ISSN 1870-4069

# Grafos de análisis sintáctico con gramáticas HRGs

Yolanda Moyao[1], Darnes Vilariño[1], Carlos Guillén[2], José de Jesús Lavalle[1]

[1] Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación

[2] Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias Físico Matemáticas

{ymoyao,dvilarino,jlavalle}@cs.buap.mx, cguillen@fcfm.buap.mx

**Resumen.** En este trabajo se realiza una investigación respecto al estado actual de los trabajos relacionados con la forma en que las Gramáticas de Reemplazo de Hiperaristas (HRGs) pueden ser usadas para definir modelos de lenguajes para transformación de grafos, con el objetivo de resolver problemas en diferentes áreas del Procesaminto de Lenguaje Natural (PLN) [5], tales como, análisis sintáctico y semántico, desambiguación del sentido de la palabra, comprensión de texto y resumen, por nombrar algunas. Debido a que el uso de las gramáticas HRGs puede facilitar la comprensión y generación de aplicaciones del Procesamiento de Lenguaje Natural(PLN), se ha incrementado el interés por el estudio del análisis basado en grafos, en partíclar, los grafos se consideran como una herramienta apropiada para representar la estructura semántica dentro del PLN.

**Palabras clave:** Gramática de reemplazo de hiperarista, procesamiento de lenguaje natural y lenguajes basados en grafos.

## Graphs of Syntactic Analysis with Grammars HRGs

**Abstract.** In this work an investigation is made regarding the current state of the works related to the way in which the Hyperarist Replacement Grammar (HRGs) can be used to define language models for graph transformation, with the objective of solving problems in different Areas of Natural Language Processing (PLN) [5], such as, syntactic and semantic analysis, disambiguation of the sense of the word, text comprehension and summary, to name a few. Because the use of grammars HRGs can facilitate the understanding and generation of applications of Natural Language Processing (PLN), interest in the study of graph-based analysis has increased, in particular, graphs are considered as a tool appropriate to represent the semantic structure within the PLN.

**Keywords.** Hyperarist replacement grammar, natural language processing and graph-based languages.

## 1. Introducción

Las gramáticas HRGs, son una generalización de la Gramática Libre de Contexto (CFG) para grafo lenguajes. Un grafo lenguaje es hipergrafo dirigido con hiperaristas etiquetadas sobre marcos semánticos, entidades y roles o argumentos. Una CFG es una gramática que construye cadenas al ir reemplazando símbolos con nuevas subcadenas, mientras que una gramática HRG crea grafos al ir reemplazando aristas con subgrafos.

En los últimos años uno de los temas que ha recibido atención, dentro de la comunidad de Procesamiento de Lenguaje Natural (NLP), son las gramáticas de grafos libres de contexto, ya que éstas pueden ser usadas para generar lenguajes de cadenas sensibles al contexto.

Una de las tareas algorítmicas más importantes en relación con las gramáticas de grafos libre de contexto, así como en las Gramáticas de Reemplazo de Hiperaristas (HRGs), es el análisis sintáctico [5].

Las HRGs son los candidatos formales para la generación y análisis de la representación semántica basada en grafos [18], de hecho, se consideran una herramienta efectiva para resolver problemas de análisis y generación automática de lenguajes libres de contexto, este tipo de gramáticas sirven para resolver cualquier problema dentro del PLN cuya representación pueda llevarse a cabo a través de Grafos Aciclicos Dirigidos (DAG)[10].

Sin embargo, existen problemas dentro del área de PLN que al hacer uso de las HRGs pueden generar lenguajes de grafos libres de contexto para los cuales el análisis semántico es NP-Completo, por tal motivo las investigaciones se enfocan al estudio de técnicas como el árbol de descomposición y el ancho de árbol que hagan a los algoritmos eficientes para algunas clases de grafos restringidos [4].

Existen algunos casos especiales cuya complejidad puede ser polinomial, como el analizador especializado para las HRGs canónicas, las cuales puede ser analizadas de manera eficiente en un tiempo de orden $O(n^c)$, donde c es el número máximo de nodos del lado derecho de la regla presentado por [10]. En este artículo se presentan algunos métodos aplicados por los diferentes autores para las aplicaciones del PLN, a través del uso de las gramáticas HRGs, de la descomposición arbórea y el ancho arbóreo.

El presente artículo está organizado de la siguiente manera. En la sección 2 se presentan los avances en los trabajos de investigación relacionados a la aplicación de los HRGs, en problemas de PLN, en la sección 3 se muestra de la misma manera los avances obtenidos para resolver problemas de PLN con el uso de AMR; por último en las conclusiones se discuten algunas posibles líneas de investigación.

## 2. HRGs

Los algoritmos para analizar grafos se sabe que de manera general son de complejidad exponencial, es por ello que se está trabajando en diferentes líneas de investigación, enfocadas a la aplicación de teoría de transformación de grafos

a NLP, con el objetivo de desarrollar algoritmos que sean eficientes para la generación y análisis de la representación semántica basada en grafos de propósito particular, tal es el caso del estudio y desarrollo de técnicas que se basan en la traslación de las gramáticas HRGs a analizadores de grafos, los cuales ofrecen un gran potencial para aplicaciones del lenguaje natural, en particular, para la generación y comprensión del mismo [2].

Una Gramática de Reemplazo de Hiperarista (HRG) es una tupla $G = (N, T, P, S)$ donde,

- $N$ y $T$ son conjuntos disjuntos finitos de símbolos terminales y no terminales.
- $S \in N$ es el simbolo inicial.
- $P$ es un conjunto finito de producciones de la forma $A \to R$, donde $A \in N$ y $R$ es un grafo fragmentado sobre $N \cup T$.

Un hipergrafo fragmentado es una tupla $(V, E, l, X)$, donde $(V, E, l)$ es un hipergrafo y $X \in V^+$ es una lista de nodos disjuntos llamados nodos externos. Los nodos externos indican como integrar un grafo dentro de otro grafo durante la derivación.

La complejidad en tiempo del algoritmo presentado en [13], es polinomial para el reconocimiento de lenguajes generados por la gramática HRG para grafos conectados de grado acotado, dicho algoritmo está basado en el trabajo presentado por Rozenberg et al. en [17] quiénes mostraron que el análisis sintáctico en gramáticas Controladas por Etiquetas de Nodo (BNLC) acotadas puede realizarse en tiempo polinomial para grafos de grado acotado, árboles, grafos bipartitos completos, grafo outerplanar maximal, grafo con ancho banda $\leq K$, grafos con ancho de corte $\leq K$.

Por otro lado, se tiene el caso del uso de las HRGs libres de contexto que pueden ser representadas por modelos basados en árboles y que además facilitan el uso de herramientas para autómatas arbóreos [12], también, se han presentado otros algoritmos en tiempo polinomial basados en esta técnica, tal es el caso del método para la traducción directa de HRGs a analizadores de grafos combinatorios [14], al igual que el método de [15], en el que utiliza las HRG2s, el árbol de descomposición y el ancho de árbol ($tw$). Éste juega un papel relevante en teoría de grafos, siendo una característica importante en el algoritmo de árbol de unión del aprendizaje autómatico [15], el cual ha demostrado ser valioso para el proceso de un análisis eficiente. [9]

Por otro lado, [3] describe a detalle un algoritmo más eficiente que el propuesto por [13] para reconocimiento de grafos, esto es factible al considerar que la complejidad computacional de las reglas de reescritura se puede mejorar, debido a que, al llevar a cabo el proceso de binarización de una CFG, que consiste en descomponer una regla de deducción en dos o mas reglas; donde cada una de las nuevas reglas tiene un número más pequeño de variables que la regla original.

También presenta un método de optimización que permite la ejecución en tiempo polinimial del algoritmo de análisis, siempre y cuando el ancho de árbol y el grado del grafo estén acotados; a pesar de esto, aún no se tiene un sistema completo que permita darle solución al problema cuando el tamaño del grafo es muy grande.

En la Fig. 1a se puede observar un grafo cuyo árbol de descomposición se muestra en la Fig. 1b, con un ancho de árbol igual a 3.

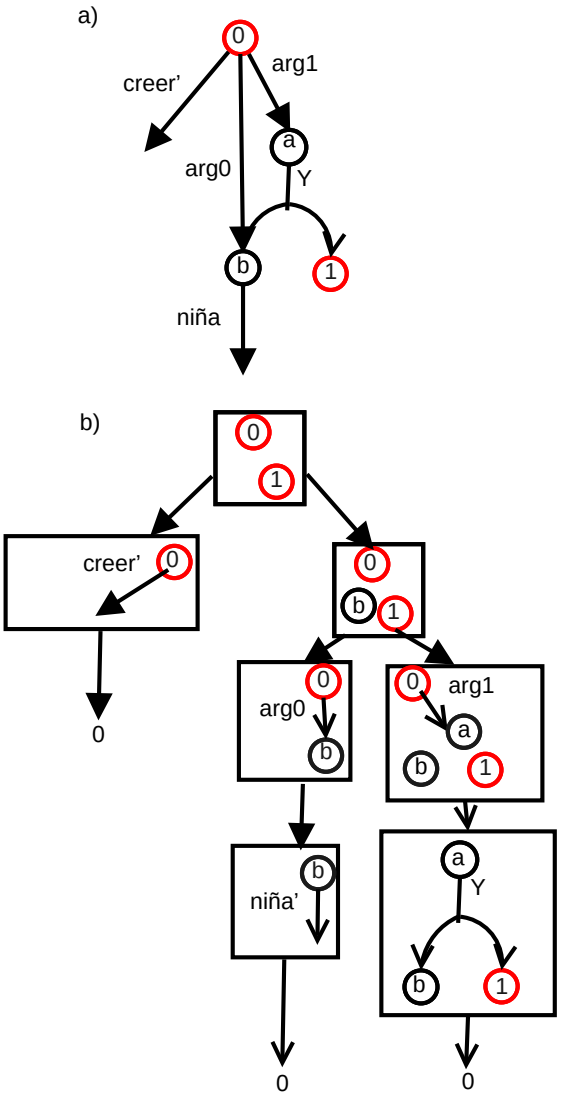


**Fig. 1.** (a) Parte derecha de la regla, y (b) Una descomposición de árbol.

También, se tienen los modelos generativos de lenguaje de grafos, como el que se introduce en [10], donde se presenta un marco simple para extraer automáticamente HRGs, basado en la definición de árbol de descomposición y

luego recorre el árbol para extraer reglas de una forma muy similar a como se extraen reglas de un corpus de árboles RTG (Gramáticas de Árbol Regulares).

Un árbol de descomposición de un grafo $G = (V, E)$ es un tipo de árbol, en el que se tiene un subconjunto de vértices G's para cada nodo. Los nodos de éste árbol T se definen con el conjunto I, y las áristas como el conjunto $F$. El subconjunto de V asociado con el nodo i de T se denota por $X^i$, donde $i \in I$ es un subconjunto de $V$, y el árbol $T$ tiene las siguientes propiedades:

1. Cubierta de vértice: Cada vértice de $G$ está contenido en al menos un nodo del árbol.
2. Cubierta de arista: Para cada arista e del grafo, hay un nodo árbol $n$ tal que cada vértice de $\alpha(e)$ está en $n$.
3. Manejo de intersección: Dados cualesquiera dos nodos árbol $n_0$ y $n_1$, ambos contienen al vértice $v$, todos los nodos árbol en el único camino de $n_0$ a $n_1$ también contienen a $v$.

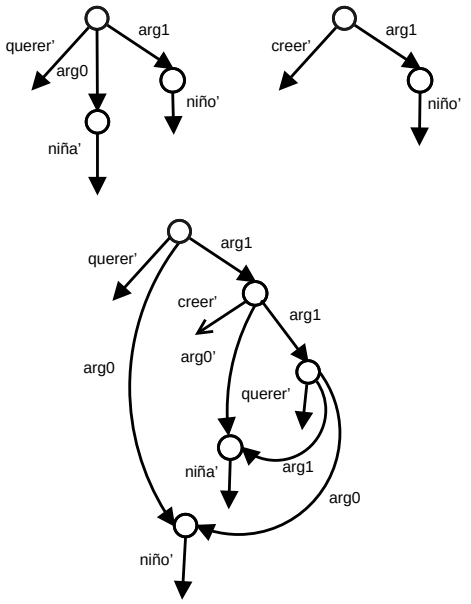


**Fig. 2.** Algunos elementos de un lenguaje de grafos, representando los significados de (en el sentido de las agujas del reloj desde la esquina superior izquierda): "La niña quiere al niño", "El niño se cree" y "El niño quiere que la niña crea que él la quiere".

Por otro lado, se tiene el caso de la propuesta que hace uso de árbol de descomposición de reglas y grafos de entrada y que trabaja con representaciones basadas en el límite de subgrafos, que ya han sido procesados [5]. Otra de las

líneas de investigación, se basa en lenguajes DAG ponderados y que son de interés en NLP, debido a que pueden ser usados para representar la estructura semántica de grafos de forma similar como los de AMR. [6].

A continuación, se muestra un ejemplo con el objetivo de aclarar el concepto de las gramáticas HRGs. Considere un lenguaje de grafo ponderado que involucre solo dos tipos de marcos semánticos (quiero y creo), dos tipos de entidades (niño y niña) y dos roles (arg0 y arg1). La Fig. 2 muestra algunos grafos de este lenguaje. La Fig. 3 muestra cómo derivar uno de estos grafos usando una HRG.

La derivación comienza con un única arista etiquetada con el símbolo no terminal $S$. El primer paso de re-escritura reemplaza esta arista con un subgrafo, que podemos leer como "El muchacho quiere que algo ($X$) involucra a sí mismo..[El] segundo paso de re-escritura reemplaza la arista $X$ con otro subgrafo, que podríamos leer como "El chico quiere que la chica crea algo ($Y$) involucrando a ambos ". La derivación continúa con un tercer paso de re-escritura, después del cual ya no hay aristas etiquetadas sin elementos terminales.

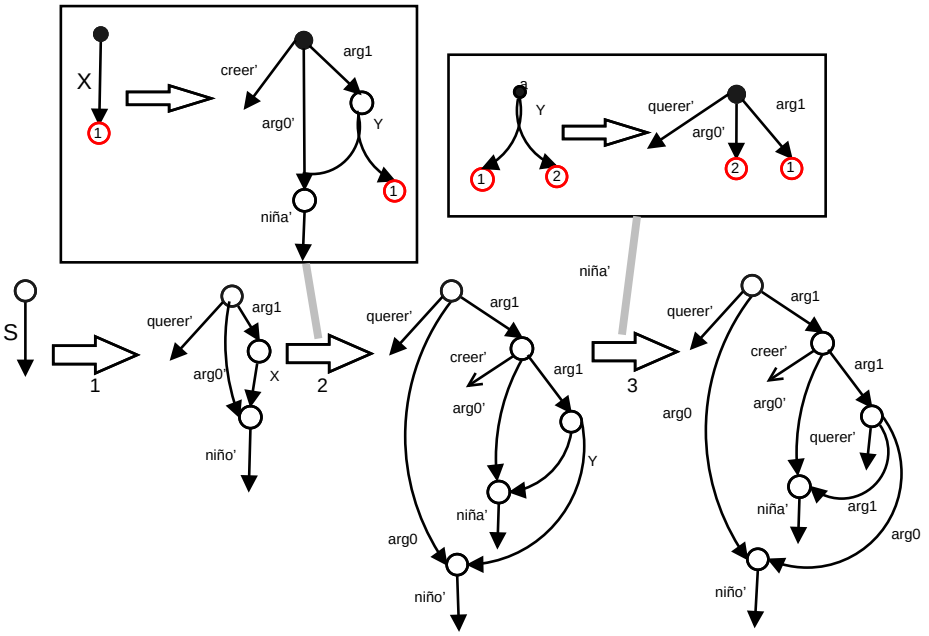


**Fig. 3.** Derivación de una gramática de sustitución de hiperarista para un grafo que representa el significado de "el chico quiere que la chica crea que él la quiere".

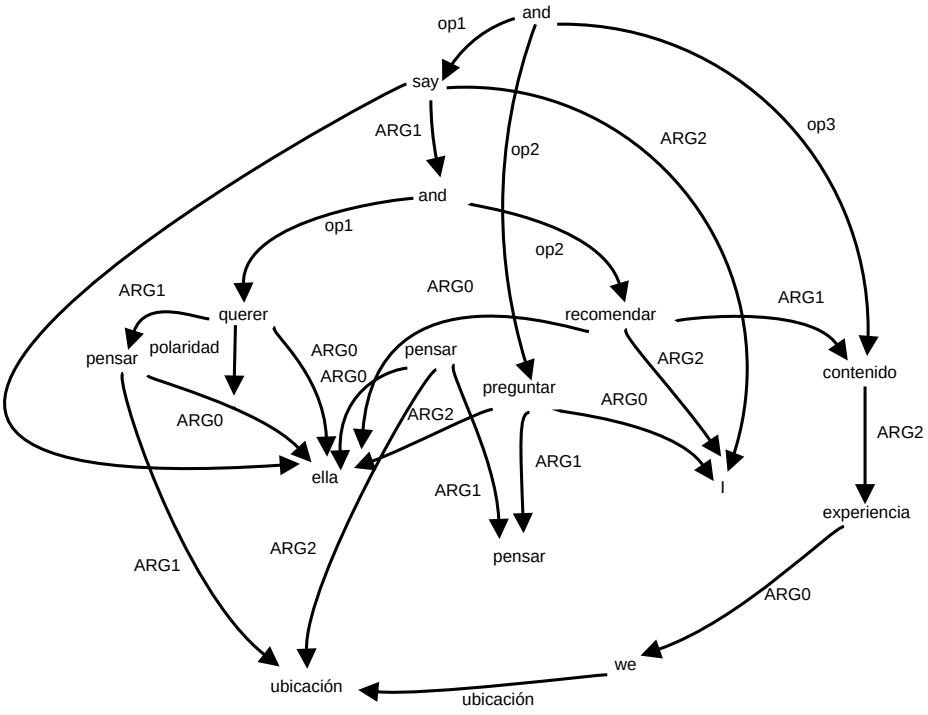


**Fig. 4.** Una AMR.

## 3. Grafos para la Representación de Significado Abstracto (AMR)

Las AMR fueron introducidas por [1] como una notación gráfica independiente del dominio para la semántica de significados en lenguaje natural. Las AMR tienen un propósito similar en el ámbito de la representación semántica como lo hace el conocido árbol constituyente para la representación sintáctica. Para este último, se tiene una gran variedad de modelos formales que representan la estructura de la representación correcta de la gramática de árbol regular.[8] y [7].

Un AMR generalmente no es un árbol sino un Grafo Aciclico Dirigido(DAG) como el AMR (algo simplificado y abstracto). Sus vértices son principalmente conceptos de PropBank. [11] conectados por aristas que están etiquetadas por etiquetas de roles, de forma intuitiva, suministrando los conceptos con sus argumentos semánticos.

En este caso, los árboles de dependencia muestran la similitud, pero también la diferencia, entre varios conceptos, que comparten un argumento semántico.

A continuación, se presenta un ejemplo, cuyo grafo se puede observar en la Fig. 4. "Le pregunté qué pensaba sobre dónde estaríamos y ella dijo ella no

quiere pensar en eso, y que debería estar feliz con las experiencias que hemos tenido (el cual Yo soy)."Los nombres de los cuadros de PropBank (las etiquetas de los vértices) se han simplificado en aras de la legibilidad. Se pueden encontrar en [3], que también usa este ejemplo, como nota al margen; uno puede notar que la AMR no especifica si su dicho fue la respuesta a mi pregunta, al revés, o los dos eran independientes. Esto, sin embargo, contribuiría a una discusión sobre los límites de AMR en lugar de modelos de autómatas formales para AMR.

Los AMR pueden ser utilizados para la construcción de Máquinas de Traducción Automática (MT), representación de corpus multilingues, entre otras tareas de PLN.

## 4. Conclusiones

De acuerdo a los trabajos revisados en este articulo se puede observar que en los últimos años ha ido creciendo el interés por la investigación de aplicaciones de gramáticas modeladas con grafos para resolver problemas dentro del área de PLN. Se pueden mencionar algunos resultados importantes, tales como, el uso de las HRGs que son consideradas como una herramienta efectiva para resolver problemas de comprensión y generación del lenguaje natural, siempre y cuando el problema pueda ser representado por grafos dirigidos y aciclicos.

También se puede concluir que de acuerdo a los trabajos relacionados al uso del árbol de descomposición de un grafo, en general tiene un comportamiento computacional $NP - Completo$, sin embargo, cuando el ancho de árbol para el grafo es acotado entonces, el comportamiento computacional para el algoritmo se puede reducir a un comportamiento de orden polinomial $O(n^c)$.

Dentro de las posibles líneas de investigación relacionadas con la presente revisión, se puede mencionar la necesidad de incorporar otras métricas arbóreas que conduzcan a algoritmos eficientes para clases menos restrictivas de grafos gramáticas, al mismo tiempo la inclusión de un análisis y tratamiento con parámetro fijo tratable más profundo.

## Referencias

1. Banarescu, L., Bonial, C., Cai, S., Madalina, Georgescu, Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., Schneider, N.: Abstract meaning representation for sembanking. Proceedings 7th Linguistic Annotation Workshop, ACL 2013, Workshop (2013)
2. Chiang, D., Drewes, F., Gildea, D., Lopez, A., Satta, G.: Weighted DAG automata for semantic graphs. Submitted (2013)
3. Chiang, D., Andreas, J., Bauer, D., Moritz, H.K., Jones, B., and Knight, K.: Parsing graphs with hyperedge replacement grammars. Proceedings of the 51st Meeting of the ACL (2013)
4. Drewes, F., Habel, A., Kreowski, H.: Hyperedge replacement graph grammars. In: Rozenberg, G., editor, Handbook of Graph Grammars and Computing by Graph Transformation, World Scientific, 95–162 (1997)

5. Drewes, F., Hoffmann, B., Minas, M.: Predictive top-down parsing for hyperedge replacement grammars. Proceedings 8th Int'l Conf. on Graph Transformation (ICGT'15), Lecture Notes in Computer Science, Springer (2015)
6. Drewes, F.: Tutorial: Introduction to Graph Transformation Report from Dagstuhl Seminar 15122 Formal Models of Graph Transformation in Natural Language Processing Edited by Drewes F., Knight K., and Kuhlmann M. (2017)
7. Drewes, F.: DAG Automata for Meaning Representation. Proceedings of the 15th Meeting on the Mathematics of Language, London, UK, Association for Computational Linguistics, pages 88–99, (2017)
8. Gécseg, F., Steinby, M.: Tree Automata. Akadémiai Kiadó, Budapest. Online version available under https://arxiv.org/ abs/1509.06233 (1984)
9. Gildea, D.: Grammar factorization by tree decomposition. Computational Linguistics, Association for Computational Linguistics, 37(1), 231–248 (2011)
10. Jones, B., Andreas, J., Bauer, D., Hermann K., Knigh, K.: Semantics-Based Machine Translation with Hyperedge Replacement Grammars. Proceedings of COLING 2012: Technical Papers, pages 1359–1376, COLING 2012, Mumbai, December (2012)
11. Kingsbury, P., Palmer, M.: From Treebank to propbank. Proc. 3rd Intl. Conf. on Language Resources and Evaluation (LREC 2002) (2002)
12. Knight, K., Graehl, J.: An overview of probabilistic tree transducers for natural language processing. Proceedings of the 6th International Conference on Intelligent Text Processing and Computational Linguistics (2005)
13. Lautemann, C.: The complexity of graph languages generated by hyperedge replacement. Acta Informatica, 27, 399–421 (1990)
14. Mazanek, S., Minas, M.: Parsing of hyperedge replacement grammars with graph parser combinators. Proc. 7th International Workshop on Graph Transformation and Visual Modeling Techniques (2008)
15. Moot, R.: Lambek grammars, tree adjoining grammars and hyperedge replacement grammars. Proc. TAG+9, pages 65–72 (2008)
16. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco, CA, 2nd edition (1988)
17. Rozenberg, G., Welzl, E.: Boundary NLC graph grammars basic definitions, normal forms, and complexity. Information and Control, 69, 136–167 (1986)
18. Teichmann, C. Drewes, F.: Efficient HRG-Parsing for the NLP Domain. Report from Dagstuhl Seminar 15122 Formal Models of Graph Transformation in Natural Language Processing Edited by Drewes F., Knight K., and Kuhlmann M. (2015)