UFPA

Grafos Direcionados (Dígrafos)

Fechamento Transitivo

Definição

O **Fechamento Transitivo** de um dígrafo D é um dígrafo D^* com os mesmos vértices de D. No entanto, existe um arco de a para b em D^* se e somente se existir um caminho de a para b.

Fechamento Transitivo

Definição

O **Fechamento Transitivo** de um dígrafo D é um dígrafo D^* com os mesmos vértices de D. No entanto, existe um arco de a para b em D^* se e somente se existir um caminho de a para b.

Ou seja, se existe um arco do vértice a para o vértice b e um arco de b para c em D, então existe um arco de a para c em D^*.

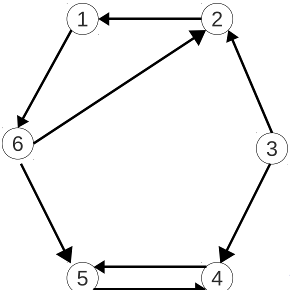
Fechamento Transitivo

Definição

O **Fechamento Transitivo** de um dígrafo D é um dígrafo D^* com os mesmos vértices de D. No entanto, existe um arco de a para b em D^* se e somente se existir um caminho de a para b.

Ou seja, se existe um arco do vértice a para o vértice b e um arco de b para c em D, então existe um arco de a para c em D^*.

Qual o fechamento transitivo do seguinte dígrafo ?



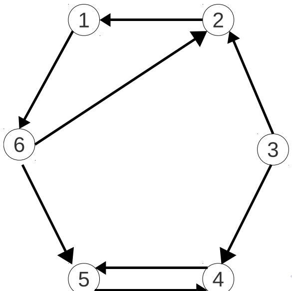
Fechamento Transitivo

Definição

O **Fechamento Transitivo** de um dígrafo D é um dígrafo D^* com os mesmos vértices de D. No entanto, existe um arco de a para b em D^* se e somente se existir um caminho de a para b.

Ou seja, se existe um arco do vértice a para o vértice b e um arco de b para c em D, então existe um arco de a para c em D^*.

Qual o fechamento transitivo do seguinte dígrafo ?

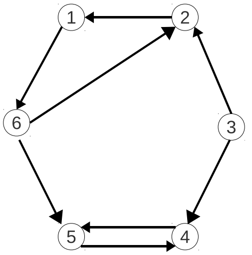


Encontrando o Fechamento Transitivo

1. **Fecho transitivo com matrizes Booleanas**

Matriz Booleana

Matriz com entradas binárias (0 ou 1)



$$G = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Encontrando o Fechamento Transitivo

Produto de Matrizes Booleana

- Dada duas matrizes $(n \times n)$ A e B, encontrar $C = A * B$.
- No produto de *matrizes convencionais* as operações são: $*$ e $+$
- No produto de *matrizes booleanas* temos: $\wedge$ e $\vee$
- Produto de Matrizes Booleanas $C = A * B$:
  $c_{ij} = \vee_{k=1}^{m}(a_{ik} \wedge b_{kj})$

Encontrando o Fechamento Transitivo

Produto de Matrizes Booleana

- Dada duas matrizes $(n \times n)$ A e B, encontrar $C = A * B$.
- No produto de *matrizes convencionais* as operações são: $*$ e $+$
- No produto de *matrizes booleanas* temos: $\wedge$ e $\vee$
- Produto de Matrizes Booleanas $C = A * B$:
  $c_{ij} = \vee_{k=1}^{m}(a_{ik} \wedge b_{kj})$

- Se A é uma matriz de adjacências de um dígrafo, podemos ter $C = A * A$, ou seja, A^2
- Os arcos em A^2 correspondem a arcos de tamanho 2 em A.
- Os arcos em A^3 correspondem a arcos de tamanho 3 em A.
- ...

Encontrando o Fechamento Transitivo

Produto de duas Matrizes $n \times n$*, ou seja,* $C = A * B$

Seja n o número de vértices, i linha da matriz A, j coluna da matriz B.

para $i = 0$ **até** $i < n$ **faça**
  para $j = 0$ **até** $j < n$ **faça**
    para $l = 0$, $C[i][j] = 0$ **até** $l < n$ **faça**
      $C[i][j] += A[i][l] * [l][j]$
    fim
  fim
fim

Encontrando o Fechamento Transitivo

Multiplicação de Matrizes Booleanas $A \times A$

Seja n o número de vértices, i linha da matriz A, j coluna da matriz B.

para $i = 0$ **até** $i < n$ **faça**
  para $j = 0$ **até** $j < n$ **faça**
    para $l = 0$, $C[i][j] = 0$ **até** $l < n$ **faça**
      se $A[i][l] = 1$ e $B[l][j] = 1$ **então**
        $C[i][j] = 1$
      fim
    fim
  fim
fim

Encontrando o Fechamento Transitivo

Fecho Transitivo

Seja n o número de vértices, o fecho transitivo de um dígrafo pode ser calculado através da multiplicação de matrizes de adjacência A (i.e. $A * A * \ldots$), adicionando-se laços e calculando caminhos de A^n no máximo.

O **fecho transitivo** é dado por: $R = A \vee A^2 \vee \ldots \vee A^n$

Encontrando o Fechamento Transitivo

Fecho Transitivo

Seja n o número de vértices, o fecho transitivo de um dígrafo pode ser calculado através da multiplicação de matrizes de adjacência A (i.e. $A * A * \ldots$), adicionando-se laços e calculando caminhos de A^n no máximo.

O **fecho transitivo** é dado por: $R = A \vee A^2 \vee \ldots \vee A^n$

Por que não calcular $A^{n+1}, A^{n+2}, A^{n+3}, \ldots$?

Encontrando o Fechamento Transitivo

Fecho Transitivo

Seja n o número de vértices, o fecho transitivo de um dígrafo pode ser calculado através da multiplicação de matrizes de adjacência A (i.e. $A * A * \ldots$), adicionando-se laços e calculando caminhos de A^n no máximo.

O **fecho transitivo** é dado por: $R = A \vee A^2 \vee \ldots \vee A^n$

Por que não calcular $A^{n+1}, A^{n+2}, A^{n+3}, \ldots$ **?**

Princípio da casas de pombo !!!

Encontrando o Fechamento Transitivo

Fecho Transitivo

Seja n o número de vértices, o fecho transitivo de um dígrafo pode ser calculado através da multiplicação de matrizes de adjacência A (i.e. $A * A * \ldots$), adicionando-se laços e calculando caminhos de A^n no máximo.

O **fecho transitivo** é dado por: $R = A \vee A^2 \vee \ldots \vee A^n$

Por que não calcular $A^{n+1}, A^{n+2}, A^{n+3}, \ldots$?

Princípio da casas de pombo !!!

Qual o custo computacional ?

Encontrando o Fechamento Transitivo

Fecho Transitivo

Seja n o número de vértices, o fecho transitivo de um dígrafo pode ser calculado através da multiplicação de matrizes de adjacência A (i.e. $A * A * \ldots$), adicionando-se laços e calculando caminhos de A^n no máximo.

O **fecho transitivo** é dado por: $R = A \vee A^2 \vee \ldots \vee A^n$

Por que não calcular $A^{n+1}, A^{n+2}, A^{n+3}, \ldots$ **?**

Princípio da casas de pombo !!!

Qual o custo computacional ?

n multiplicações de matrizes booleanas
n^3 cada multiplicação
Total: $O(n^4)$

Encontrando o Fechamento Transitivo

Fecho Transitivo

Seja n o número de vértices, o fecho transitivo de um dígrafo pode ser calculado através da multiplicação de matrizes de adjacência A (i.e. $A * A * \ldots$), adicionando-se laços e calculando caminhos de A^n no máximo.

O **fecho transitivo** é dado por: $R = A \vee A^2 \vee \ldots \vee A^n$

Por que não calcular $A^{n+1}, A^{n+2}, A^{n+3}, \ldots$?

Princípio da casas de pombo !!!

Qual o custo computacional ?

n multiplicações de matrizes booleanas
n^3 cada multiplicação
Total: $O(n^4)$

É possível melhorar este custo computacional ?

Encontrando o Fechamento Transitivo

Exercício

Dada as seguintes matrizes booleanas, encontre o fecho transitivo de cada uma destas matrizes;

$$G_1 = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

$$G_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

O Algoritmo de Warshall (Fechamento Transitivo)

2. **O Algoritmo de Warshall (ideia)**

- Seja um grafo $G = (V, E)$ com n vértices, o algo. calcula o fechamento transitivo de uma relação.
- Seja um caminho $a_1, a_2, \ldots, a_n$ em G, um vértice interior (Vi) a qualquer vértice $\neq a_1$ e a_n.
- O algoritmo inicia pela matriz booleanda $M_0 = A$ e calcula n matrizes booleanas $M_1, M_2, \ldots, M_n$.
- **Para** $0 \leq k \leq n$**.** $M_k[i, j] = 1$ **sse existir um caminho em** G **entre** a_i **e** a_j **com Vi** $\in \{a_1, a_2, a_3, \ldots, a_k\}$
- $M_n = M_{R^*}$ pois a entrada (i, j) da matriz M_{R^*} será um 1 somente se existir um caminho de v_i para v_j, com todos os vértices interiores no conjunto $\{v_1, \ldots, v_n\}$.

O Algoritmo de Warshall (Fechamento Transitivo)

Em outras palavras...

Seja um conjunto de vértices $\{a_1, a_2, a_3, \ldots, a_n\}$, existe um caminho entre dois vétices vértices a_i e a_j sse existir:

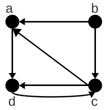
Uma aresta de a_i para a_j

$\vee$ um caminho de a_i para a_j através de um vértice a_1

$\vee$ um caminho de a_i para a_j através de um vértice a_1 e/ou a_2

$\vee \ldots \vee$ um caminho de a_i para a_j através qualquer um dos outros vértices

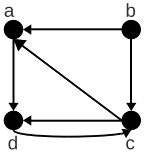
Exemplo: Seja o seguinte grafo onde: $a_1 = a, a_2 = b, a_3 = c$ e $a_4 = d$.



$$M_0 = A = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

O Algoritmo de Warshall (Fechamento Transitivo)

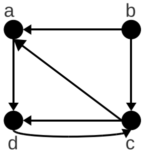
M_1 - caminho onde o vértice interno é $a_1 = a$. Obs: Todos os caminhos de tamanho 1 podem ser utilizados.



$$M_1 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & \color{red}{1} \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

O Algoritmo de Warshall (Fechamento Transitivo)

M_1 - caminho onde o vértice interno é $a_1 = a$. Obs: Todos os caminhos de tamanho 1 podem ser utilizados.

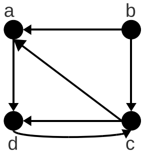


$$M_1 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

M_2 - caminhos onde os vértices internos são $a_1 = a$ e/ou $a_2 = b$.

O Algoritmo de Warshall (Fechamento Transitivo)

M_1 - caminho onde o vértice interno é $a_1 = a$. Obs: Todos os caminhos de tamanho 1 podem ser utilizados.

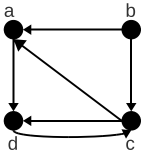


$$M_1 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & \color{red}{1} \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

M_2 - caminhos onde os vértices internos são $a_1 = a$ e/ou $a_2 = b$. Como não existe $M_2 = M_1$.

O Algoritmo de Warshall (Fechamento Transitivo)

M_1 - caminho onde o vértice interno é $a_1 = a$. Obs: Todos os caminhos de tamanho 1 podem ser utilizados.



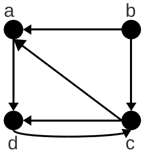
$$M_1 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

M_2 - caminhos onde os vértices internos são $a_1 = a$ e/ou $a_2 = b$. Como não existe $M_2 = M_1$.

M_3 - caminhos onde os vértices internos são $a_1 = a, a_2 = b$ e/ou $a_3 = c$.

O Algoritmo de Warshall (Fechamento Transitivo)

M_1 - caminho onde o vértice interno é $a_1 = a$. Obs: Todos os caminhos de tamanho 1 podem ser utilizados.



$$M_1 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

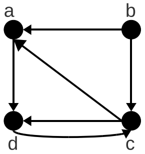
M_2 - caminhos onde os vértices internos são $a_1 = a$ e/ou $a_2 = b$. Como não existe $M_2 = M_1$.

M_3 - caminhos onde os vértices internos são $a_1 = a, a_2 = b$ e/ou $a_3 = c$.

$$M_3 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} \qquad M_4 = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

O Algoritmo de Warshall (Fechamento Transitivo)

M_1 - caminho onde o vértice interno é $a_1 = a$. Obs: Todos os caminhos de tamanho 1 podem ser utilizados.



$$M_1 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

M_2 - caminhos onde os vértices internos são $a_1 = a$ e/ou $a_2 = b$. Como não existe $M_2 = M_1$.

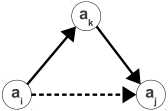
M_3 - caminhos onde os vértices internos são $a_1 = a, a_2 = b$ e/ou $a_3 = c$.

$$M_3 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} \qquad M_4 = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

M_4=<u>Fecha. T.</u>-caminhos VI $= a_1 = a, a_2 = b, a_3 = c$ e/ou $a_4 = d$.

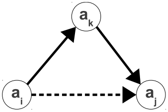
O Algoritmo de Warshall (Fechamento Transitivo)

- O algoritmo calcula o M_{R^*} computando
$M_0 = A, M_1, \ldots, M_n = M_{R^*}$
- Supondo que M_{k-1} foi calculado, pode-se calcular M_k. Existe um caminho de a_i para a_j com $Vi \in \{a_1, a_2, a_3, \ldots, a_k\}$ sse:



O Algoritmo de Warshall (Fechamento Transitivo)

- O algoritmo calcula o M_{R^*} computando
$M_0 = A, M_1, \ldots, M_n = M_{R^*}$
- Supondo que M_{k-1} foi calculado, pode-se calcular M_k. Existe um caminho de a_i para a_j com $Vi \in \{a_1, a_2, a_3, \ldots, a_k\}$ sse:

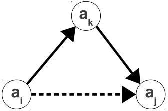


Todo $Vi \in \{a_1, a_2, a_3, \ldots, a_{k-1}\}$ Assim, se $M_{k-1}[i,j] = 1$, então $M_k[i,j] = M_{k-1}[i,j] = 1$ **OU**

$a_k \in Vi$ Existe um caminho $a_i \rightarrow a_k$ com $Vi \in \{a_1, a_2, a_3, \ldots, a_{k-1}\}$. Então: $M_{k-1}[i,k] = 1$

O Algoritmo de Warshall (Fechamento Transitivo)

- O algoritmo calcula o M_{R^*} computando
$M_0 = A, M_1, \ldots, M_n = M_{R^*}$
- Supondo que M_{k-1} foi calculado, pode-se calcular M_k. Existe um caminho de a_i para a_j com $Vi \in \{a_1, a_2, a_3, \ldots, a_k\}$ sse:

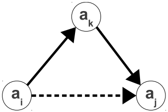


Todo $Vi \in \{a_1, a_2, a_3, \ldots, a_{k-1}\}$ Assim, se $M_{k-1}[i,j] = 1$, então $M_k[i,j] = M_{k-1}[i,j] = 1$ **OU**

$a_k \in Vi$ Existe um caminho $a_i \rightarrow a_k$ com $Vi \in \{a_1, a_2, a_3, \ldots, a_{k-1}\}$. Então: $M_{k-1}[i,k] = 1$
E existe um caminho de $a_k \rightarrow a_j$ com $Vi \in \{a_1, a_2, a_3, \ldots, a_{k-1}\}$. Então: $M_{k-1}[k,j] = 1$.

O Algoritmo de Warshall (Fechamento Transitivo)

- O algoritmo calcula o M_{R^*} computando
$M_0 = A, M_1, \ldots, M_n = M_{R^*}$
- Supondo que M_{k-1} foi calculado, pode-se calcular M_k. Existe um caminho de a_i para a_j com $Vi \in \{a_1, a_2, a_3, \ldots, a_k\}$ sse:



Todo $Vi \in \{a_1, a_2, a_3, \ldots, a_{k-1}\}$ Assim, se $M_{k-1}[i,j] = 1$, então $M_k[i,j] = M_{k-1}[i,j] = 1$ **OU**

$a_k \in Vi$ Existe um caminho $a_i \to a_k$ com $Vi \in \{a_1, a_2, a_3, \ldots, a_{k-1}\}$. Então: $M_{k-1}[i,k] = 1$
E existe um caminho de $a_k \to a_j$ com $Vi \in \{a_1, a_2, a_3, \ldots, a_{k-1}\}$. Então: $M_{k-1}[k,j] = 1$.

$$\mathrm{M_k[i,j] = M_{k-1}[i,j] \vee (M_{k-1}[i,k] \wedge M_{k-1}[k,j])}$$

O Algoritmo de Warshall (Fechamento Transitivo)

Exemplo:

$$G = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

O Algoritmo de Warshall (Fechamento Transitivo)

Exercicio 1:

$$G = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Exercicio 2:

$$G = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Encontrando o Fechamento Transitivo

O Algoritmo de Warshall (Fechamento Transitivo)

Entrada: Matriz $A_{n \times n}$ sem arestas paralelas
Saída: A matriz contendo o fecho transitivo

```
para k = 1 até n faça
    para i = 1 até n faça
        para j = 1 até n faça
            A[i][j] = A[i][j] ∨ (A[i][k] ∧ A[k][j])
        fim
    fim
fim
```

Qual o custo computacional ?

Encontrando o Fechamento Transitivo

O Algoritmo de Warshall (Fechamento Transitivo)

Entrada: Matriz $A_{n \times n}$ sem arestas paralelas
Saída: A matriz contendo o fecho transitivo
para $k = 1$ **até** n **faça**
    para $i = 1$ **até** n **faça**
        para $j = 1$ **até** n **faça**
            $A[i][j] = A[i][j] \vee (A[i][k] \wedge A[k][j])$
        fim
    fim
fim

Qual o custo computacional ?

$$O(n^3)$$

É possível melhorar este custo computacional ?

Referências

- LEISERSON , Charles E.; STEIN, C.; RIVEST, Ronald L., CORMEN, Thomas H. **Algoritmos: Teoria e Prática**, 1ª.ed. Campus, 2002 (caps. 22 a 26);
- GERSTING, Judith L. **Fundamentos Matemáticos para a Ciência da Computação**. 5a. Edição. LTC Editora, 2004. 616p. ISBN-10: 8521614225. ISBN-13: 978-8521614227.
- ROSEN, Kenneth H. **Matemática Discreta e suas Aplicações**. Tradução da 6a. edição em inglês, McGrawHill, 2009, ISBN 978-85-77260-36-2
- GROSS, Jonthan L., YELLEN, Jay. **Graph Theory and Its Applications**, Second Edition, Chapman and Hall/CRC, 2005.
- SEDGEWICK, Robert. **Algorithms in Java, Part 5: Graph Algorithms**, 3rd Edition, Addison-Wesley Professional, 2003;