ÁRBOLES ARRAIGADOS. APLICACIONES.

Nancy Alonso, Elisa S. Oliva
Facultad de Ciencias Exactas, F. y Naturales. Universidad Nacional de San Juan. Argentina.
nalonso@unsj-cuim.edu, eoliva@iinfo.unsj.edu.ar

Resumen

Los árboles, diagramas arborescentes o arborigramas, constituyen una subclase de los grafos, importantes en el estudio de estructuras de datos, ordenaciones y codificación. En computación son de gran ayuda en la organización de datos y para visualizar relaciones. Este trabajo aborda aplicaciones de los árboles arraigados, que se trabajan en la asignatura Matemática Discreta de las Licenciaturas en Sistemas de Información y Ciencias de la Computación. Se muestran árboles que describen expresiones algebraicas, expresiones lógicas, árboles en la codificación, de búsqueda, árboles de alcanzabilidad en Redes de Petri y árboles que permiten determinar símbolos accesibles y/o generadores en una Gramática.

Fundamentación

Los avances tecnológicos de los últimos años y el proceso de acreditación de carreras de grado han producido cambios en la currícula de las Licenciaturas en Ciencias de la Computación y en Sistemas en Información, apoyando el desarrollo de muchos temas de aplicación, entre los que se encuentran los Métodos Discretos.

La asignatura Matemática Discreta tiene como objetivo fundamental proporcionar contenidos matemáticos de carácter eminentemente discreto, los cuales permiten al estudiante comprender y manipular estructuras finitas, tales como los grafos y sus aplicaciones, fundamentales en Teoría de la Computación.

En esta propuesta se muestran árboles que describen expresiones algebraicas, expresiones lógicas, árboles en la codificación, de búsqueda, árboles de alcanzabilidad en Redes de Petri y árboles que permiten determinar símbolos accesibles y/o generadores en una Gramática.

Metodología

La metodología aplicada, está enfocada en que los alumnos aprendan a desarrollar su capacidad lógico-matemática y lograr una participación activa en cada uno de los procesos de enseñanza-aprendizaje.

Los contenidos teóricos se desarrollan con métodos inductivos, deductivos, de análisis, síntesis y analógicos. Se emplean formas expositivas verbales, interrogativas y dialogadas y

se incentiva la investigación, guiando al alumno en la búsqueda de temas relacionados con la asignatura. Las Guías Prácticas se trabajan en forma grupal bajo la supervisión del equipo docente y cada Guía se acompaña de un anexo con ejercicios propuestos.

Contenidos teóricos de la propuesta

Árboles Arraigados

Un árbol arraigado de raíz v_0 es un grafo dirigido $G=(V,R)$ en el cual existe un vértice v_0, denominado raíz del árbol, que no admite trayectorias en sí mismo y para cada vértice $v_i \neq v_0$, existe una única trayectoria $T_{v_o v_i}$ en el grafo G.

Si $(v_i, v_j) \in R$, v_j es un vástago o hijo del vértice v_i.

Elementos de un árbol arraigado
Hoja (vértice terminal): Vértice que no tiene vástagos.
Vértice interno: Vértice que no es raíz ni hoja.
Nivel 0: Conjunto unitario cuyo elemento es la raíz del árbol.
Nivel k (k>0): Conjunto de vástagos de los vértices del nivel k−1
Altura de G: máx {long. $T_{v_o v_i}$, $v_i \in V$} (cantidad de niveles no nulos).

Tipos de árboles
Etiquetado: Tienen vértices y/o arcos con etiquetas o nombres.
n-ario: Los vértices, que no son hojas, tienen a lo sumo n vástagos.
n-ario completo: Todos los vértices, que no son hojas, tienen n vástagos.
Ordenado: Si se ha prefijado un orden en su conjunto de vértices, por niveles o por ramas.

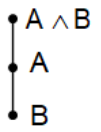
Árboles de expresiones algebraicas. Reglas de construcción de un árbol algebraico.

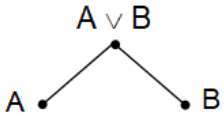
El árbol que representa una expresión algebraica tiene raíz y vértices internos etiquetados con operaciones del álgebra y hojas etiquetadas con elementos del portador del álgebra.

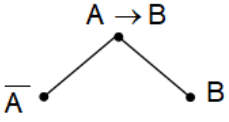
1.- La raíz del árbol se etiqueta con la operación principal de la expresión.
2.- Si la etiqueta de un nodo es una operación binaria, se dibujan dos vástagos.
Las etiquetas de los vástagos depende de los miembros que componen la operación, si es una expresión algebraica se etiqueta con la operación principal, si es un elemento del portador del álgebra, se etiqueta con ese elemento (hoja del árbol).
3.- Si la etiqueta de un nodo es una operación unitaria se dibuja sólo un vástago.
4.- Finaliza el proceso cuando todas las ramas finalizan en vértices etiquetados con un elemento del portador del álgebra, los cuales serán las hojas del árbol.

Arboles Lógicos. Reglas de Construcción de un Árbol Lógico.

El árbol lógico que representa una expresión lógica, tiene raíz etiquetada con la expresión lógica y vértices internos etiquetados con expresiones lógicas o variables lógicas, sus hojas se etiquetan con variables lógicas.

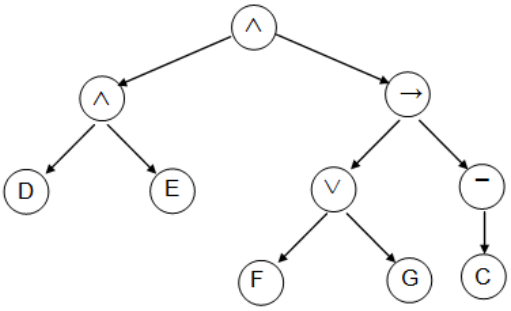




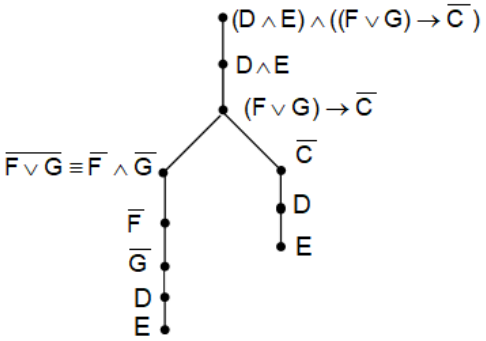


Ejemplo

Árbol algebraico de $(D \wedge E) \wedge ((F \vee G) \rightarrow \overline{C})$

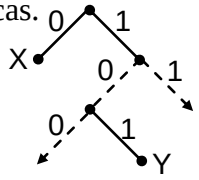


Árbol lógico de $(D \wedge E) \wedge ((F \vee G) \rightarrow \overline{C})$



Aplicación de árboles arraigados en la codificación. Código de Huffman.

El código de Huffman representa los caracteres con cadenas de 0 y/o 1 de longitud variable, tiene prefijado un alfabeto cuyos símbolos están codificados con estas cadenas. Cada código de Huffman se define por medio de un árbol arraigado binario completo, ordenado, con hojas etiquetadas con símbolos del alfabeto y las siguientes características.



- La raíz y vértices internos no están etiquetados.
- Las hojas están etiquetas con símbolos del alfabeto.
- Cada vértice, no hoja, tiene dos vástagos, el arco izquierdo etiquetado con 0 y el arco derecho etiquetado con 1.

Cuando en un árbol de Huffman leemos una cadena de 0s y 1s determinamos una palabra, también el proceso inverso, dado un árbol de Huffman y una palabra, determinamos la cadena que la codifica.

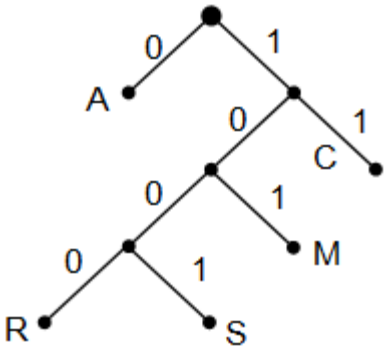
Ejemplo

Dado el árbol arraigado del código de Huffman con alfabeto X={A,C,M,R,S}.
Determinamos la palabra que codifica la cadena 11010010
y el código para la palabra ARMAS.

* Determinamos las cadenas que codifican los símbolos del alfabeto

A: 0 C: 11 M: 101 R: 1000 S: 1001

La palabra que codifica 11010010 es CASA, 11 0 1001 0
C A S A



* Para la codificación de ARMAS, reemplazamos cada uno de sus símbolos por la cadena que lo codifica: $\underset{0}{A}\ \underset{1000}{R}\ \underset{101}{M}\ \underset{0}{A}\ \underset{1001}{S}$, ARMAS se codifica con 0100010101001.

Árboles de búsqueda en un grafo. Métodos de búsqueda.

Un problema general, en la teoría de árboles arraigados, implica la exploración de un grafo en busca de cierto vértice que tenga asociada alguna propiedad o datos específicos de lo que modela el grafo. Suponemos grafos conexos sencillos y realizaremos la búsqueda por medio de un árbol arraigado. Existen dos formas de búsqueda, la búsqueda en amplitud (BFS) y la búsqueda en profundidad (DFS).

Búsqueda en amplitud (BFS)

La búsqueda en amplitud, o de expansión, se puede utilizar para determinar la distancia más corta, mínimo número de arcos que hay que recorrer para pasar, del vértice inicial al vértice seleccionado y se realiza por niveles del árbol.

Procedimiento

Se selecciona un vértice inicial, raíz del árbol de búsqueda. Se visitan los vértices a distancia 1 del inicial (adyacentes), en el orden de la lista y se escribe una secuencia de vértices visitados.

Se visitan los vértices a distancia 2 del inicial y se visitan los vértices adyacentes a éstos, en el orden de la lista, y que no fueron visitados…… Se visitan los vértices a distancia k-1 del inicial que no fueron visitados y se visitan los vértices adyacentes a éstos, en el orden de la lista, y que no fueron visitados. El algoritmo finaliza cuando se visitan todos los vértices del grafo.

Notar que en cada paso de búsqueda, se seleccionan caminos más cortos desde la raíz, es conveniente agregar una etiqueta a los arcos que indique el orden de lectura de la búsqueda.

Búsqueda en profundidad (DFS)

Esta búsqueda se realiza por ramas del árbol.

Procedimiento

Se marca el vértice inicial y se visita y marca su primer adyacente. Se visita el primer adyacente del marcado en el nivel anterior.

- si no fue visitado, se marca y se pasa a su adyacente
- si ya fue visitado, se vuelve al adyacente marcado en el nivel anterior y se repite el proceso.
- si se encuentra un vértice que no tiene adyacentes no visitados, se regresa al vértice del nivel anterior que tenga vértices adyacentes no marcados.

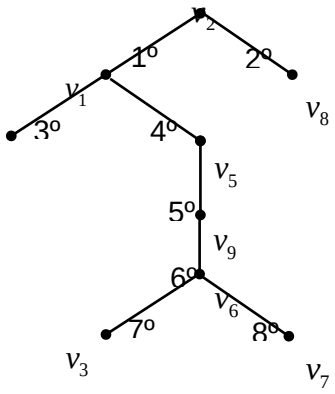
El algoritmo finaliza cuando se visitan a todos los vértices del grafo.

El árbol de Búsqueda en profundidad (DFS), con raíz en el vértice inicial, se construye con los pasos de búsqueda, sus arcos se etiquetan con el orden de lectura de la búsqueda.

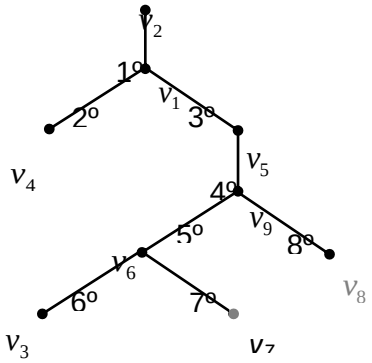
Ejemplo

***Fijamos como vértice inicial* v_2.**

Árbol de búsqueda en amplitud



Árbol de búsqueda en profundidad



Para G_1	
v_1	v_2 v_4 v_5
v_2	v_1 v_8
v_3	v_4
v_4	v_1
v_5	v_2 v_4 v_9
v_6	v_3 v_7
v_7	v_6
v_8	v_2
v_9	v_5 v_6 v_8

Visualmente, podemos concluir que:

Secuencia de vértices visitados en la búsqueda en amplitud es $v_2\ v_1\ v_8\ v_4\ v_5\ v_9\ v_6\ v_3\ v_7$.

Secuencia de vértices visitados en la búsqueda en profundidad es $v_2\ v_1\ v_4 v_5 v_9 v_6 v_3 v_7 v_8$.

Aplicación de Árboles Arraigados en Redes de Petri Marcadas. Árbol de Alcanzabilidad.

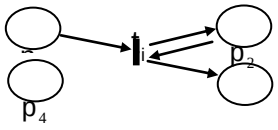
Una Red de Petri es un grafo bipartito $G=(P\cup T,R,\mu)$, con relación $R\subseteq(P\times T)\cup(T\times P)$

Cada elemento $p_i\in P$ es un "lugar" de la Red y se representa con un círculo.

Cada elemento $t_i\in T$ es una "transición" y se representa con una pequeña barra. $\mu: P\to N\cup\{0\}/\ \mu(p_i)=n_i$, n_i "número de fichas del lugar p_i"

Cada estado de la Red es una n-úpla $M=(n_1,n_2,...,n_n)$.

Con M_0 indicamos el estado inicial de la Red.



p_1 *y* p_2 , son lugares de entrada de la transición t_i
p_2 y p_3 son lugares de salida de la transición t_i
p_2 es lugar de entrada y salida de t_i
p_4 no es lugar de entrada ni de salida de t_i

Las Redes de Petri se utilizan para modelar Sistemas, cada transición modela un suceso o evento del Sistema y sus lugares de entrada modelan las condiciones que se exigen para que se produzca ese suceso o evento.

En la modelación de un Sistema, es importante saber qué condiciones se deben cumplir para que un determinado evento pueda suceder. En una Red de Petri que modela un problema se necesita saber cuándo una transición puede actuar (está habilitada).

Habilitación de transiciones

En una Red de Petri marcada, $G=(P\cup T,R,\mu)$, una transición t_i está habilitada si todo lugar de entrada tiene al menos una ficha. En una Red con n lugares, la habilitación de una transición t_i se indica con la n-úpla:

$$H_{t_i}=(i_1,i_2,...,i_h,...,i_n),\quad i_h=\begin{cases}1\ si\ \ p_h\ es\ lugar\ de\ entrada\ de\ t_i\\ 0\ si\ \ p_h\ no\ es\ lugar\ de\ entrada\ de\ t_i\end{cases}$$

Cambios que produce una transición t_i

Una vez producido un evento de un Sistema, sus condiciones pueden cambiar.

Si una Red de Petri marcada, está en un estado $M=(n_1,n_2,...,n_n)$, y se dispara una transición t_i, el cambio que produce se indica con la n-úpla:

$$C_{t_i} = (i_1, i_2, ..., i_k, ..., i_n), \qquad i_k = \begin{cases} 1 \ si \ \ p_k \ lugar \ de \ salida \ de \ \ t_i \\ -1 \ \ si \ \ p_k \ lugar \ de \ entrada \ de \ \ t_i \\ 0 \ \ si \ \ p_k \ no \ \ es \ lugar \ de \ entrada \ \ ni \ \ de \ \ salida \ de \ t_i \\ 0 \ \ si \ \ p_k \ \ es \ \ lugar \ de \ entrada \ \ y \ \ salida \ \ de \ \ t_i \end{cases}$$

Cuando se dispara la transición t_i, la Red pasa a un nuevo estado $M' = M + C_{t_i}$

Estados alcanzables

Un estado M' es alcanzable, desde el estado M, si existe una sucesión de transiciones $t_{i_1}, t_{i_2}, ..., t_{i_k}$ que al ser disparadas, en ese orden, dejan a la Red en el estado M', lo que se simboliza $M' = M(t_{i_1} t_{i_2} \ ... t_{i_k})$.

Estado terminal

Un estado de la Red, es terminal (**T**) o "sin salida" en la Red, si en él no existen transiciones habilitadas.

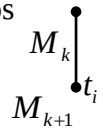
Estado duplicado

Un estado de la Red, es duplicado (**D**) si fue alcanzado anteriormente, por otra sucesión de transiciones.

Árbol de alcanzabilidad

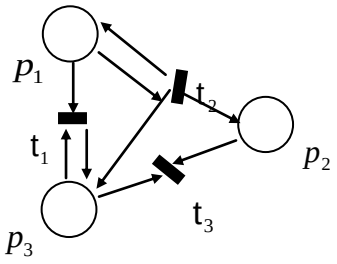
Otra aplicación de árboles arraigados es analizar la estructura dinámica de una Red de Petri. El árbol de alcanzabilidad de una Red de Petri informa cuáles estados son alcanzables, duplicados, terminales, si es una Red viva, acotada, etc.

Reglas de construcción de un árbol de alcanzabilidad

- La raíz del árbol se etiqueta con el estado inicial M_0.
- Los vértices internos y las hojas se etiquetan con estados alcanzables.
- Cada estado alcanzable M_k tendrá tantos vástagos como transiciones estén habilitadas en M_k.
- Si dibujamos  interpretamos $M_{k+1} = M_k + C_{t_i}$

Ejemplo

Árbol de alcanzabilidad de la siguiente Red de Petri con M_0=(1,1,1).

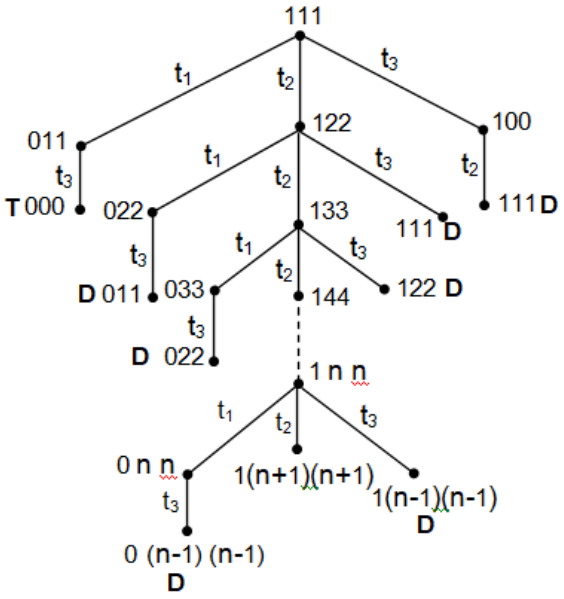


Habilitación y cambio de transiciones:

$H_{t_1} = (1,0,1)$ $H_{t_2} = (1,0,0)$ $H_{t_3} = (0,1,1)$

$C_{t_1} = (-1,0,0)$ $C_{t_2} = (0,1,1)$ $C_{t_3} = (0,-1,-1)$

Árbol de alcanzabilidad de la Red de Petri



Visualmente, podemos concluir:

La red no es acotada.

La red no es viva (000 es estado terminal).

Estados alcanzables:

$(1,0,0) = M_0(t_3)$

$(1,k,k) = M_0(t_2^{k-1}) = M_0(t_2^k t_3)$ (k>0)

$(0,k,k) = M_0(t_2^{k-1} t_1) = M_0(t_2^k t_1 t_3)$ (k>0)

$(0,0,0) = M_0(t_1 t_3)$

Aplicación de los árboles arraigados en Gramáticas

Una gramática $G=(N,\Sigma,S,P)$, es una estructura algebraica formada por cuatro elementos.

N: Conjunto finito de símbolos no terminales

Σ: Conjunto finito de símbolos terminales (alfabeto del lenguaje)

S: Símbolo no terminal denominado símbolo inicial o axioma $(S \in N)$.

P: Relación de producción, definida en $(N \cup \Sigma)^* N (N \cup \Sigma)^* x (N \cup \Sigma)^*$

$(\alpha,\alpha') \in P$ determina una "regla de producción" que simbolizaremos $\alpha\,\alpha\ \alpha'$ y se lee "la cadena α produce la cadena α'" (α se denomina "cabeza" y α' "cuerpo" de la regla).

Relación deriva

La relación deriva se simboliza con $\underset{k)}{\Rightarrow}$ y está definida en $(N \cup \Sigma)^*$.

$\beta\delta\sigma \underset{k)}{\Rightarrow} \beta\mu\sigma$ si y sólo si existe la regla de producción k) $\delta \alpha \mu$

$\alpha \underset{k)}{\Rightarrow} \alpha'$, se lee "la cadena α' se deriva de la cadena α por la regla k) ".

Si α' se deriva de α aplicando reglas $k_1), k_2), \ldots, k_n)$ escribimos $\alpha \underset{k_1)\ldots k_n)}{\Rightarrow^*} \alpha'$.

Símbolos accesibles y generadores de una gramática

Un símbolo $X \in (N \cup \Sigma)$ es alcanzable si existe una derivación $S \Rightarrow^* \alpha X \beta$

Un símbolo $X \in N$ es generador si existe una derivación $X \Rightarrow^* w, \quad w \in \Sigma^*$.

Un símbolo $X \in N$ (no terminal) es útil si es alcanzable y generador

El axioma de un lenguaje no vacío siempre se lo considera "útil", es generador y aunque puede ser o no alcanzable.

Árbol arraigado para determinar símbolos alcanzables.

- La raíz del árbol se etiqueta con S, axioma de la Gramática.
- La raíz tendrá tantos hijos como reglas de producción tengan como cabeza el axioma.
- Cada vértice interno tendrá tantos hijos como reglas de producción puedan aplicarse a través de la relación deriva.
- Un vértice será hoja cuando no sea posible aplicar la relación deriva.
- Todo símbolo perteneciente a una cadena vértice del árbol, será alcanzable.
- Todo símbolo que no pertenece a cadena alguna de los vértices del árbol, es no alcanzable.

Las hojas de estos árboles de raíz S, con cadenas que contienen sólo símbolos terminales, son cadenas pertenecientes al lenguaje de la gramática.

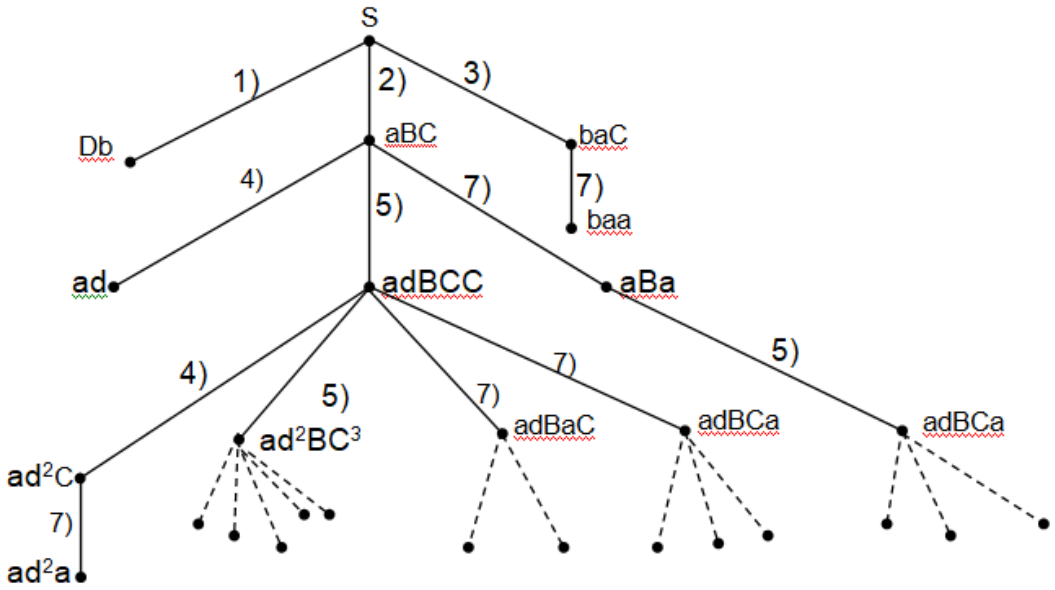
Árbol arraigado para determinar símbolos generadores.

- La raíz del árbol se etiqueta con el símbolo no terminal *X*.
- La raíz tendrá tantos hijos como reglas de producción tengan como cabeza el símbolo *X*.
- Cada vértice interno tendrá tantos hijos como reglas de producción puedan aplicarse a través de la relación deriva.
- Un vértice será hoja cuando no sea posible aplicar la relación deriva.
- Si el árbol tiene, por lo menos, una hoja etiquetada con una cadena sólo de símbolos terminales, *X* es generador.
- Si no es posible, encontrar una hoja cuya cadena contenga sólo símbolos terminales, X es no generador.

Ejemplo

G= (N,Σ,S,P), N={A,B,C,D,E,S}, $\Sigma = \{a,b,c,d\}$ y reglas de producción: 1)S α Db; 2) S α aBC; 3)S α baC; 4) BC α d; 5)B α dBC; 6)AaS α da; 7) C α a; 8)E α Ba

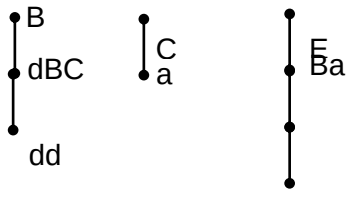
Árbol arraigado para determinar símbolos alcanzables de G.



Visualmente, podemos concluir:

Símbolos alcanzables: a, b, d, B, C y D.
Símbolos No alcanzables: c, A, E y S.
Cadenas del lenguaje generado por G: ba^2, ad, ad^2a, ad^3a^2, ad^ka^{k-1} k>0…

Árbol arraigado para determinar símbolos generadores de G.



Visualmente, podemos concluir:

Símbolos generadores: B, C, E y S.
Los símbolos No terminales A y D, no son cabeza de regla, no se puede aplicar la relación deriva, por lo tanto A y D son No generadores.

Guía Práctica

Ejercicio 1.- Realizar el árbol algebraico de la expresión $\left(\left(2+6x\right)\div\left(4y^3t\right)\right)-\left(z-4\right)$. Indicar raíz, hojas, niveles y altura del árbol. Clasificar el árbol.

Ejercicio 2.- Realizar el árbol algebraico y el árbol lógico de la expresión lógica $(A \to B) \to \left(\left(R \wedge \left(\overline{C \wedge \overline{B}}\right)\right) \vee \left(E \wedge \left(\overline{S} \to \overline{F}\right)\right)\right)$. Indicar hojas, vértices internos y altura del árbol.

Ejercicio 3.- Construir el árbol que define un código de Huffman de alfabeto {A,E,M,R,S} con codificaciones: A:001 ; E:000 ; M:010 ; R:011 ; S:11
Codificar la palabra MAREA. Decodificar las cadenas 11000001 y 01100001000111.

Ejercicio 4.- En función de la lista de adyacencia del grafo G_1 presentada en los contenidos teóricos, realice los árboles de búsqueda en amplitud y búsqueda en profundidad, considerando vértice inicial v_5 y v_9, respectivamente.

Ejercicio 5.- En la Red de Petri de la base teórica realizar el árbol de alcanzabilidad considerando estado inicial *Mo*=(1,0,1). Indique si *M*=(1,5,6) es un estado alcanzable. Indique si se puede ejecutar la secuencia $(t_2)^7\ t_3$, si fuese posible, dar el estado alcanzado.

Ejercicio 6.- Si en la gramática de los contenidos teóricos elimina la reglas 1) y 3), analice en esta nueva gramática los símbolos alcanzables, no alcanzables, generadores y no generadores.

Conclusiones

- Conceptos básicos de la Teoría de Grafos, como son las trayectorias, permiten abordar, sin dificultad, el desarrollo de la teoría Árboles Arraigados y sus aplicaciones.
- Los grafos, en particular los árboles, a través de sus diagramas, posibilitan interpretar visualmente situaciones sencillas y estudiar su generalización a otras situaciones.
- Los árboles (diagramas arborescentes), constituyen una de las subclases más útiles de grafos, importantes en el estudio de estructuras de datos, ordenaciones, codificación y problemas de optimización.
- En las carreras Licenciaturas en Sistemas de Información y Ciencias de la Computación, estos diagramas, se utilizan con frecuencia, siendo gran ayuda en la organización de datos y en la visualización de relaciones.

Referencias bibliográficas

Abellanas, M. (1990). *Análisis de algoritmos y teoría de grafos*. Madrid: RA-MA.

Chillemi, A., Alonso, N. (2013). *Matemática discreta para informáticos*. Argentina: UNSJ

Cubero, E., Moreno, M., Salomon, R. (2007). *Teoría de Autómatas y Lenguajes Formales*. Madrid: Mc Graw Hill.

Grassman, W., Tremblay, J. (1997). *Matemática discreta y lógica*. España: Prentice Hall

Grimaldi, R (1998). *Matemáticas discreta y combinatoria*. México: Prentice Hall

Hopcroft J., Ullman J. (2008). *Introducción a la Teoría de Autómatas, Lenguajes y Computación*. México: CECSA

Johnsonbaugh,R. (1990). *Matemáticas Discretas*. México: Iberoamérica

Korfhage R. (1974). *Lógica y Algoritmos*. México: Limusa