *constituency relation*, such that the constituent structures become phrase-structure rules or rewrite rules. The alternative *dependency relation* is a one-to-one correspondence instead: for every word in a sentence, there is the same kind of node in the syntactic structure, which turns out to be a graph rather than a tree. In generative linguistics, the selection between these two types of grammars relies on technical issues such as efficiency of the syntactic parsing algorithm, ability to handle ambiguity, and the benefit of visualizing the syntactical structure of the sentence in the case of constituency grammars.

In linguistics, the derivation path is not the recipe to construct a valid sentence, but the proof steps of its membership to the target language. In this way, the derivation path for constituency grammars and dependency grammars, and even for distinct formalisms based upon the constituency relation, is different. In Categorical-Generative Analysis, there is another reason to choose constituency grammars, which is the assumption that the grammar effectively describes the process by which the actors interact with the social system. In Chomsky Normal Form, rules always have one symbol in the head and two in the body (i.e., $A \rightarrow B, C$) because the type of process or event in the left side (B) causally precedes the other on the right side (C). Therefore, the full realization of the non-terminal B in terminal symbols must precede the realization of C. Since each event is a consequence of one or more alternative configurations of past event outcomes, the parsing algorithm uses leftmost derivation only.

This paper proposes the same evolutionary trajectory for Categorical-Generative Analysis: to substitute the Chomsky's generative grammar formalism with a lexicalized grammar. Because both them rely on the constituency relation, the assumption that the metaphysical properties of path dependency and contingence are equivalent to the computational properties of recursion and context-sensitivity, which is analogous to the equivalence between the categories of social processes and formal languages, holds. The next sections present two lexicalized grammars, pregroup grammars and categorial grammars, which are the formalisms that can provide the lexicalization of the categories of social processes as required.

3 Pregroup Grammars

Pregroup grammar (PG) is a language formalism that is in the class of type logical grammars (Lambek, 1999). Precisely, it consists of a set of words L, a set of *basic* or *atomic* types T, the free pregroup P(B) that is generated by T, and a *dictionary* relation $\vdash : L \rightarrow T$ that relates each word to one type.

A type can be (i) *atomic* or *basic*, such as sentence (s) and noun (n); (ii) *simple*, which are iterated adjoints of basic types (e.g., n^l, n^r, n^{ll}, n^{rr}), and (iii) *composite*, which is a composition of the basic and simple types using the composite operation (.) on the set of simple types. Therefore, word patterns become compositions of grammatical functions, in which left-adjoint simple types represent symbols that must precede the denoted category, right-adjoints must succeed it, and there is always one basic type, which results from the function call.

If P is the set of simple types, then the set of all types T(P) satisfies (i) $P \subset T(P)$; (ii) if $\alpha \in T(P)$ and $\beta \in T(P)$ then $\alpha . \beta^l \in T(P)$; and (iii) $\alpha . \beta^r \in T(P)$. Due to this definition, there is a kind of *type hierarchy*: a type γ is a subtype of φ if and only if γ occurs within φ.

In the English language, the basic types are nouns and sentences, while the composite types are articles, prepositions and verbs. Complex types refer to positions in the structure of the sentence where other complex types and basic types must be located. Consider transitive verbs ($n^r.s.n^l$), which require a noun phrase to the right (n^r) as the subject and other noun phrase to the left (n^l) as the predicate, returning a sentence (s); it differs from intransitive verbs ($n^r.s$) that require no predicate. In its turn, articles ($n.n^l$) require a noun to the left (n^l) to return a noun phrase (n). This way the complex types refer one to each other. For example, in the sentence "cats eat mice", "cats", "mice" $\vdash$ n are the basic types, and "eat" $\vdash n^r.s.n^l$ is the composite type, which derives to "eat mice" $\vdash n^r.s$ (or to "cats eat" $\vdash s.n^l$), and then to "cats eat mice" $\vdash$ s. In the case of the sentence "the cats eat mice", "the" $\vdash n.n^l$ such that the reduction "the cats" $\vdash (n.n^l).(n)$ $\vdash$ n must take place at the derivation path before the second reduction with "eat mice" $\vdash n^r.s$ completes the procedure, resulting in s.

3.1 A Brief Review of Group Theory

The mathematical foundation for the Pregroup Grammar formalism is Group Theory, which studies the algebraic structures known as groups. Many mathematical structures (e.g., cryptographic systems, grammars, vector spaces) and physical systems (e.g., molecular symmetry and the standard model of particle physics) regarded as groups equipped with additional operations and axioms. This section introduces some necessary definitions to explain the concept of pregroup.

Definition 3.1.1. A *preorder* or *quasiorder* (S, *R*) is a binary relation *R* (eventually →, or ≤) on the set S, which is (i) *reflexive*, that is, it relates every element a in S to itself, such that a *R* a; and (ii) *transitive*, that is, for all elements a, b, c in S, whenever a *R* b and b *R* c hold, a *R* c also holds.

In Category Theory, the symbol → means an arbitrary binary relation, while ≤ use to mean an ordering relation. Precisely, a *partially ordered set* or *poset* (S, *) is a partial order relation ≤ on the set S that denotes the sequential arrangement of the elements of S, in which for some pairs of elements in S, one of the elements precedes (or succeeds) the other. It is reflexive, transitive, and antisymmetric. For example, the power set (S^*) of any given set is a poset.

In its turn, a *totally ordered set* (ℜ, ≤) is still a partial order relation that holds for every pair of elements in S, but the converse is not always true. For example, the relation "less than or equal to" (≤) on the set of real numbers ℜ is a total order relation.

Definition 3.1.2. A *group* is any tuple G = (S, →, •, 1), or simply (S, •), in which S is a set; • is a binary operation on S, that is, • : S × S → S, but satisfying three axioms: (i) *associativity*, that is, for all a, b, c in S, (a • b) • c = a • (b • c) holds; (ii) *identity element*, that is, there exists an element 1 in S such that for every element a in S, a • 1 = 1 • a = a holds; and (iii) every element a in S has an *inverse*.

A *monoid* or *semigroup* is a group M with no inverse. In other words, it is a set equipped with only an associative binary operation and identity element. In Category Theory, monoid is a category with a single object such that all morphisms depart from this single object to itself. For example, a *free monoid* is the monoid (Σ^*, •), in which Σ^* is the set of all finite sequences of symbols from the alphabet set Σ (strings), including the empty string ε; and • is the concatenation operation on S, that is • :

$\Sigma^* \times \Sigma^* \rightarrow \Sigma^*$. Usually, concatenation may also use the symbols point (.), comma (,), or no symbol between the pair of terms in Σ^* at all.

In its turn, a *preordered monoid* is a preorder relation (S, ≤) along with a monoid (S, ≤, •), such that the binary operation • satisfies the axiom of *monotony* (i.e., given a, b, c, d in S, a ≤ c and b ≤ d imply a • b ≤ c • d), or equivalently the axiom of *substitution* (i.e., b ≤ d imply a • b • c ≤ a • d • c).

Finally, a *pomonoid* is a preordered monoid (S, ≤, •) with a partially ordered set (S, ≤), such that given a, b in S, a ≤ b ≤ a implies a = b, that is, the relation ≤ is *antisymmetric*.

Definition 3.1.3. A *pregroup* is a tuple P = (S, ≤, •, *l*, *r*, 1), in which the relation ≤ satisfies two axioms: (i) if a ≤ b then c • a ≤ c • b and a • c ≤ b • c; and (ii) *l* / *r* are unary operations called left and right *adjoints*.

Notice that (S, ≤) is a *preorder*, (S, ≤, •) is a *monoid*, and (S, ≤, •, 1) is a *partially-ordered monoid*. In addition, any pregroup is a *pomonoid*, such that every object a in S has left and right adjoints, satisfying four axioms known as the *Ajdukiewicz laws*: (i) and (ii) $a^l \bullet a \leq 1 \leq a \bullet a^l$ (i.e., contraction and expansion to the left); and (iii) and (iv) $a \bullet a^r \leq 1 \leq a^r \bullet a$ (i.e., contraction and expansion to the right).

Finally, a ≤ b if and only if $b^l \leq a^l$ if and only if $b^r \leq a^r$. The following equalities also hold in every pregroup: $1^l = 1^r = 1$; $a^{lr} = a^{rl} = a$; $(a \bullet b)^l = b^l \bullet a^l$; $(a \bullet b)^r = b^r \bullet a^r$.

Given a sentence type s in T and the types for a sequence of words t_1, …, t_n in T, the derivation procedure in pregroup grammars demonstrates $t_1, \ldots, t_n \rightarrow s$. The *switching lemma* (Lambek, 1999) makes the parsing problem for pregroups decidable: for any pair of types t, t' ∈ T, if t → t' then there is a type t"∈ T such that t → t" without expansions and t" → t' without contractions.

3.2 The Problems with Pregroup Grammars

Between the context-free and context-sensitive classes of languages, there is infinite number of computational complexity levels of particular interest: the patterns found in natural languages that are not regular nor context-free result from the so-called *mildly context-sensitive grammars*, which are often efficiently parsable and inferable from positive evidence (Oates et al., 2006).

Definition 3.2.1 Any class of languages is mildly context-sensitive if and only if (Joshi, 1985): (i) it includes the class of all ε-free context-free languages, where ε is the empty string; (ii) all its languages are *semi-linear*, which means that they are of constant growth; (iii) it contains the languages in which there are sentences exhibiting the *multiple agreement* pattern, which is the set $\{a^n b^n c^n : a, b, c \in \Sigma^+ , n \geq 1\}$, the *cross-serial dependency* pattern, which is the set $\{a^m b^n c^m d^n : a, b, c, d \in \Sigma^+ , m, n \geq 1\}$, and the *duplication* pattern, which is the set $\{ww : w \in \Sigma^+\}$; and (iv) the membership problem for any sentence is decidable in polynomial time.

There are many alternative grammar formalisms to describe languages at each mildly context-sensitive level. Understanding of the constraints on the production rules that generates these patterns and prooving their expressive power are central problems in Formal Language Theory. Two grammars have *weak equivalence* between themselves if and only if both generate the same set of valid sentences, that is, the formal language they generate is the same. Similarly, two grammars has *strong (or structural) equivalence* between themselves if and only if both generate the same set of derivation trees, which are abstract syntactic objects representing the sequence of syllogistic proof steps necessary to verify if a given sentence belongs to a specific language.

Pregroup grammars are weakly equivalent to context-free grammars (Buszkowski, 2001), such that they do not have enough expressive power to parse natural languages. Although there are extensions to this class of grammars that are weakly equivalent to some mildly context-sensitive grammar (Genkin et al., 2010; Kobele & Kracht, 2005), they do not have a specific function type, and instead make use of inverse types combined with its monoidal operation. Consequently, pregroup grammars cannot use neither lambda-calculus nor function denotations to assign semantics, making this task quite complicated. Providentially, another weakly equivalent class of lexicalized grammar formalisms seems to be more suitable to describe mildly context-sensitive languages: Combinatory Categorial Grammar.

4 Categorial Grammars

Categorial Grammar (CG) is a language formalism that belongs to the class of type logical grammars, establishing an interface between surface syntax and underlying semantic representation. Also called AB-grammar

(Ajdukiewicz, 1935; Bar-Hillel, 1953; Lambek, 1958), it is a lexicalized grammar with two type inference rules, making it weakly equivalent to the class of Context-Free Grammar. Given a sequence of words with the syntactic types assigned to them, the type returned after the derivation procedure, consisting of a sequence of syllogistic proof steps, is the non-terminal symbol for the valid sentence (S).

In its turn, Combinatory Categorial Grammar (CCG) is an extension to CG (Steedman, 1987, 1996) with a slightly larger set of type inference rules borrowed from combinatory logic, which is weakly equivalent to the mildly context-sensitive class of grammars known as Linear Indexed Grammar (Vijay-Shanker & Weir, 1994).

A type can be either *primitive* or *complex*, which is a combination of the forms $\alpha\backslash\beta$ or α/β, where α is the resulting type and β is the type of the argument taking place either to the left or to the right, respectively. Thus, complex types are *functors* that take a type β as an argument, specified in the right side of the positioning symbol (either "\" or "/"), and return other type α, specified on the left side. Both the categories $\alpha\backslash\beta$ and α/β map β into α, but the former maps β to the left ($\beta^l \rightarrow \alpha$) and the latter maps β to the right ($\beta^r \rightarrow \alpha$).

When used for modeling languages, each type reflects a grammatical function, which results in a type (α) if it has as argument another type located to either the left side ($\alpha\backslash\beta$) or the right side (α/β) of the lexical item in the respective grammatical structure. Finally, there are operations for combining syntactic structures that derive new types, which work as type inference rules.

For the English language, primitive types are sentence (S), noun (N), and noun phrase (NP), whereas complex types represent types of verbs and other grammatical functions. For example, in some valid sentences (S), there is a transitive verb (V), which is a type for the syntactic structure with one noun phrase in its left side (NP^l) as the subject, and other noun phrase in its right side (NP^r) as the predicate – in the Pregroup Grammar notation, it is $v : np^l \,.\, np^r \rightarrow s$, while in CCG notation (i.e., $\alpha\backslash\beta/\beta$), it is $S\backslash NP^l/NP^r$. Consequently, the complex type for transitive verbs is (S\NP)/NP, which means that a verb of this kind forms a valid sentence (S) if and only if a noun phrase precedes it (NP^l), and another noun phrase (NP^r) follows it. In other words, a transitive verb is like a function that takes two instances of the same type as arguments (NP^l and NP^r) and returns the type of sentence S. For example, in "cats eat mice", there are "eat" $\vdash$ (S\NP)/NP and "cats", "mice" $\vdash$ NP, which derive to "eat mice" $\vdash$ (S\NP) and "cats eat mice" $\vdash$ S.

4.1 Combinatory Logic

In logic, a variable, which is a symbol that represents a value, can be classified as bounded or free depending on whether it is bound by a quantifier, such as "for all" (∀) or "there exists" (∃), or is not bound by any quantifier.

Combinatory logic (Curry, 1930; Schonfinkel, 1924) is a formal logic notation to write formulas without bounded variables by means of using such a limited set of primitive functions with no free variables. Known as *combinators* (Curry & Feys, 1958), these primitive functions with no free variables are higher-order functions that use function application and other previously defined combinators to define a result from its arguments. In this sense, combinators replace variable binding term operators and eliminates the need for quantifiers – denoted by the symbols ∀ ("for all") and ∃ ("there is at least one") –, which is an alternative solution to the so-called *problem of substitution*. For example, given a relation $\forall x_1, \ldots, x_n\ R(x_1, \ldots, x_n)$ with no free variable, the substitution of the bounded variables $x_1, \ldots, x_n$ for the terms $t_1, \ldots, t_n$ results in another term $R(t_1, \ldots, t_n)$.

Since Combinatory logic (CL) can specify recursive functions, which are computable functions, it is a model of computation like Lambda Calculus (Church, 1932) and Turing machines (Turing, 1936). In fact, it relies upon a pair of key notions from Lambda Calculus to provide a set of combinators: abstraction and application.

Define *abstraction* to be such a term of the form $\lambda v.E_i$, where v is a variable known as the *formal parameter* of the abstraction, and E_i is the *body* of the abstraction. This term symbolizes a function applied to an argument, binding the formal parameter v to this argument, and returning E_i with every occurrence of v replaced by the argument.

Define now *application* to be other term of the form $(E_i\ E_j)$, which is the execution of the function E_i with E_j as its argument, where E_i is called *applicand*, and E_j is the argument to replace all occurrences of the formal parameter v in the body of the applicand. The result is a new term that is equivalent to the old one. In other words, given the abstraction $\lambda v.E_i$, the application $((\lambda v.E_i)\ a)$ is the same making v := a in all occurrences of v in E_i. For example, for any application, the identity combinator is (I x) = x.

Since substitution is a critical operation in any formal system that uses bounded variables, such as first-order logic and other high-order logics, CL imitates the λ abstraction although it does not offer a variable binding operator. Accurately, CL is a term rewriting system. A combinatory term is like a lambda term, but primitive functions are combinators, that is, functions with no free variables. Each combinator has a reduction rule like (P $_{x1}$, …, x_n) = E.

4.2 Combinators

During the derivation procedure for a specific sentence, complex types result from the application of one *combinator* to an instance of the type provided as the argument. The differences between the classes of categorical grammars are the set of combinators and the computational complexity level of the class of languages that these combinators can generate.

Categorial grammars (Ajdukiewicz, 1935; Bar-Hillel, 1953; Lambek, 1958) make use of a pair of combinators: *forward application*, which is the logical operation (i) $> : \alpha/\beta , \beta \Rightarrow \alpha$, that is, $\alpha/\beta : f , \beta : x \Rightarrow \alpha : (f\,x)$; and *backward application*, which is (ii) $< : \beta , \alpha\backslash\beta \Rightarrow \alpha$, that is, $\beta : x , \alpha\backslash\beta : f , \Rightarrow \alpha : (f\,x)$. Both work as type inference rules.

Combinatory categorial grammars (Steedman, 1987, 1996) expand the CG's set of pairs of combinators by adding: two *function composition* combinators, (iii) $B> : \alpha/\beta , \beta/\gamma \Rightarrow \alpha/\gamma$, that is, $\alpha/\beta : f , \alpha/\gamma : g \Rightarrow \alpha/\gamma : \lambda x.f(g\,x)$, and (iv) $<B : \beta\backslash\gamma , \alpha\backslash\beta \Rightarrow \alpha\backslash\gamma$, that is, $\alpha\backslash\beta : f , \beta\backslash\gamma : g \Rightarrow \alpha/\gamma : \lambda x.g(f\,x)$; and two *type-raising* combinators (v) $T> : \alpha \Rightarrow T/(T\backslash\alpha)$, that is, $\alpha : x \Rightarrow T/(T\backslash\alpha) : \lambda f.(f\,x)$, and (vi) $<T : \alpha \Rightarrow T\backslash(T/\alpha)$, that is, $\alpha : x \Rightarrow T\backslash(T/\alpha) : \lambda f.(f\,x)$. Observe that the type-raising rules turn arguments into functions over functions over these arguments, which let arguments to compose.

Finally, many authors extended the classic version of CCG with new combinators, improving the expressive power of this of class of mildly context-sensitive formalisms while isolating all cross-linguistic variation in the lexicon and managing a universal set of inference rules (Kuhlmann et al., 2015; Wood, 1993). It is the case of an extended CCG for capturing long-range dependencies (Steedman, 2000), which enlarges the CCG's set of combinators with: two *substitution* combinators, (vii) $S> : (\alpha/\beta)/\gamma$

, $\beta/\gamma \Rightarrow \alpha/\gamma$ and (viii) $<S : \beta\backslash\gamma , (\alpha\backslash\beta)\backslash\gamma \Rightarrow \alpha\backslash\gamma$; two *cross-substitution* combinators, (ix) $Sx> : (\alpha/\beta)\backslash\gamma , \beta\backslash\gamma \Rightarrow \alpha\backslash\gamma$ and (x) $<Sx : \beta/\gamma , (\alpha\backslash\beta)/\gamma \Rightarrow \alpha/\gamma$; and one *coordination* combinator, (xi) $<\&> : \alpha , CONJ , \alpha \Rightarrow \alpha$, in which CONJ is a simple type, or even the empty string ε. For the purpose of this work, this is the required set of combinators (Fig. 1).

```
(i)       > : α/β , β ⇒ α
(ii)      < : β , α\β ⇒ α
(iii)     B> : α/β , β/γ ⇒ α/γ
(iv)      <B : β\γ , α\β ⇒ α\γ
(v)       T> : α ⇒ T/(T\α)
(vi)      <T : α ⇒ T\(T/α)
(vii)     S> : (α/β)/γ , β/γ ⇒ α/γ
(viii)    <S : β\γ , (α\β)\γ ⇒ α\γ
(ix)      Sx> : (α/β)\γ , β\γ ⇒ α\γ
(x)       <Sx : β/γ , (α\β)/γ ⇒ α/γ
(xi)      <&> : α , CONJ , α ⇒ α
```

Fig. 1. A set of combinators for combinatory categorial grammars (Steedman, 2000).

5 Example: Theory of The Firm

Neoclassical economics, which is the mainstream school of economic thought, asserts that the value of an economic good is the result of the maximization of both utility by consumers with income constraints and profits by firms with budget and information constraints. In this sense, the neoclassical theory of the firm explains the existence, behavior and structure of economic organizations by their ability to make decisions to maximize the difference between revenue and costs. It is an alternative social system to the market-price mechanism whenever is more efficient to produce under a bureaucracy. The firm’s behavior becomes manifest through decisions about what to produce and where to allocate capital, which depends on how profits increase thereafter.

However, since firms often do not behave like the neoclassical prediction, economists and other social researchers created extensions to the theory of the firm in order to make it adequate to changing economic and market structures as well as specific empirical settings. Consequently, there exists concurrent grand theories and middle-range theories to explain and predict the behavior of the firm.

Contemporary theories of the firm consider profit maximization such a short-run goal, while acknowledging that the firms also exist to pursue long-run goals, such as growth and sustainability. The first alternative, the contract-based view of the firm, which relies upon transaction costs, has its roots in the work of Ronald Coase (1937), but it incorporates many neoclassical assumptions yet. Next, a number of theories of the firm arose with the focus on specific issues, such as the managerial view (Baumol, 1959) in the principal-agent relationship, the behavioral view (Cyert & March, 1963) in the criticism to neoclassical assumptions of profit maximization and perfect information, and the resource-based view (Penrose, 1959) in idiosyncratic assets like productive knowledge and inter-organizational relationships.

This paper is part of a research project that aims to develop a meta-theoretical framework for explaining forms of economic coordination using a computational complexity approach (Braga, 2017b). Precisely, the proposed methodology involves using grammars for analyzing social processes that occur within a group of economic agents that maintain a stable membership configuration over time. The goal is to distinguish between the market and other forms of economic coordination, including the firm, based on the complexity of the causal relations resulting from the social structures operating in specific empirical settings.

5.1 The Assumption of Equivalence to Linguistic Structures

Consider some structural constraints from the market model of perfect competition: (a) a large number of buyers and sellers; (b) the product is homogeneous; (c) every participant is a price taker; (d) all participants are rational, which means that trades occur to maximize their economic utilities; (e) there are no barriers to entry or exit of market participants; (f) factors of production have perfect mobility; (g) consumers and producers have perfect knowledge, such that they always known the price and the utilities they get from the product; (h) there is no transaction costs. In this canonical model, interactions between buyers and sellers randomly generate a type of market event, which is *trade*, in such a way that each event outcome is independent from the others and the perfect competition process is stochastic in nature.

Consider now that the market model has non-zero transaction costs. In this case, the existence of the firm is possible, but the limits of the firm, which is a configuration of internalized and externalized economic activities, may change over time in terms of the relationship between administrative costs and transaction costs. This decision-making process is also stochastic in nature, so there is no difference between firms and markets in this sense.

Given the assumption that both firms and markets are social systems to establish causal relations between their decision-making events over time, they clearly do so in distinct ways. Markets, under their canonical structural constraints, always randomly generate events. In contrast, firms and non-market forms of economic coordination do not exhibit the same structural constraints of markets. Instead, they often generate patterns of sequences of types of either individual action or interaction in a computational complexity level that markets can never achieve due to the properties of path dependence and contingency from many of their outcomes.

Consequently, under the equivalence between coordination structures and linguistic structures, patterns of sequences of decision-making event outcomes exhibit the same properties of sentences of a formal language, which establishes computational complexity as a criterion to distinguish the categories of social processes that occur in coordination structures of all types from those that cannot occur in markets. The market structures are weakly equivalent to regular grammars, while firms and the non-market coordination structures are weakly equivalent to mildly context-sensitive grammars.

The reason for using a "mildly" class instead of the context-sensitive class is due to the pragmatist assumption that no causal claim relies on emergent forms or processes, which is also the statement that any system is a causal consequence of another distinct system only. On the contrary, *downward causation* is the assumption that higher levels of emergent phenomena can causally influence their lower levels' developments, which is a kind of radical holism.

The derivation of context-sensitive rules in Penttonen Normal Form, which is A, B → A, D, implies that a process A "causes" the substitution of a process B for D, even though A is not fully realized in terms of terminals for event outcomes yet. This is the same as accepting the causal mechanism of downward causation. The arrow of time must always go in the direction of future, such that only unfolded event outcomes can have a causal influence on events yet to happen.

The goal is the holistic understanding of an emergent form or entity by describing each category of its constituent processes, in terms of the underlying causal mechanism, not the contrary. Consequently, there is a constraint on the computational complexity level in CGA, which is the level of mildly context-sensitive grammars.

5.2 The Hypothesis of Systemic Competence Development

Consider now the hypothesis of systemic competence development at the firm (Braga, 2017b, 2020), which is the first work elaborated relying upon Categorical-Generative Analysis. This section re-elaborates that multiple case study but substitutes the Chomsky formalism with a combinatory categorial grammar instead. In other words, it is the same theoretical model, the same set of empirical evidence from two pairs of cases, and the same CGA analytical procedure, which results in the same conclusions. The difference is the grammar formalism in use: the former is a top-down, iterative approach for parsing sequences of symbols, while the latter was used in a bottom-up, declarative approach.

The original CGA for the social process of competence development at the firm begins with a grammar (Fig. 2) in the Chomsky Normal Form for this category of process (S). It consists of the process of *relationships with partners* (RR) followed by *generation of capabilities and economic goods* (GG). The first social process involves a sequence of outcomes of either *combination of idiosyncratic resources* ({c}) or *information and knowledge exchange* ({i}), while the resulting event outcome is a single *economic good generation* ({g}) instance.

```
S  → GG, CD
CD → GG, CD
   | GG
GG → RR, G
RR → R, RR
   | R
R  → {c} | {i}
G  → {g}
```

Fig. 2. A constituency grammar for the competence development process.

The hypothesis of the possibility of systemic competence development (H1) means mapping each type X into X[..], where the square brackets contain a configuration of contextual conditions allowing the surprising fact: the *systemic good generation* ({g'}) to satisfy systemic needs rather than market needs. First, this fact becomes a new event outcome for the type *generation of economic good* (G). Next, the retroductive procedure consists of hypothesizing knowledge exchange relationships that are not available to the rivals of the firm ({i'}) as a condition for the creation of a systemic economic good ({g'}), which in turn enables the creation of one or more economic goods ({g}) to the market in a future system state. In addition, the hypothesized exogenous contextual conditions are (i) the high impact of the systemic problem in the firm's performance and (ii) the appropriation by the firm of a part of the economic value created to the socioeconomic system in the form of a new source of rent.

The goal is not predicting when (or if) a surprising fact is going to take place, but tracing the decision-making process to identify the patterns of sequences of event outcomes and configurations of contextual conditions enabling the occurrence of the surprising fact, which then becomes a new event outcome. In the same social system, the hypothesized generative mechanism occurs repeatedly under the same configuration of contextual conditions – i.e., initial exogenous conditions and patterns of sequences of unexpected past outcomes –, which characterizes an extension of the category of the social process under inquiry.

The research may identify the configurations of contextual conditions using the Quine-McCluskey algorithm (McCluskey, 1959; Quine, 1952) because they are set-theoretic relations between the sequence of past event outcomes and the present state transition. Next, it comprises the substitution of the resulting alternative grammar rules (e.g., producing g and g') for strictly context-sensitive rules (i.e., which are non-context-free). Consequently, this step eliminates the ambiguity introduced by the acknowledgement of the surprising fact as a new event outcome.

After identifying the configurations of contextual conditions related to the instances of the surprising fact within each case study, the next step of CGA consists of mapping the initial context-free grammar (Fig. 2) into a new mildly context-sensitive grammar using the Indexed Grammar formalism (Aho, 1968). The problem of using this formalism is that each expected configuration of contextual conditions in a social process must be either copied or translated into another configuration – i.e., $X[\hat{}\varphi..] \rightarrow Y_{(L)}[\hat{}\varphi..], Y_{(R)}[\hat{}\varphi..]$ or $X[\hat{}\varphi..] \rightarrow Y_{(L)}[\hat{}\lambda..], Y_{(R)}[\hat{}\lambda..]$, respectively – to each of its sub-processes until the occurrence of all the expected event outcomes. This means the need to change both the rule set (i.e., the graph of the category of process) and the alphabet set (i.e., the set of all event outcomes). In the case of substituting the Indexed Grammar formalism for the Combinatory Categorial Grammar formalism, there is a need to change the lexicon only, which becomes such an alphabet set in which each possible event outcome is equipped with additional structure.

Using the CCG formalism, the lexicon has event types for *economic good generation* ({g} ├ S\RR), *knowledge exchange relationship* ({i} ├ RR), and *combination of idiosyncratic resources* ({c} ├ RR\RR). The economic good (g) returns a valid sentence (S) if and only if preceded by an instance of the process of relationships with partners (RR), which is a sequence of zero or more event outcomes for combination of resources (RR\RR) preceded by one or more outcomes for the knowledge exchange relationship (RR). These relations between event types follow the assumption A1 that instances of the type of event for economic good generation (g) are independent of each other. For example, the derivation path for the instance "i, c, g, i, i, g" is:

i , c , g ├ RR , RR\RR , S\RR $\Rightarrow_{(<)}$ RR , S\RR $\Rightarrow_{(<)}$ S

i , i , g ├ RR , RR , S\RR $\Rightarrow_{(<\&>)}$ RR , S\RR $\Rightarrow_{(<)}$ S

i, c, g , i, i, g ├ S , S $\Rightarrow_{(<\&>)}$ S

Consider now the assumption A2 that the instances of the event type for economic good generation (g) are part of an evolutionary path with other instances of the same event type, such that there is interdependence between them (i.e., there is g_i that extends g_{i-1}). Using same example, {g} ├ S\RR becomes {g} ├ (S\S)\RR, such that the new derivation path is:

i, c, g_{i-1} , i, i, g_i ├ $S_{i-1}\backslash S_{i-2}$, $S_i\backslash S_{i-1}$ $\Rightarrow_{(<B)}$ $S_i\backslash S_{i-2}$

The type "S\S" is absent in the lexicon, such that there is the need for the application of another combinator, which will apply to the event type for the initial exogenous contextual conditions (S). Consequently, the list of combinators in this derivation procedure – i.e., (<), (<), (<&>), (<B), and finally (<) for S, S\S – are the proof steps for the statement that this is an instance of the category of competence development (S).

Consider now the hypothesis H1 that implies the new event outcome, a systemic economic good (g'), which comes to solve a systemic problem hindering the generation of another economic good (g) in the future, but under the assumption A1. In the same example, the substitution of {g} ├ S\RR for both {g'} ├ (S/S')\RR' and {ḡ'} ├ S'\RR still implies that economic goods are either systemic (g'), or non-systemic ($\bar{g}$'), such that the derivation path for the instance "i, c, g', i, i, $\bar{g}$'" is now:

i', c, g', i, i, $\bar{g}$' $\vdash$ S/S' , S' $\Rightarrow_{(>)}$ S

Finally, the same hypothesis H1, but under the assumption A2. In the example, the substitution of {g} $\vdash$ (S\S)\RR for {g'} $\vdash$ ((S/S')\S)\RR' and {$\bar{g}$'} $\vdash$ (S'\S)\RR is such that:

i', c, g', i, i, $\bar{g}$' $\vdash$ (S/S')\S , S'\S $\Rightarrow_{(Sx>)}$ S\S

Or in the case that there are a previous economic good to be extended (g_{i-1} $\vdash$ $S_{i-1}\backslash S_{i-2}$):

i', c, g_i', i, i, $\bar{g}_i$' $\vdash$ (S_i/S'$_i$)\$S_{i-1}$, S_i'\$S_{i-1}$ $\Rightarrow_{(Sx>)}$ $S_i\backslash S_{i-1}$

The examples above comprise a single instance of the social process of competence development. However, in real-life situations, there may be many concurrent instances of the same category of process, or even distinct categories of processes taking place within a firm through the actions of a community of economic agents that remains relatively stable over time, such that the process instance's procedural memory holds information about its execution. In this case, unlike the strictly sequential derivation procedure of categorial grammars, the left and right types of events may not be immediately adjacent, but rather causally antecedent or consequent types of events. The numbers in the types (e.g., S_i) are important for making causal links between pairs of instances, even though some instances may not have an index number, and thus may take part in more than one instance (e.g., the S for the initial exogenous contextual conditions).

6 Conclusions

This paper is another chapter of the Categorical-Generative research project, which proposes an analytic framework for formulating middle-range theories based on the ontological and epistemological assumptions of Pragmatist Critical Realism. The analytical procedure starts with the apprehension of a surprising or anomalous fact and elaboration of an informed guess, or explanatory hunch. Using the grammar model, this hunch is then deliberately and recursively taken backward in the given sequence of event outcomes, for analysis and adjustment of the grammar, resulting in a hypothesis worthy of empirical testing. This methodology is capable of tackling any concrete category of complex, dynamic, and contingent social process as a pattern of decision-making events about individual action or interaction under the influence of the hypothesized structures. The result is a plausible theoretical explanation for a historical phenomenon that has taken place in a specific empirical setting.

The assumption that social systems are computational devices means that their constituent social processes are equivalent to Turing machines or partial recursive functions. Social systems emerge from concurrent complex, dynamic, and contingent social processes that occur in specific empirical settings. However, modeling social phenomena as an emergent result from the collective behavior of agents still entails interpreting how the properties of this mathematical model relate to the assumptions of the ontology in use. The acknowledgement of contradictions with the mainstream theory in the form of contingent developments also explains the heterogeneity of social phenomena observed by coexisting middle-range theories.

The first challenge of this methodology was the proposition of a meta-theoretical framework to explain the structures of economic coordination between markets and hierarchies from a computational complexity point of view. The ontological assumption of the equivalence of the structures of economic coordination with linguistic structures enables the analysis of sequences of decision-making event outcomes as chains of symbols belonging to a formal language. There is empirical evidence supporting the hypothesis of systemic competence development, which is a pattern of socioeconomic behavior that cannot occur in the market structure because of the constraint on the computational complexity level of its decision rules.

The main result is the proposition of a theorem: the firm and all non-market forms of economic coordination exist to generate categories of social processes exhibiting a pattern of causal relations between their event outcomes that the market can never support due to the structural constraints from the assumption of perfect competition. The systemic competence development process is an example of this kind of social phenomenon that cannot occur in market structures.

CGA provides a way to discover categories of social processes within real social systems according to the assumptions of Pragmatist Critical

Realism, which allows scientific inferences that are not in the scope of the social positivist, normal science. The present work finalizes the CGA methodology with a kind of algebra based on an alternative grammar formalism, which achieves the same conclusions as the first empirical research but in a more precise and intelligible way.

References

Aho, A. V. (1968). Indexed Grammars---An Extension of Context-Free Grammars. *Journal of the ACM*, *15*(4), 647–671. https://doi.org/10.1145/321479.321488

Ajdukiewicz, K. (1935). Die syntaktische Konnexität. *Studio Philosophica*, *1*, 1–27.

Bar-Hillel, Y. (1953). A Quasi-Arithmetical Notation for Syntactic Description. *Language*, *29*, 47–58.

Baumol, W. J. (1959). *Business behavior, value and growth*. Macmillan.

Braga, B. da R. (2017a). A Categorical-Generative Theory of Social Processes: About Describing a Complex, Dynamic and Contingent Pattern of Decision-Making Events. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3033880

Braga, B. da R. (2017b). Unveiling the Systemic Nature of the Firm Using a Grammar of Social Processes. *SSRN Electronic Journal*, *December*. https://doi.org/10.2139/ssrn.3105701

Braga, B. da R. (2020). A Grammar of the Distinctive Competence Development at the Firm for the Solution of Systemic Problems. *Journal of Innovation Management*, *8*(2), 87–118. https://doi.org/10.24840/2183-0606_008.002_0007

Buszkowski, W. (2001). Lambek grammars based on pregroups. In P. D. Groote, G. Morrill, & C. Retor (Eds.), *LACL 2001. LNCS (LNAI)* (vol. 2099, pp. 95–109). Springer.

Chomsky, N. (1956). Three models for the description of language. *IEEE Transactions on Information Theory*, *2*(3), 113–124. https://doi.org/10.1109/TIT.1956.1056813

Chomsky, N. (1957). *Syntactic Structures*. Mouton.

Chomsky, N. (2016). The language capacity: architecture and evolution. *Psychonomic Bulletin & Review*, *24*(1), 200–203. https://doi.org/10.3758/s13423-016-1078-6

Church, A. (1932). A Set of Postulates for the Foundation of Logic. *The*

Annals of Mathematics, *33*(2), 346. https://doi.org/10.2307/1968337

Coase, R. H. (1937). The Nature of the Firm. *Economica*, *4*(16), 386–405. https://doi.org/10.2307/2626876

Crawford, S. E. S. S. ., & Ostrom, E. (1995). A Grammar of Institutions. *The American Political Science Review*, *89*(3), 582–600. http://www.jstor.org/stable/2082975

Curry, H. B. (1930). Grundlagen der Kombinatorischen Logik. *American Journal of Mathematics*, *52*(3), 509. https://doi.org/10.2307/2370619

Curry, H. B., & Feys, R. (1958). *Combinatory logic*. North-Holland.

Cyert, R., & March, J. G. (1963). *A Behavioral Theory of the Firm* (2nd ed.). Willey-Blackwell.

Epstein, J. M. (2006). Agent-based computational models and generative social science. In *Generative Social Science: Studies in Agent-Based …* (Vol. 4, Issue 5, pp. 41–60). Princeton University Press. https://doi.org/10.1002/(SICI)1099-0526(199905/06)4:5<41::AID-CPLX9>3.0.CO;2-F

Everaert, M. B. H., Huybregts, M. A. C., Chomsky, N., Berwick, R. C., & Bolhuis, J. J. (2015). Structures, Not Strings: Linguistics as Part of the Cognitive Sciences. *Trends in Cognitive Sciences*, *19*(12), 729–743. https://doi.org/10.1016/j.tics.2015.09.008

Genkin, D., Francez, N., & Kaminski, M. (2010). Mildly context-sensitive languages via buffer augmented pregroup grammars. *Lecture Notes in Computer Science*, *6200 LNCS*, 144–166. https://doi.org/10.1007/978-3-642-13754-9_7

Jerne, N. K. (1985). The generative grammar of the immune system. *The EMBO Journal*, *4*(4), 847–852. https://doi.org/10.1002/j.1460-2075.1985.tb03709.x

Joshi, A. K. (1985). Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? In D. R. Dowty, L. Karttunen, & A. M. Zwicky (Eds.), *Natural language parsing* (pp. 206–250). Cambridge University Press. https://doi.org/10.1017/CBO9780511597855.007

Kobele, G. M., & Kracht, M. (2005). *On Pregroups, Freedom, and (Virtual) Conceptual Necessity* (No. 1; Working Papers in Linguistics).

Kuhlmann, M., Koller, A., & Satta, G. (2015). Lexicalization and Generative Power in CCG. *Computational Linguistics*, *41*(2).

https://doi.org/doi:10.1162/COLI a 00219
Lambek, J. (1958). The mathematics of sentence structure. *American Mathematical Monthly*, *65*, 154–170.
Lambek, J. (1999). Type Grammar Revisited. In A. Lecomte, F. Lamarche, & G. Perrier (Eds.), *Logical Aspects of Computational Linguistics. LACL 1997. Lecture Notes in Computer Science* (1582nd ed., pp. 1–27). Springer. https://doi.org/10.1007/3-540-48975-4_1
McCluskey, E. L. (1959). Minimization of boolean functions. *Bell System Technical Journal*, *35*(April), 149–175.
Merton, R. K. (1968). *Social Theory and Social Structure*. Free Press.
Nellhaus, T. (1998). Signs, social ontology, and critical realism. *Journal for the Theory of Social Behaviour*, *28*(1), 1–24. https://doi.org/10.1111/1468-5914.00060
Oates, T., Armstrong, T., Bonache, L. B., & Atamas, M. (2006). Inferring grammars for mildly context sensitive languages in polynomial-time. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4201 LNAI*, 137–147. https://doi.org/10.1007/11872436_12
Penrose, E. T. (1959). *The Theory of the Growth of the Firm*. John Wiley.
Quine, W. V. O. (1952). The Problem of Simplifying Truth Functions. *The American Mathematical Monthly*, *59*(8), 521–531. https://doi.org/10.2307/2308219
Schonfinkel, M. (1924). Uber die Bausteine der mathematischen Logik. *Mathematische Annalen*, *92*(3–4), 305–316. https://doi.org/10.1007/BF01448013
Steedman, M. (1987). Combinatory grammars and parasitic gaps. *Natural Language & Linguistic Theory*, *5*, 403–439.
Steedman, M. (1996). *Surface Structure and Interpretation*. The MIT Press.
Steedman, M. (2000). *The Syntactic Process*. MIT Press.
Turing, A. M. (1936). On Computable Numbers, With an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42: 230–265; cor. ibid., s2-43: 544–546 (1937). https://doi.org/10.1112/plms/s2-42.1.230
Vijay-Shanker, K., & Weir, D. J. (1994). The Equivalence of Four Extensions of Context-Free Grammars. *Mathematical Systems Theory*, *27*(6), 511–546.

http://srodev.sussex.ac.uk/id/eprint/504/1/mst94.pdf
Wolfram, S. (2002). *A New Kind of Science*. Wolfram Media, Inc. https://www.wolframscience.com/nks/
Wood, M. M. (1993). *Categorial Grammar*. Routledge.