Aplicaciones en TreeBag

Mauricio Aguirre, Eric Jeltsch, Gerardo Rosales.

Departamento de Matemáticas - Area de Computación
Universidad de La Serena, Benavente 980,
La Serena, Chile.

{maguirre, ejeltsch, grosales}@dns.uls.cl

Resumen: *En este trabajo se presenta el sistema* ***TreeBag*** *(****Tree-Ba****sed* ***G****eneration) el cual combina en forma activa diferentes tipos de objetos que son de nuestro interés, como son las gramáticas de árboles, transductores de árboles y álgebras. Basado en TreeBag queremos mostrar una variedad de aplicaciones en donde este sistema puede ser utilizado como una herramienta de apoyo para la enseñanza, aprendizaje e investigación en el campo de la informática teórica.*

Palabras claves: *Herramienta para la Enseñanza y Aprendizaje.*

(*) Trabajo financiado por Proyecto Nº 0220-2-20, DIULS (Dirección de Investigación de la Universidad de La Serena).

1 INTRODUCCIÓN

En general nuestro interés se relaciona con el hecho de que en Ciencias de la Computación es frecuente representar la información por árboles o grafos, y por ende existen objetos muy apropiados para describir conocimientos, información, o representar modelos. Ahora cualquier cambio local que pueda afectar a un árbol o grafo, puede quedar registrado a través de una regla o producción, de manera que cualquier proceso dinámico que represente cambios de estado, puede representarse a través de un conjunto de reglas, que conforman las llamadas gramáticas de árbol, las cuales proporcionan una alternativa para representar la información de una forma más general, en vez de utilizar cadenas de caracteres, facilitando de esta manera el acceso a un amplio espectro de aplicaciones en diversas áreas, como por ejemplo: Reconocimiento Sintáctico de Formas, Inferencia Gramatical, Semántica de Lenguajes, así como en la generación sistemática y manipulación de diseños gráficos, fractales, o la simulación del crecimiento de células o plantas en la Biología. Vea [Bar88], [PL90], [PJS92] y [FV98].

Las herramientas de apoyo generosos en visualizaciones es un gran resultado dentro de la ingeniería de software [Po98], desde productos comerciales como LEGO Mindstorms(W3), que está basado en el diseño, construcción y programación de un robot, hasta productos generados por universidades, que de acuerdo a nuestro interés hemos querido destacar aquellos que están basados en reglas o producciones de ciertas gramáticas o autómatas, [GS97] tales como: CollageSystem(W4), que es una familia de programas basados en la generación y animación de Formas Pictóricas, cuyo trasfondo teórico se sustenta en las gramáticas Collage, [HK91]. Otra herramienta es L-System(W5), que es un sistema basada en la generación de plantas con la componente de crecimiento y animación de las mismas, cuyo formalismo teórico se sustenta en las gramáticas de Lindenmayer, que son un tipo de gramáticas de contexto libre, [PL90]. Treebag que es un sistema que provee una amplia gama de recursos para la visualización de gramáticas de árboles y transductores en forma gráfica, este sistema está basado en la generación de gramáticas de árboles y la transformación realizada por los transductores, además ofrece la opción de usar algún tipo de álgebras para interpretar los árboles que se generan, originando con esto una amplia gama de formas pictóricas, [Dre00] y [Dre96]. Fujaba(W2) que es un sistema que realiza una jerarquerización de modelos usando el Lenguaje Unificado de Modelo UML, y por último el sistema AGG(W6) que es una familia de programas basados en gramáticas de grafos.

Hemos escogido TreeBag(W1), porque no tan sólo ofrece las ventajas de ser un sistema construido enteramente en Java, lenguaje de programación incorporado a los planes regulares de la carrera de Ingeniería en Computación de la Universidad de La Serena desde 1998, sino que en él se pueden combinar en forma activa cuatro diferentes tipos de componentes que son de nuestro interés: *Gramáticas de árboles* que generan árboles, *transductores de árboles* que transforman una entrada de árboles en una salida de árboles, *Álgebras* que interpretan árboles como expresiones sobre algún dominio y por último el despliegue de los valores que adquieren los árboles en la pantalla. En este contexto se muestran una variedad de aplicaciones, orientada a los procesos de enseñanza, aprendizaje e investigación en temas de Informática Teórica, tales como los lenguajes formales, compiladores, teoría de autómatas, la visualización del proceso generativo en las gramáticas de grafos, las transformaciones de los mismos a través de los transductores, la inferencia gramatical o reconocimiento de formas.

2 Estructura del sistema TreeBag

TreeBag es un sistema implementado en Java (JDK1.2) cuya interfaz se basa en el despliegue de ventanas (frames). La ventana principal es llamada "Treebag 1.2 worksheet", en la cual el usuario puede interactivamente construir una red de componentes TreeBag. Las clases habilitadas de los componentes están divididas en 4 categorías, ellas son: gramáticas de árboles, transductores de árboles, álgebras y desplieges.

La instancia de una clase es definida en un archivo de texto con una sintaxis especifica y puede ser invitada sobre el *worksheet*, donde ella es representada como un nodo. Estos nodos pueden estar conectados vía arcos para establecer la relación entrada-salida, para lo cual basta dibujar un arco desde una gramática de árbol o transductor de árbol a otro transductor de árbol. La ventana principal posee además un menú con las opciones básica para cargar(load), grabar(save), limpiar(clear) una hoja de trabajo(clear) y salir del sistema(quit).

La hoja de trabajo en TreeBag es la ventana principal del sistema, como se puede visualizar en la Fig.1 a) en donde el usuario puede situar instancias de componentes como nodos de un grafo abierto, estableciendo relaciones de entrada y salida al poner arcos dirigidos entre estos nodos. En b) se visualiza el archivo de configuración de la hoja de trabajo *configuration* , que es guardada como un archivo de texto que contiene la ubicación de los componentes, el estado de cada uno, y los arcos que les relacionan. Así Treebag solo necesita leer este archivo para saber su estado y poderlo desplegarlo por pantalla.

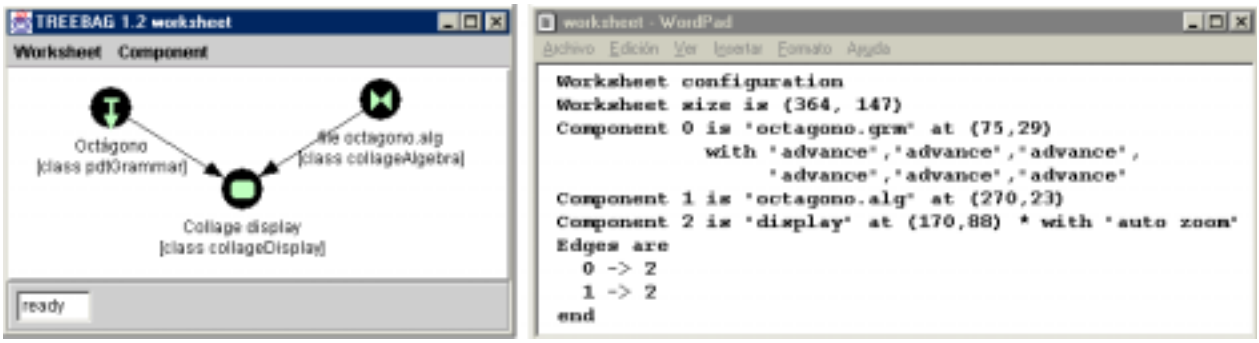


Fig. 1 *a) Treebag worksheet* *b) Archivo de configuración de worksheet.*

2.1 Componentes

La base teórica está sustentada por gramáticas y transductores de árboles, las gramáticas están representadas por el icono , este incluye: las gramáticas regulares de árbol, las gramáticas de árbol ETOL y las gramáticas totales deterministas paralelas de árbol. Por otra parte, los transductores están representados por el icono e incluye: Transductores de árbol top-down, Transductor YIELD, y una metaclase de transductores de árbol llamado iterator. Para la interpretación de la información generada disponemos de álgebras, cuyo icono representativo es , este componente evalúa los árboles de salidas en el dominio de los enteros, strings, árboles y los collages de dos dimensiones, así como también las álgebras que corresponden a formalismos de código de cadenas y tortugas (turtle), capaces de generar dibujos lineales. Y finalmente, para desplegar por pantalla es necesario utilizar los displays, representados por el icono , estos pueden desplegar una representación textual de objetos: números, strings, y árboles, o una representación gráfica mediante collages de dos dimensiones o dibujos lineales.

2.2 Configuración de componentes

Cada componente está declarado mediante una instancia de clase, a su vez, una instancia de clase está definida en un archivo de texto con una sintaxis particular y que pueden ser cargados sobre la hoja de trabajo como un nodo. Por ejemplo en la Fig. 2, se define una gramática regular de árbol, usual a las gramáticas regulares de cadenas de caracteres según la jerarquía Chomsky, en donde las producciones son de la forma, terminales o la concatenación entre un terminal y no-terminales.

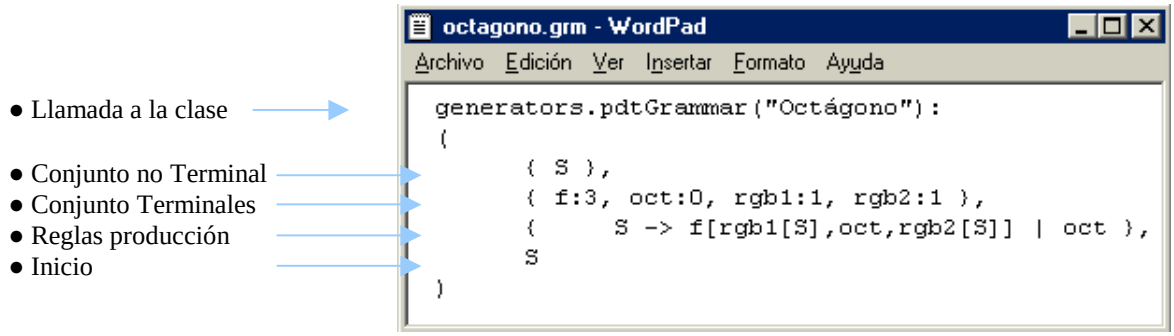


Fig. 2 *Gramática de árbol Regular.*

Notar que en el conjunto de terminales, f tiene rango 3, oct tiene rango 0, rgb1 y rgb2 tienen rango 1, esto significa que el árbol asociado tendrá 3, 0, 1 y 1 hijos respectivamente. Un álgebra collage que interpreta en forma gráfica la gramática anterior es:

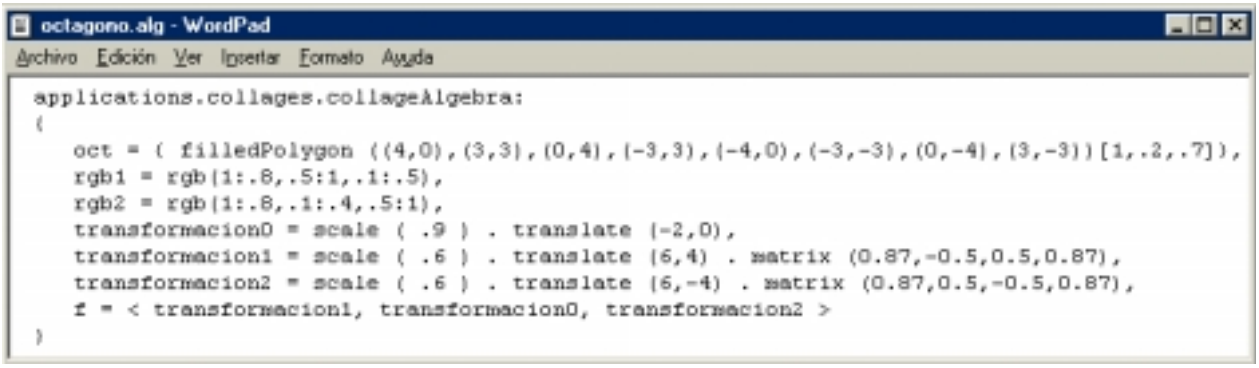


Fig. 3 *Álgebra collage para la gramática de la Fig. 2.*

En esta álgebra definimos la forma geométrica, que son transformaciones 2D y de color que tendrán asociados los elementos de la gramática. Así, el elemento "oct" de la gramática es un polígono relleno de 8 vértices y color inicial en RGB en una escala de 0 a 1 dado por [1, 0.2, 0.7]. Análogamente, los elemento no terminales *rgb1* y *rgb* están definidos por transformaciones de color, en donde la notación "1: 0.8 "especifica el rango de variación para el color rojo (R). También se definen las variables transformacion0, transformacion1 y transformacion2 como una concatenación de transformaciones en dos dimensiones, el escalamiento está dado por *scale*, la traslación por *translate* y rotación por *matrix*, matriz de rotación ((cosα , -senα),(senα, cosα)). Este conjunto de transformaciones se utilizan para indicar que el símbolo *f* está compuesto por

ellas, recordar que tenía rango 3. Para los componentes displays, basta incluir en el archivo de configuración la llamada a la clase y el argumento *Postcript disabled*.

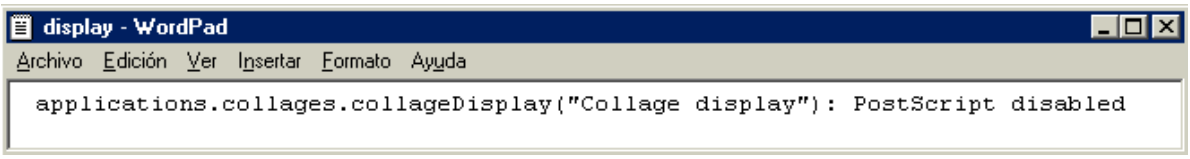
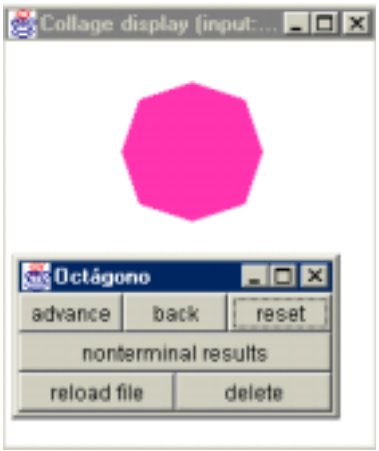


Fig. 4 *Ejemplo de Displays*

Es posible definir más de un álgebra y un displays para una gramática. Para agregar el componente a la hoja de trabajo worksheet, basta con hacer doble clic sobre el fondo, aparecerá un cuadro de dialogo de archivo en al cual se selecciona el correspondiente y presionamos el botón abrir. Para crear los arcos basta con hacer clic en un nodo y arrastrar el puntero del mouse hacia otro nodo. Existen algunas restricciones, no es posible por ejemplo, hacer un arco desde una gramática a un álgebra, o de un álgebra a una gramática, en todo caso Treebag realiza este tipo de validaciones y no deja establecer un arco cuando no hay compatibilidad entre los nodos relacionados.

2.3 Funcionamiento

Al tener ya configurada la hoja de trabajo worksheet con un conjunto de componentes veamos como funciona. Al hacer doble clic sobre algún nodo se despliega una ventana con un conjunto de opciones, Fig. 5(a), que tiene un botón con la opción *advance*, el que al presionarlo repetidas veces se genera un proceso recursivo, generando tras 10 iteraciones la Fig.5(b) que es un árbol de derivación tras una interpretación Collage , dando origen a una figura collage:



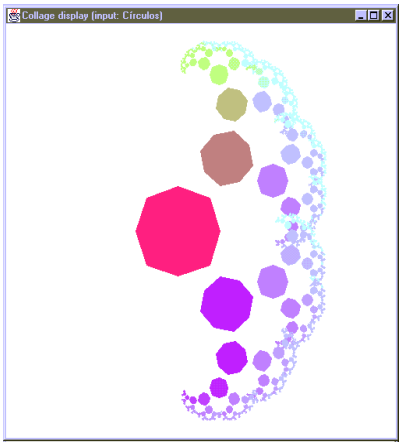


Fig. 5 *a) Estado inicial* *b) Luego de 10 iteraciones*

3 Aplicaciones de Treebag

TreeBag es una herramienta que provee una amplia gama de recursos para la visualización de diversos tipos de gramáticas de árbol y transductores de árbol en forma gráfica. Es posible la representación mediante números, strings, valores booleanos, árboles n-arios, fractales, collages, algoritmos y diagramas de flujo de un programa en algún lenguaje de programación.(Pascal, Java, Xml). Entre sus ejemplos desarrollados se encuentran la Curva del Dragón, curvas de Hilbert, triángulo de Sierpinsky, gramática de Barnsley, entre otros. A continuación se describirán algunas utilidades que ofrece Treebag.

3.1 Visualización de las producciones en Gramáticas de Árboles

Es posible asociar a la gramática de la Fig. 2 un par de elementos que nos permitirán desplegar por pantalla el árbol de derivación que se genera luego de aplicar las reglas de producción un número determinado de iteraciones, estos elementos son Álgebra y Display de árbol, basta con dirigir un arco desde el icono de la gramática y un arco desde el álgebra hasta el icono del display y hacer doble clic sobre él, la Fig. 6 muestra el árbol generado.

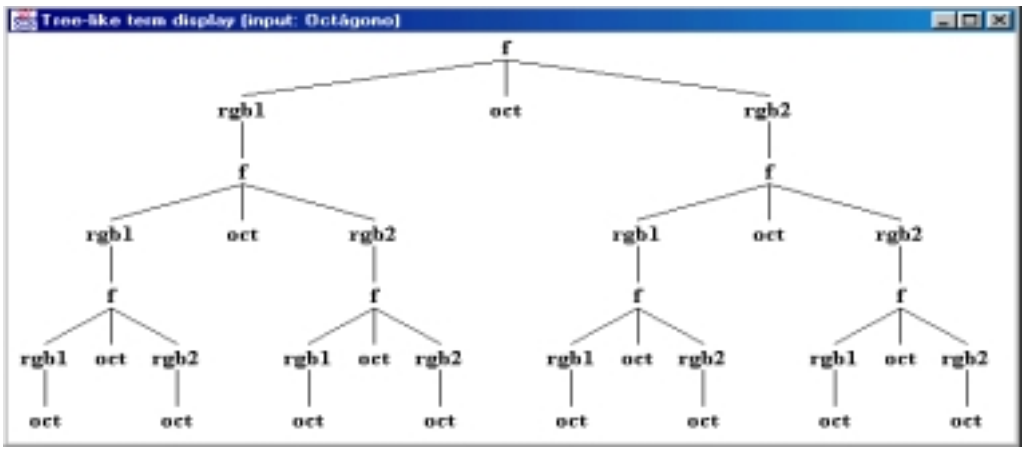


Fig. 6 *Display de árbol para la gramática de la Fig. 2.*

Mediante esta visualización podemos ver fácilmente que la cadena *"f rgb1 f rgb2 f rgb2 oct"* pertenece al lenguaje generado por la gramática *Octágono.* No así el elemento *"f oct rgb1 oct".*

3.2 Visualización de Árboles sintácticos.

Como las expresiones regulares nos proporcionan medios para especificar o definir un lenguaje, todas las cadenas que tienen un patrón en particular son las que lo conforman, ahora, si asociamos este conjunto de patrones generativos con sintaxis de algún lenguaje de programación, podemos obtener el seudo código particular de tal lenguaje. Usando gramática libre de contexto, que generan árboles de sintaxis, la Fig. 7 muestra un subconjunto de ciclos *while* y asignaciones de valores en un programa Pascal.

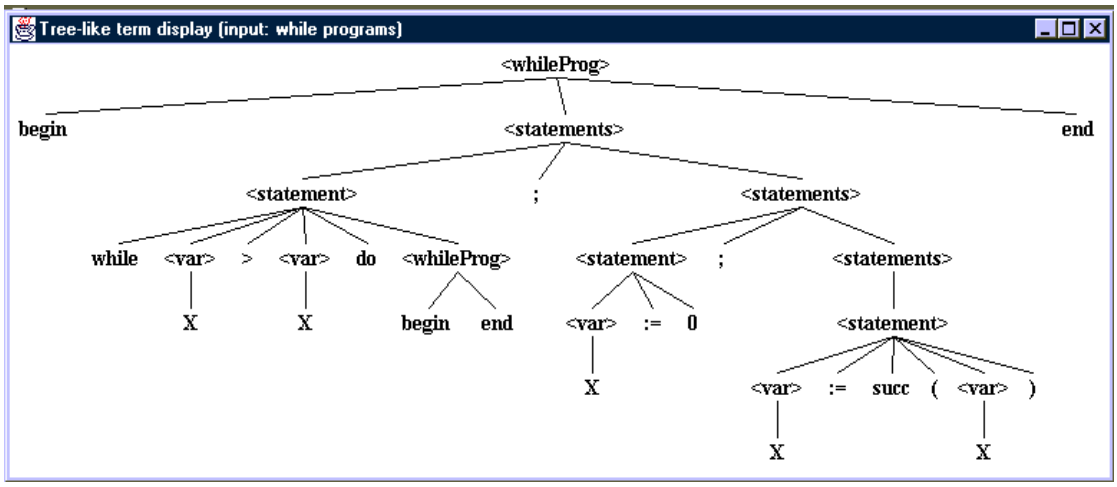


Fig. 7 *Árbol de derivación. Seudo código en Pascal.*

3.3 Visualización mediante Strings

TreeBag nos provee además de un componente "Textual display" para visualizar el recorrido del árbol de derivación en forma de strings y la representación de expresiones algebraicas. Volviendo a la Fig.7, dado el recorrido *inorden* del árbol de derivación obtenemos una cadena válida para la sintaxis de un programa en Pascal, tal como lo muestra la Fig. 8.

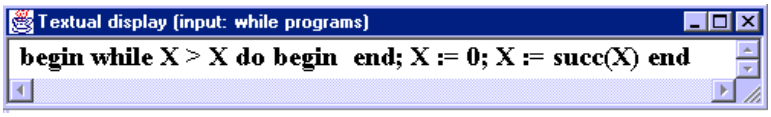


Fig. 8 *Seudo código en Pascal representado como un String*.

3.4 Visualización mediante Collage y Tortugas(Formas o Patrones)

La interpretación de gramáticas de árboles por álgebras turtle(tortuga), permite interpretar la gramática como una concatenación de dibujos lineales, cuyo trazado depende de un ángulo definido. Esta concatenación de líneas generan fractales como en la Fig. 9. Este tipo de álgebras es mucho mas simple de configurar. Basta con definir el nombre de la clase (constructor) pasando como argumento el ángulo de giro de la tortuga.

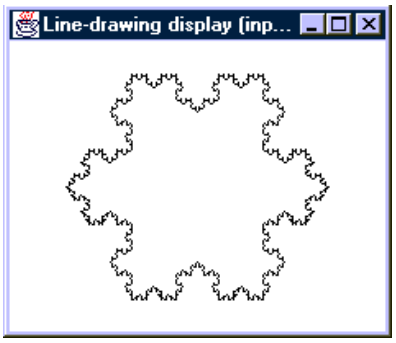


Fig. 9 *Fractal por Line-Drawing*.

3.5 Inferencia Gramatical

El problema de la inferencia gramatical se refiere a encontrar una descripción sintáctica, por ejemplo Gramáticas, Autómatas, Transductores o algún sistema, que permite la generación o el reconocimiento automático de un conjunto finito de modelos en S_+, el que está compuesto por un conjunto finito de modelos pertenecientes a algún lenguaje, también llamados ejemplos positivos, los que pueden ser cadenas de caracteres, árboles, grafos, modelos u otros. En nuestro caso, la descripción sintáctica es un sistema *td-generador con interpretes,* mientras que S_+ consta de árboles, y posiblemente un conjunto de árboles del complemento de S_+, también llamados ejemplos negativos. Para mayor información sobre este tema, vea http://eurise.univ-st-etienne.fr/gi/gi.html.

En este contexto se ha utilizado TreeBag para interpretar un tipo particular de generador de árbol top-down, llamado *td-generador*, el cual esta compuesto de una gramática de árbol regular y un transductor de árbol**.** El objetivo final es visualizar el proceso generativo de la gramática de árbol, en donde los árboles son interpretados por una álgebra Collage generándose formas pictóricas o modelos de un determinado dominio.

Sea S_+ = { F[S, S, S], G[A], S[H], G[G[A]], G[G[G[A]]], A[H], F[S, S, G[F'[S, S, S]]], F[S, S, G[F'[H, H, H]]], G[G[G[F'[S, S, S]]], S[G[F'[S, S, S]]] } un conjunto finito de ejemplos dentro de un lenguaje de árboles. Basado en algoritmo de inferencia [AJR02] se obtiene la gramática regular g = ({S, A}, {$F^{(3)}$, $F'^{(3)}$, $G^{(1)}$, $H^{(0)}$}, P, S), en donde las producciones son:

P={ $S \rightarrow F[S, S, S]$, $S \rightarrow G[A]$, $S \rightarrow H$, $A \rightarrow F'[S, S, S]$, $A \rightarrow G[A]$, $A \rightarrow H$}.

El td-generador es G = (g_{total} , λ), de donde L(G) contiene al menos las formas pictóricas de S_+.

Ahora definamos una álgebra collage capaz de interpretar los árboles primitivos en formas geométricas, tal como lo muestra la Fig. 10.

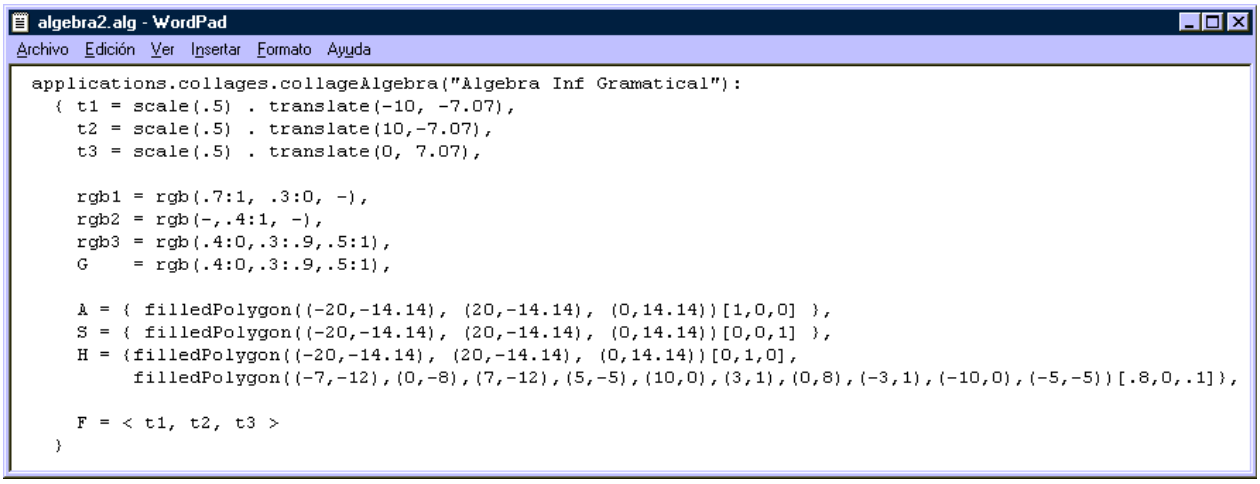


Figura 10.- *Álgebra collage asociada a la gramática inferida.*

Al elemento H de la gramática le hemos asociado la forma gráfica básica que es un triángulo inicialmente de color verde, lo mismo sucede con los elementos S y A pero con color inicial rojo y azul respectivamente. Los elementos propios del álgebra t_1, t_2 y t_3 realizan las transformaciones geométricas en 2 dimensiones. Se define que el elemento F está compuesto por las

transformaciones t_1, t_2 y t_3, estas serán aplicados en cada inferencia a los elementos que componen F en la gramática. Por ultimo se defina una transformación de color para el elemento G de la gramática En la Fig.11 se muestra el proceso generativo de la gramática inferida mediante la representación del árbol de derivación y el álgebra collage. En la primera de ellas (esquina superior izquierda) podemos ver la representación del símbolo inicial **S**. En la segunda (a la derecha) se visualiza un árbol primitivo por la aplicación de la producción **S → F[S,S,S]** donde, el triángulo inferior-izquierdo corresponde al hijo izquierdo de F en el árbol de derivación, el triángulo inferior-derecho corresponde al hijo medio de F y el triángulo superior con el hijo derecho de F. En la tercera(esquina superior derecha) vemos que la rama izquierda de la raíz F del árbol tiene como hijo a H por la producción **S → H**, H se definió en la gramática como elemento terminal de rango 0, por lo que ya no se sigue ramificando, no así con los hijos medio F y derecho G. Fig. 12 muestra el proceso tras varias iteraciones.

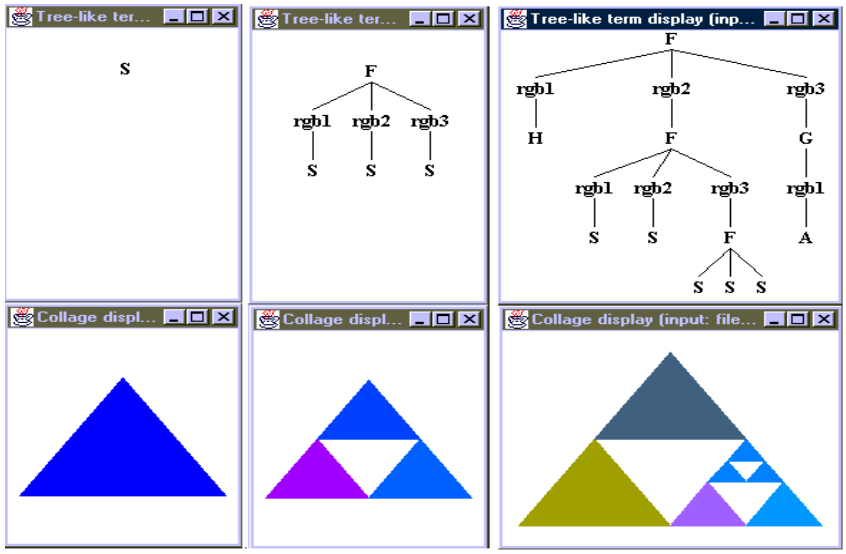


Fig. 11 *Visualización de un proceso generativo de árboles*

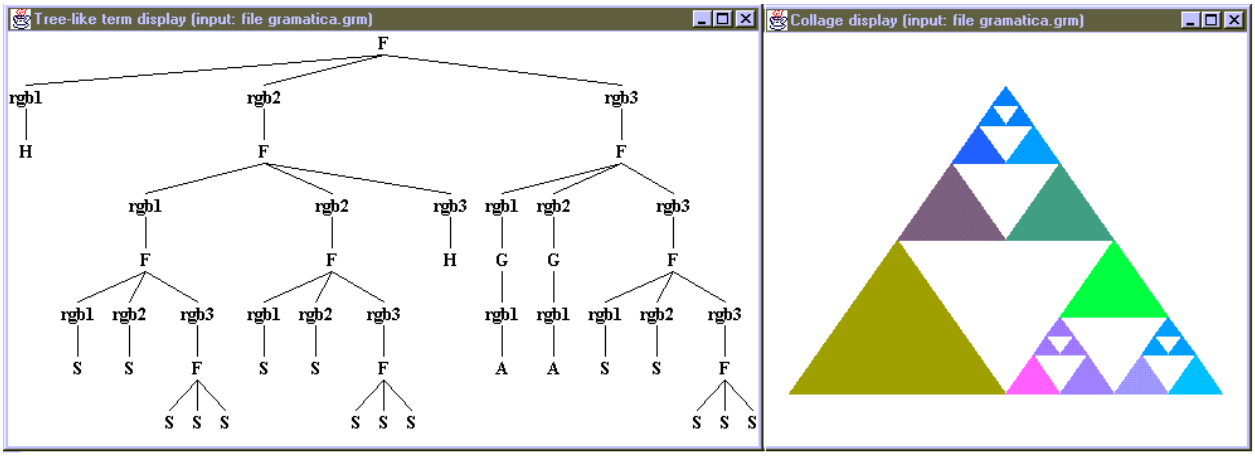


Fig. 12 *Proceso de inferencia tras varias iteraciones*.

3.6 Transductores

Notar que el proceso de inferencia no ha considerado transductores sino que hemos aplicado solamente las gramáticas de árboles y la interpretación vía álgebras Collage, sin embargo el proceso de inferencia es posible de generalizarlo si consideramos las propiedades que posee TreeBag en donde hemos utilizado en el proceso de derivación de árboles un transductor intermedio λ y un álgebra Collage arbitraria, generándose lo que muestra la Fig. 13, en donde a) muestra la visualización tras la interpretación de la regla de producción **S → F[S,S,S]** de la gramática g de la sección 3.5, en b) la transformación de la regla anterior por el transductor λ usando la transformación **P[F[x1,x2,x3]] → st2[P[x1],P[x3]]** descrita por la flecha, en c) se interpreta *cuadro* por un álgebra collage.

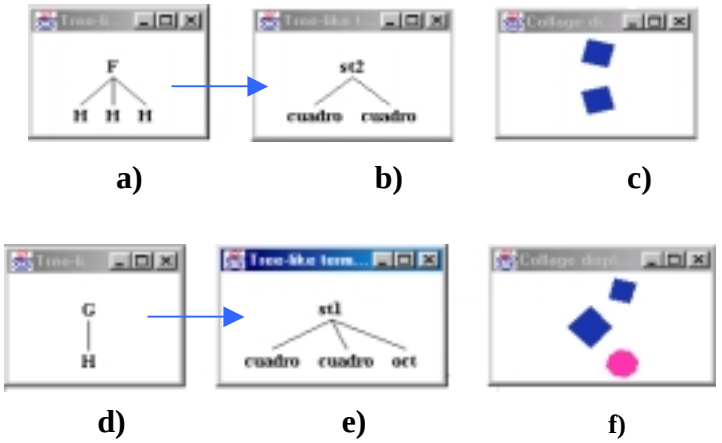


Fig. 13 *Transformación e Interpretación de un árbol.*

Análogamente, d) considera la producción **S → G[A]**, en e) se transforma mediante la regla **P[G[x1]] → st1[P[x1],P[x1],P[x1]]** descrita por la flecha en un árbol con 3 nodos hijos, y en f) se muestra la visualización del árbol a través de un álgebra collage arbitraria. Finalmente, luego de varias iteraciones se transforma la forma pictórica original (similar a la mostrada en Fig. 12), en otra forma pictórica, pero esta afectada por el transductor λ, según muestra la Fig. 14.

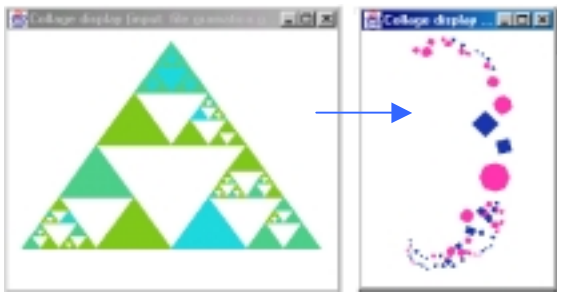


Fig. 14 *Transformación del triángulo de Sierpinsky mediante el transductor λ.*

4 Conclusión

En este trabajo se muestran las aplicaciones preliminares que pueden surgir con la herramienta TreeBag, basada en gramáticas y transductores de árboles. Esta muestra marca solo el comienzo de nuestro interés, pues estamos realizando la implementación del algoritmo de inferencia propuesto en [AJR02], el cuál sea considerado más que un módulo de inferencia para TreeBag, sino como un sistema modular de inferencia independiente. En este mismo sentido nuestro interés se centra en construir nuevos ejemplos, álgebras y gramáticas que sean de interés.

Referencias

[Po98] Jörg Poswig:Visuelle Programmierung, Addison-Wesley , 1998.

[Bar88] Michael Barnsley. *Fractals Everywhere*, Academic Press, Boston, 1988.

[Dre00] Frank Drewes. *Tree-Based Picture Generation*. Theo.Comp. Sc. 246: 1-51, 2000.

[Dre96] Frank Drewes. *Computation by tree transductions*. Doctoral dissertation, University of Bremen, Germany, 1996.

[FV98] Zoltán Fülöp, Heiko Vogler. *Syntax-Directed Semantics:Formal Models Based on Tree Transducers*. Springer, 1998.

[GS97] F. Gécseg, M. Steinby. *Tree Languages*. In G. Rozenberg and A. Salomaa, editores, Handbook of Formal Languages. Vol. III: Beyond Words, Cap.I, pag. 1-68. Springer, 1997.

[HK91]A. Habel, H.-J. Kreowski. *Collage Grammars*, Lect. Not. Comp. Sci. 532, 411-429, 1991.

[PJS92] Heinz-Otto Peitgen, Hartmut Jürgens y Dietmar Saupe. *Chaos and Fractals*. New Frontiers of Science. Springer-Verlag, New York, 1992.

[AJR02] M. Aguirre, E. Jeltsch y G. Rosales. Inferencia gramatical basada en gramáticas de árboles regulares con rango. Informe Técnico DIULS, 2002.

[PL90] P. Prusinkiewicz , A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer-Verlag, New York, 1990.

(W1) http://www.informatik.uni-bremen.de/theorie/treebag/(TreeBag)

(W2)http://www.uni-paderborn.de/fachbereich/AG/schaefer/ag_dt/PG/Fujaba/fujaba.html(Fujaba)

(W3) http://mindstorms.lego.com (LEGOMind)

(W4) http://www.informatik.uni-bremen.de/theorie/cs/ (CollageSystem)

(W5) http://www.cpsc.ucalgary.ca/Redirect/bmv/lstudio(L-System)

(W6) http://tfs.cs.tu-berlin.de/agg(AGG)