The Chomsky-Schützenberger Theorem with Circuit Diagrams in the Role of Words

Tobias Heindel[1]

[1]Institute for Computer Science, University of Leipzig, Leipzig, Germany

Abstract

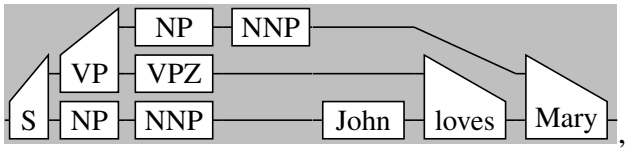
Guided by the idea of letters with finite lists of predecessors and successors, the paper develops the Chomsky-Schützenberger theorem for languages of arrows in any free PROP with a finite set of generators on the positive integers. The setting of monoidal categories is essential to obtain a well-behaved generalization of rational set, from monoids to monoidal categories.

Introduction

String diagrams have been gaining popularity over the last decade, especially in cross-disciplinary work on physics, logic, and computation. They arise in semantics of subject and object relative pronouns [10] as the graphical language [11] of compact closed categories. String diagrams are very intuitive, yet have formal semantics and thus bear the potential to convey theoretical subject matter to a wide audience [12].

The present paper considers string diagrams as syntactic entities that play the role of words in formal language theory in the spirit of Lafont's work on Boolean circuits [8]. Alphabets will be generalized to signatures of symbols with non-empty lists of "inputs" and "outputs". The arrows of free PROPs over such signatures can be seen as acyclic layouts of logic gates [11, Theorem 3.12].

We generalize Chomsky grammars in the obvious way to study context-free languages of arrows in free PROPs. Intuitively, context-free languages consist of (compositions of) string diagrams of tree-like shape. As example, consider the string diagram



,

which matches the phrase structure with its sentence, embedded as a branch into a tree. Note that this is not a tree but a non-trivial acyclic graph.

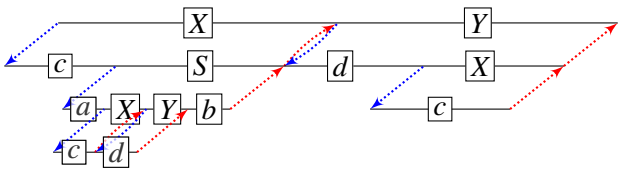
Our main contribution is the Chomsky-Schützenberger theorem for context-free languages of free PROPs, generalizing a classic result of formal language theory. In comparison to similar work on trees [1], the proposed Dyck languages of signatures are naturally seen as languages of matching brackets. The relation to context-free and recognizable graph languages [6] is left for future research.

1 Prelude: `dimension++`

One trait that we can find in formal language theory that extends existing results from words to more general structures is an extra dimension in illustrations of the objects that play the role of words. For example, if we consider the grammar

$$S \to aXYb \qquad X \to c \mid cS \qquad Y \to d \mid dX$$

we can use the illustration



in the x-z-plane to represent a parallel derivation: sentential forms are like bead necklaces laid out horizontally and derivation steps are rectangles.

We can read off the corresponding leftmost derivation by following the solid lines in direction of the x-axis, switching z-coordinate along dotted arrows. We apply a production whenever we encounter a blue arrow, replacing the variable to the right of the source by the sentential form to the right of its target that forms the opposite side of a rectangle in the x-z-plane (delimited by the next red arrow).

In the present paper, circuit diagrams [5] will play the role of words. To guide the intuition about context-free grammars of circuit diagrams, we think of symbols as (placeholders for) gates that are connected by wires and of productions as implementation of non-terminal gates by more complex circuits, which drive the derivation of circuit layouts of basic gates.

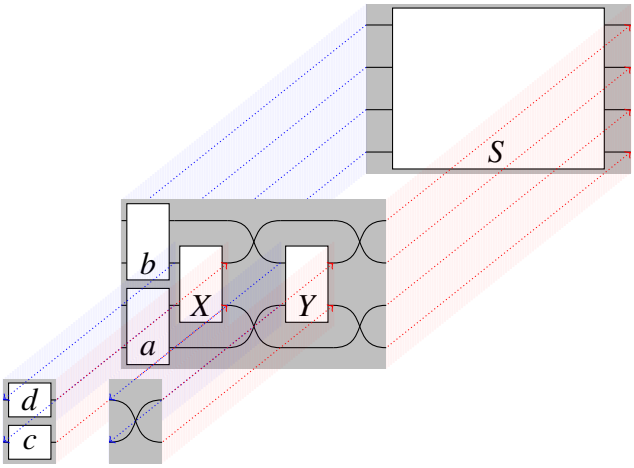


Figure 1: Parallel derivation of a circuit grammar

The extra dimension of circuit diagrams w.r.t. words is evident in Figure 1, which happens to be the illustration of a derivation in a circuit grammar—to be defined, after formalizing sequential and parallel composition of circuit diagrams in symmetric monoidal categories.

2 Preliminaries and notation

We start with notational conventions for symmetric monoidal categories and a definition of signature that induces free product and permutation categories (PROP) [9, 13].

Given a locally small category C and objects $A, B \in C$, the homset of arrows from A to B is denoted by $C(A, B)$. We shall use diagrammatic composition of arrows, denoted by the semicolon, i.e., the composition of arrows $f\colon A \to B$ and $g\colon B \to C$ in C, is denoted $f;g$. The identity on an object $A \in C$ is $\mathrm{id}_A\colon A \to A$. The monoidal product $\otimes$ of any PROP $(C, \otimes, I, \gamma)$ binds stronger than composition, i.e., $f;g \otimes h;k = f;(g \otimes h);k$ whenever the compositions are defined for arrows f, g, h, k in C.

A *signature* is a graph on the natural numbers, given by a triple $\mathbf{\Sigma} = (\Sigma, s, t)$ of a set Σ of *symbols* and two functions $s, t\colon \Sigma \to \mathbb{N}$, mapping symbols to their *arity* and *coarity*, respectively; we call it *alphabet-like* if both, arity and coarity of every symbol are positive. For a symbol $a \in \Sigma$, we write $a\colon m \to n$ if $s(a) = m$ and $t(a) = n$. The coproduct of signatures (Σ_1, s_1, t_1) and (Σ_2, s_2, t_2) is $(\Sigma_1 + \Sigma_2, [s_1, s_2], [t_1, t_2])$, the coproduct in the comma category $\mathbf{Set}/\mathrm{Id}_{\mathbf{Set}} \times \mathrm{Id}_{\mathbf{Set}}$. The *opposite* of a signature $\mathbf{\Sigma} = (\Sigma, s, t)$, denoted $\mathbf{\Sigma}^{\mathrm{op}}$, is (Σ, t, s). A *sub-signature* of $\mathbf{\Sigma}$ is a signature $\mathbf{\Upsilon} = (\Upsilon, i, o)$ such that $\Upsilon \subseteq \Sigma$, $i \subseteq s$, and $t \subseteq o$ (identifying functions with their graphs).

We fix a signature $\mathbf{\Sigma} = (\Sigma, s, t)$ for the remainder of the paper. The free PROP with generators $\mathbf{\Sigma}$ is denoted by $\mathcal{F}\mathbf{\Sigma}$. For any sub-signature $\mathbf{\Upsilon} \subseteq \mathbf{\Sigma}$, we assume $\mathcal{F}\mathbf{\Upsilon}$ to be a symmetric monoidal subcategory of $\mathcal{F}\mathbf{\Sigma}$ in the obvious manner. Finally, arrows of free PROPs are often called *cicuit diagrams*.

3 Grammars in free PROPs

We define the analogue of Chomsky grammars and formal languages in the setting of free PROPs, focussing on the context-free case. A *language* over a signature is a set of arrows in its free PROP. The basic idea of deriving in a context-free grammar consists in replacing a (non-terminal) symbol of the signature by an arrow in the free PROP that matches the arity and co-arity of the symbol and is specified by the grammar. The basic principle at work is rewriting or reduction in context, which is common in formalisms of theoretical computer science, such as the λ-calculus, configuration graphs of automata in (coloured) product categories [4], or graph rewriting [7], especially in relation to symmetric monoidal theories [3].

A *production* in a free PROP over $\mathbf{\Sigma}$ is a set of pairs of $\mathcal{F}\mathbf{\Sigma}$ arrows that share domain and codomain, i.e., a subset

$$R \subseteq \bigcup_{m,n \in \mathbb{N}} \mathcal{F}\mathbf{\Sigma}(m, n) \times \mathcal{F}\mathbf{\Sigma}(m, n).$$

Productions are thought of as directed and their

components are called *left* and *right hand side*, respectively. Re-using notation for productions, we have the following examples (cf. Figure 1).

$$S \to a \otimes b ; \mathrm{id}_1 \otimes X \otimes \mathrm{id}_1 ; \gamma \otimes \gamma ; \mathrm{id}_1 \otimes Y \otimes \mathrm{id}_1 ; \gamma \otimes \gamma$$
$$X \to c \otimes d$$
$$Y \to \gamma$$

A *rewriting context* for a production (l, r) is a quadruple (f, i, j, g) of arrows f, g in $\mathcal{F}\mathbf{\Sigma}$ and natural numbers i, j such that $f ; \mathrm{id}_i \otimes l \otimes \mathrm{id}_j ; g$ is defined in $\mathcal{F}\mathbf{\Sigma}$. The *derivation relation* of a production (l, r), denoted by $\Rightarrow_{l,r}$, relates two arrows h and k in $\mathcal{F}\mathbf{\Sigma}$ if $h = f ; (\mathrm{id}_i \otimes l \otimes \mathrm{id}_j) ; g$ and $k = f ; (\mathrm{id}_i \otimes r \otimes \mathrm{id}_j) ; g$ hold for some rewriting context (f, i, j, g). A derivation of the above productions is illustrated in Figure 1 (in its parallel form).

The definition of grammar is as expected.

Definition 1 (Circuit grammar)**.** A *circuit grammar* is a quadruple $\mathcal{G} = (\mathbf{\Sigma}, \mathbf{\Upsilon}, P, S)$ where

- $\mathbf{\Sigma} = (\Sigma, s, t)$ and $\mathbf{\Upsilon} = (\Upsilon, \iota, o)$ are signatures of *terminals* and *variables*, respectively, such that $\Upsilon \cap \Sigma = \varnothing$;
- P is a finite set of productions in the free PROP $\mathcal{F}(\mathbf{\Sigma} \cup \mathbf{\Upsilon})$ whose left hand sides do *not* belong to $\mathcal{F}\mathbf{\Sigma}$ where $\mathbf{\Sigma} \cup \mathbf{\Upsilon}$ is the component-wise union $(\Sigma \cup \Upsilon, s \cup \iota, t \cup o)$; and
- $S \in \Upsilon$ is the *start symbol*.

The grammar $\mathcal{G}$ is *context-free* if left hand sides of all productions are variables.

Concerning syntacticity of grammars, it can be shown that each production has a corresponding expression of the following specification.

$$\mathtt{f} ::= u \mid \gamma \mid \mathrm{id}_0 \mid \mathrm{id}_1 \mid (\mathtt{f};\mathtt{f}) \mid (\mathtt{f} \otimes \mathtt{f}) \quad (u \in \Sigma \cup \Upsilon)$$

An arrow f in $\mathcal{F}\mathbf{\Sigma}$ is *derivable* in a grammar $(\mathbf{\Sigma}, \mathbf{\Upsilon}, P, S)$ if

$$S \Rightarrow_{l_1,r_1} f_1 \cdots \Rightarrow_{l_n,r_n} f_n = f$$

holds for some sequence of productions $(l_i, r_i) \in P$ and arrows f_i in $\mathcal{F}(\mathbf{\Sigma} \cup \mathbf{\Upsilon})$ $(i = 1, \ldots, n)$. The *language* of a grammar $\mathcal{G}$, denoted by $L(\mathcal{G})$, is the set of all $\mathcal{F}\mathbf{\Sigma}$-arrows that are derivable. Finally, a *context-free circuit language* is the language of some context-free circuit grammar.

4 The theorem

We shall introduce natural counterparts of Dyck languages and rational sets to generalize the Chomsky-Schützenberger Theorem.

Theorem 1 (Chomsky-Schützenberger)**.** *A language L over an alphabet Σ is context-free if, and only if, there exists an alphabet Ξ, a rational set R over $\Xi + \Xi = \{0, 1\} \times \Xi$, and a homomorphism $h \colon (\Xi + \Xi)^* \to \Sigma^*$ such that $L = h(D_\Xi \cap R)$.*

Concerning the Dyck language, note that it corresponds to the least sub-monoid $\mathcal{D}_\Xi$ of the free monoid over $\Xi + \Xi = \{0, 1\} \times \Xi$ that contains the word $(0, u)w(1, u)$ whenever $u \in \Xi$ is a letter and the word w belongs to $\mathcal{D}_\Xi$. Replacing sub-monoid by monoidal subcategory, the *Dyck category* over $\mathbf{\Xi}$ is the least monoidal subcategory $\mathcal{D}_\mathbf{\Xi}$ of the free PROP $\mathcal{F}(\mathbf{\Xi} + \mathbf{\Xi}^{\mathrm{op}})$ such that $(0, u) ; f ; (1, u)$ is a $\mathcal{D}_\mathbf{\Xi}$-arrow whenever $u \colon m \to n$ is a symbol of $\mathbf{\Xi}$ and $f \colon n \to n$ is a $\mathcal{D}_\mathbf{\Xi}$-arrow. The *Dyck language*, denoted by $D_\mathbf{\Xi}$, is the set of arrows of the Dyck category.

Monoidal rational sets in a free PROP $\mathcal{F}\mathbf{\Xi}$ are the elements of the least set $\mathcal{R}$ such that

- $\mathcal{R}$ contains all finite sets of $\mathcal{F}\mathbf{\Xi}$-arrows;
- $L ; L' \in \mathcal{R}$ whenever $L, L' \in \mathcal{R}$ where
$$L ; L' = \left\{ f ; g \;\middle|\; \begin{array}{l} f \in L, g \in L', \\ f ; g \text{ is defined.} \end{array} \right\};$$
- $L \otimes L' \in \mathcal{R}$ whenenver $L, L' \in \mathcal{R}$ where $L \otimes L' = \{f \otimes g \mid f \in L, g \in L'\}$;
- $L^{*_\otimes} := \bigcup_{k \in \mathbb{N}} L^{k_\otimes} \in \mathcal{R}$ whenever $L \in \mathcal{R}$ where $L^{0_\otimes} = \{\mathrm{id}_0\}$ and $L^{k_\otimes} = L^{(k-1)_\otimes} \otimes L$; and
- $L^{*_;} := \bigcup_{k \in \mathbb{N}} L^{k_;} \in \mathcal{R}$ whenever $L \in \mathcal{R}$ where $L^{0_;} = \{\mathrm{id}_1\}^{*_\otimes}$ and $L^{i_;} = L^{(i-1)_;} ; L$.

Note that we also have a monoidal version of the Kleene star, as otherwise one could specify only PROP languages of arrows of bounded path width.

Theorem 2. *A* PROP *language L over an alphabet-like signature $\mathbf{\Sigma}$ is context-free if, and only if, there exists a signature $\mathbf{\Xi}$, a monoidal rational set R over $\mathbf{\Xi} + \mathbf{\Xi}^{\mathrm{op}}$, and a functor $\mathcal{H} \colon \mathcal{F}(\mathbf{\Xi} + \mathbf{\Xi}^{\mathrm{op}}) \to \mathcal{F}\mathbf{\Sigma}$ such that $L = \mathcal{H}(D_\mathbf{\Xi} \cap R)$.*

The proof re-uses the ideas of Ref. [2]. In particular the encoding of derivations in words over an extended alphabet can be adapted to encodings of derivations in circuit grammars—at least if the signature is alphabet-like. It is an open problem whether this restriction can be dropped.

5 Conclusion

Besides the rather natural notion of context-free languages of circuit diagrams, which are considered elsewhere [14], we have introduced Dyck languages in free PROPs that naturally fit the intuition of matching brackets. The main contribution is the Chomsky-Schützenberger theorem for languages of arrows in circuit PROPs for alphabet-like signatures; this generalization hinges on the notion of monoidal rational set, involving a monoidal Kleene star, similar to Ref. [4]. The theorem illustrates that free PROPs are a suitable setting for the development of classic results of language theory.

To the author's knowledge, this is the first time that the Chomsky-Schützenberger theorem has been developed for a non-trivial class of graph-like structures that do not consist of trees [1]. Besides future work on the relation to Courcelle's work [6], the notion of look-ahead in parsing offers itself as a promising research field, as part of a formal language theory for PROPs and PROs à la Chomsky.

References

[1] André Arnold and Max Dauchet. Un théorème de Chomsky-Schützenberger pour les forêts algébriques. *Calcolo*, 14(2):161–184, 1977.

[2] Jean-Michel Autebert, Jean Berstel, and Luc Boasson. *Context-Free Languages and Pushdown Automata*, pages 111–174. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

[3] Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, PawełSobociński, and Fabio Zanasi. Rewriting modulo symmetric monoidal structure. In *Proceedings of LICS 2016*, pages 710–719, New York, NY, USA, 2016. ACM.

[4] Francis Bossut, Max Dauchet, and Bruno Warin. A Kleene theorem for a class of planar acyclic graphs. *Information and Computation*, 117(2):251–265, 1995.

[5] Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, 2017.

[6] Bruno Courcelle and Joost Engelfriet. *Graph structure and monadic second-order logic: a language-theoretic approach*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge, 2012.

[7] Hartmut Ehrig, Michael Pfender, and Hans Jürgen Schneider. Graph-grammars: An algebraic approach. In *Proceedings of SWAT 1973*, pages 167–180, 1973.

[8] Yves Lafont. Towards an algebraic theory of Boolean circuits. *Journal of Pure and Applied Algebra*, 184(2-3):257–310, 2003.

[9] Saunders MacLane. Categorical algebra. *Bulletin of the American Mathematical Society*, 71(1), 1965.

[10] Mehrnoosh Sadrzadeh, Stephen Clark, and Bob Coecke. The Frobenius anatomy of word meanings I: subject and object relative pronouns. *Journal of Logic and Computation*, 23(6):1293, 2013.

[11] Peter Selinger. *A Survey of Graphical Languages for Monoidal Categories*, pages 289–355. Springer Berlin Heidelberg, 2011.

[12] Paweł Sobociński. Linear algebra with diagrams. *Chalkdust Magazine*, 5:10–17, 2017.

[13] Donald Yau and Mark W. Johnson. *A foundation for PROPs, algebras, and modules*, volume 203. American Mathematical Society, 2015.

[14] Vladimir Nikolaev Zamdzhiev. *Rewriting Context-free Families of String Diagrams*. PhD thesis, University of Oxford, Oriel College, 2016.