Hyperedge replacement grammars (HRGs, see [7, 6]) are one of the most successful formal models for the generative specification of graph languages, thanks to the fact that their language-theoretic and algorithmic properties to a great extent resemble those of context-free grammars. Unfortunately, polynomial parsing is an exception from this general rule: graph languages generated by HRGs may be NP-complete. Thus, not only is the uniform membership problem intractable (unless \( P \neq NP \)), but the non-uniform one is as well [1, 8].

Recently, Chiang et al. [5] advocated the use of hyperedge-replacement for describing meaning representations in natural language processing, and in particular the abstract meaning representations (AMRs) proposed by Banarescu et al. [2]. Chiang et al. described a general recognition algorithm building upon earlier work by Lautemann [9], together with a detailed complexity analysis. Unsurprisingly, the running time of the algorithm is exponential even in the non-uniform case, one of the exponents being the maximum degree of nodes in the input graph. Unfortunately, this is one of the parameters one would ideally not wish to limit, since AMRs may have unbounded node degree. However, AMRs and similar linguistic models to represent meaning are usually directed acyclic graphs (DAGs), a fact that is not exploited in [5]. In AMRs, the outgoing edges from a node represent linguistic arguments. They are ordered, and we can thus view the graphs as ordered. This is another crucial feature to exploit in order to achieve polynomial time parsing.

We present Order-Preserving DAG Grammars (OPDGs), a fragment of HRGs that generates single-rooted, ordered (hyper-)DAGs. It is expressive enough to capture AMR graphs and allows for polynomial time parsing, even in the uniform setting.

We further study the structure of the class of graphs generated by OPDGs. An algebraic characterization of the graphs yields a Myhill-Nerode theorem for the OPDG graph languages. This theorem is then used to define a MAT learning algorithm for OPDGs.

Definitions

A ranked alphabet is a pair \( (\Sigma, \text{rank}) \) consisting of a finite set \( \Sigma \) of symbols and a ranking function \( \text{rank} \). For a set \( F \), let \( S^* \) be the set of non-repeating sequences of elements of \( S \).

Graphs. Let \( \Sigma \) be a ranked alphabet. A (directed edge-labelled) hypergraph over \( \Sigma \) is a tuple \( g = (V, E, \text{src}, \text{tar}, \text{lab}) \) consisting of

- finite sets \( V \) and \( E \) of nodes and edges, respectively,
- source and target mappings \( \text{src}: E \rightarrow V \) and \( \text{tar}: E \rightarrow V^* \) assigning to each edge \( e \) its source \( \text{src}(e) \) and its sequence \( \text{tar}(e) \) of targets, and
- a labelling \( \text{lab}: E \rightarrow \Sigma \) such that \( \text{rank} (\text{lab}(e)) = |\text{tar}(e)| \) for every \( e \in E \).
A path in $g$ is a finite and possibly empty sequence $p = e_1, e_2, \ldots, e_k$ of edges such that for each $i \in [k - 1]$ the source of $e_{i+1}$ is a target of $e_i$. The length of $p$ is $k$, and $p$ is a cycle if $\text{src}(e_1)$ appears in $\text{tar}(e_k)$. If $g$ does not contain any cycle then it is a directed acyclic graph (DAG). The height of a DAG $G$ is the maximum length of any path in $g$. A node $v$ is a descendant of a node $u$ if $u = v$ or there is a nonempty path $e_1, \ldots, e_k$ in $g$ such that $u = \text{src}(e_1)$ and $v \in \text{tar}(e_k)$. An edge $e'$ is a descendant edge of an edge $e$ if there is a path $e_1, \ldots, e_k$ in $g$ such that $e_1 = e$ and $e_k = e'$.

The in-degree of a node $u \in V$ is $|\{e \in E \mid u \in \text{tar}(e)\}|$. The out-degree is defined symmetrically. A node with in-degree 0 is a root and a node with out-degree 0 is a leaf. For a single-rooted graph $g$, we write $\text{root}(g)$ for the unique root node.

A marked DAG is a tuple $g = (V, E, \text{src}, \text{tar}, \text{lab}, X)$ where $(V, E, \text{src}, \text{tar}, \text{lab})$ is a DAG and $X \in V^\circ$ is nonempty. The sequence $X$ is called the marking of $g$, and the nodes in $X$ are referred to as external nodes. Examples of marked DAGs can be seen in Figure 1.

**Order-preserving DAG grammars**

The grammars we define and investigate are a restriction of general hyperedge replacement grammars. When replacing a hyperedge marked by a terminal, we only allow graphs of three different kinds. All of them are DAGs and the three kinds are illustrated in Figure 1. Suppose that we are to replace a nonterminal edge labeled $A$ that has one source and three targets. Then we can replace it with a “clone graph” in which we see two copies of the edge, connected to the same nodes, in the same order. The second option is a terminal graph with only one edge. It is connected to the same nodes, in the same order, but may also introduce new nodes. The third option is a graph of height 2, where there is only one edge connected to the unique root. This edge has to be terminal. On the second level, we have nonterminals. Again, all edges that are connected to the nodes from the original graph have to have the connections in the same order as the original nonterminal edge.

**Acknowledgements.** This abstract is based on the papers [4, 3]. We thank Johanna Björklund, Frank Drewes, Petter Ericson for their parts in this work. We also thank Florian Starke for many useful discussions.
References


