Grafos de Intervalos

Problemas de Grafos y Tratabilidad Computacional

Definición

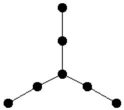
Un grafo $G = (V, E)$ es de intervalos si existe una familia $\mathcal{I} = \{I_1, \cdots, I_n\}$ sobre la recta real tal que G es el grafo intersección de $\mathcal{I}$, es decir que, $V = \{v_1, \cdots, v_n\}$ donde cada v_i corresponde al intervalo I_i y la arista $(v_i, v_j) \in E \Leftrightarrow i \neq j_\wedge$ $I_i \cap I_j \neq \emptyset$. En tal caso, $\mathcal{I}$ es un modelo o representación de intervalos de G.

Teorema

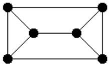
Dado G un grafo de intervalos.

- G es grafo cordal. (No es cierta la vuelta: T_2)
- $\overline{G}$ es grafo de comparabilidad. (No vale la vuelta: $\overline{C_6}$)

$T_2 = star_{2,2,2}$



$\overline{C_6}$



Lema

Sea $G = (V, E)$ un grafo que es C_4-free y de co-comparabilidad, Q_1, Q_2 dos cliques diferentes de G y F una orientación transitiva de $\overline{G}$.

1. Existe un arco en F con un extremo en Q_1 y el otro en Q_2.
2. Todas las no aristas $(v_i, v_j) \notin E$ $(v_i \neq v_j)$ que conectan Q_1 con Q_2 (poseen un extremo en Q_1 y el otro en Q_2) tienen la misma orientación ($Q_1 \to Q_2$ o $Q_2 \to Q_1$).

Otra definición

Dada una matriz de $\{0,1\}^{n\times m}$. Esta matriz cumple la propiedad de 1's consecutivos por columnas si es posible permutar sus filas de manera tal que para cada columna $j \in \{1,\cdots,m\}$, las posiciones que contienen 1's aparecen consecutivamente.

$$\underset{\mathbf{M}_1}{\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}} \rightarrow \underset{\mathbf{M}_2}{\begin{array}{c} 1 \\ 4 \\ 2 \\ 5 \\ 3 \end{array}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}} \qquad \underset{\mathbf{M}_3}{\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}$$

Teorema

Sea $G = (V, E)$ un grafo. Las siguientes afirmaciones son equivalentes.

1. G es un grafo de intervalos.
2. G no contiene a C_4 como subgrafos inducidos y $\overline{G}$ es grafo de comparabilidad.
3. La matriz clique M de G cumple la propiedad de 1's consecutivos por columnas que es lo mismo decir que se pueden ordenar linealmente los cliques de G de manera tal que para cada vértice $v \in V$, los cliques que lo contienen están en forma consecutiva en este orden lineal.

Corolario

Un grafo G es de intervalos si y sólo sí G es cordal y es de co-comparabilidad.

Algoritmo de reconocimiento

Dado un grafo G

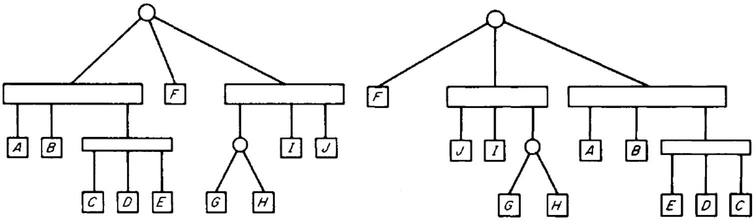
- Verificar si G es cordal ($O(m+n)$) y de co-comparabilidad ($O(MM)$).
- Verificar si G es cordal $O(m+n)$, generar su matriz clique M en el caso positivo y testear si M cumple la propiedad de 1's consecutivos por columnas.
  - ¿Hace falta generar la matriz clique porque ya cuesta $O(n^2)$ tener la matriz?
  - ¿Cuántas permutaciones de filas (cliques) habría que probar?
  - Usar PQ-tree.

PQ-tree

Dada una familia $\mathcal{F}$ de subconjuntos sobre un universo discreto y finito X. El problema es determinar si existe un orden lineal de los elemento de X de manera tal que para cada subconjunto $S \in \mathcal{F}$, los elementos de S aparecen en forma consecutiva en este orden. Se denota como $\Pi(\mathcal{F})$ el conjunto de ordenes lineales de los elementos de X que son soluciones del problema. Por ejemplo, si $\mathcal{F} = \{\{A, B, C\}, \{A, D\}\}$ entonces $\Pi(\mathcal{F}) = \{[D, A, B, C], [D, A, C, B], [C, B, A, D], [B, C, A, D]\}$. Booth y Luecker en 1975 inventaron una estructura de datos llamada PQ-tree para resolver este problema.

Veamos qué es un PQ-tree X.

- T es un árbol con una raíz definida si tiene algún nodo.
- Cada nodo hoja tiene una etiqueta distinta y sea X el conjunto de etiquetas de las hojas.
- Los nodos internos de T (es decir que no son hojas) son de 2 tipos: del tipo P (se dibujan con círculos) y del tipo Q (se dibujan con rectángulos).

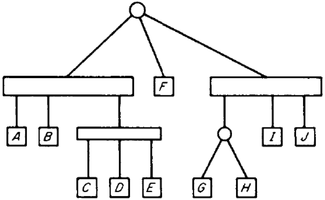


- La frontera de T (*frontera*(T)) es la permutación de X obtenida leyendo las etiquetas de las hojas de izquierda a derecha de T.
- 2 PQ-trees T y T' son equivalentes ($T \equiv T'$) si uno puede ser obtenido del otro aplicando una secuencia de las siguientes operaciones.
  1. Arbitramente permutar los hijos de un nodo del tipo P.
  2. Invertir el orden de los hijos de un nodo del tipo Q.
- $consistente(T) = \{frontera(T') \, / \, T' \equiv T\}$, es el subconjunto de permutaciones de X consistentes con respecto a T.

Teorema

- Para cada $\mathcal{F}$ sobre X existe un PQ-tree T tal que $\Pi(\mathcal{F}) = \mathit{consistente}(T)$.
- Para un PQ-tree T donde $\mathit{frontera}(T)$ es una permutación de X, existe $\mathcal{F}$ sobre X tal que $\Pi(\mathcal{F}) = \mathit{consistente}(T)$.

Como consecuencia, las clases de permutaciones consistentes de los PQ-tree forman un retículo. El árbol nulo T_0 que no tiene ningún nodo y $\mathit{consistente}(T) = \emptyset$. El árbol universal tiene un solo nodo interno (la raíz) que es de tipo P, tiene tanta hojas como elementos de X. Por lo tanto cualquier permutación de X es consistente a este árbol.

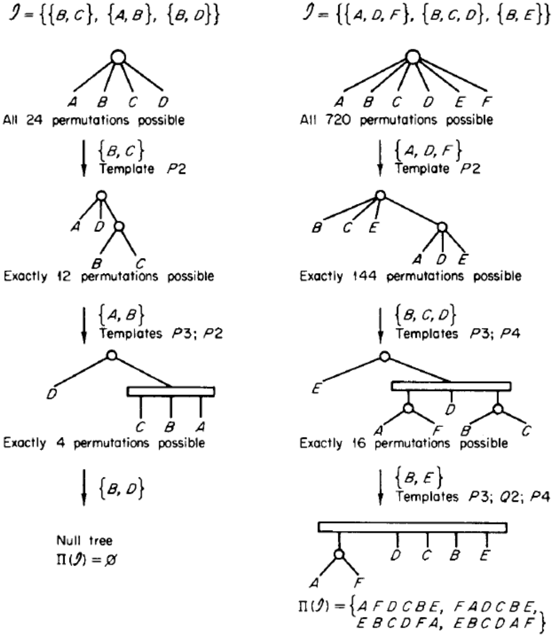


$\mathcal{F} =$
$\{\{A, B\}, \{C, D\}, \{D, E\}, \{B, C, D, E\}, \{I, J\}, \{G, H\}, \{G, H, I\}\}$

Algoritmo

Función Consecutivo($\mathcal{F}, X$)
  $T \leftarrow$ árbol universal con $|X|$ hojas
  Para cada $S \in \mathcal{F}$ hacer
    $T \leftarrow REDUCE(T, S)$
    Si T es nulo entonces devolver T
  Devolver T

- La rutina *REDUCE* utiliza 11 templates de pattern matching para aplicarlos en el árbol actual T y cómo se debe realizar los reemplazos en cada pattern. La búsqueda se hace desde abajo hacia arriba en T.
- El espacio requerido es $|X|$.
- La complejidad es $O(\mathcal{F} + |X| + \sum_{S \in \mathcal{F}} |S|)$



𝒥 = {{B, C}, {A, B}, {B, D}}
A B C D
All 24 permutations possible
{B, C}
Template P2
A D B C
Exactly 12 permutations possible
{A, B}
Templates P3; P2
D C B A
Exactly 4 permutations possible
{B, D}
Null tree
Π(𝒥) = ∅
𝒥 = {{A, D, F}, {B, C, D}, {B, E}}
A B C D E F
All 720 permutations possible
{A, D, F}
Template P2
B C E A D F
Exactly 144 permutations possible
{B, C, D}
Templates P3; P4
E A F D B C
Exactly 16 permutations possible
{B, E}
Templates P3; Q2; P4
A F D C B E
Π(𝒥) = {A F D C B E, F A D C B E, E B C D F A, E B C D A F}