Aula 5

- Componentes Fortemente Ligados
- Revisões

Definição [DAG - Directed Acyclic Graph - Grafo Dirigido Acíclico]

Um grafo dirigido $G=(V,E)$ diz-se acíclico se não contém caminhos circulares.

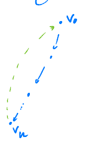
Um caminho circular p num grafo $G=(V,E)$ é uma sequência de vértices $\langle v_0, \dots, v_n \rangle$ tal que:

- $\forall_{0 \le i < n}.\ (v_i, v_{i+1}) \in E$
- $v_0 = v_n$

Lema 1: Um grafo dirigido tem um caminho circular <u>se e só se</u> uma DFS no grafo revela um arco para trás.

Prova:

- Back edge $\Rightarrow$ Caminho circular



  - Existem tree edges $\{(v_i, v_{i+1}) \mid 0 \le i < n\}$ q ligam v_0 a v_n
  - Existe um arco para trás (v_n, v_0)
  - Concluímos q o grafo contém o caminho circular $\langle v_0, \dots, v_n, v_0 \rangle$.

Comentário:

- Se a DFS não revela back edges então o grafo é acíclico.

- Caminho circular $\Rightarrow$ Back edge
  - Seja $\langle v_0, \dots, v_n, v_0 \rangle$ o caminho circular q começa no vértice c/ menor valor de tempo de descoberta
  - Quando v_0 é descoberto existe um caminho de vértices brancos a ligar v_0 a todos os vértices $\{v_i\}_{i=1}^{n}$, em particular ao vértice v_n.
  - O Teorema do Caminho Branco garante que v_n é descendente de v_0 na árvore DFS, donde se conclui q o arco (v_n, v_0) é um back edge.

Lema 2: Se $G=(V,E)$ é um dag e $(u,v) \in E$, então $v.f < u.f$.

Prova: Um dag só tem 3 tipos de arco. Os 3 tipos de arco verificam esta propriedade.

Lema 3: Se $G=(V,E)$ é um dag, então G contém pelo menos um vértice fonte (source) e um vértice alvo (sink), onde um vértice diz-se source se não tem arcos de chegada e diz-se alvo se não tem arcos de partida (respectivamente incoming e outgoing).

Prova:

- Seja s o nó com maior tempo de fim.

  $\forall v \in V.\ v.f \le s.f$

  $\forall v \in V.\ [_v\ ]_v\ [_s\ ]_s$ ou $[_s\ [_v\ ]_v\ ]_s$

  | Não pode haver um arco de v para s porque então $[_v\ [_s\ ]_s\ ]_v$

  | não pode haver um arco de v p/ s porque seria um back edge

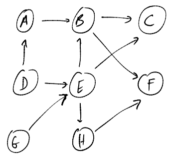
- Seja t o nó com menor tempo de fim

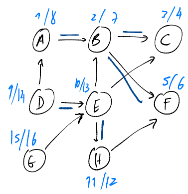
  $\forall v \in V.\ t.f \le v.f$

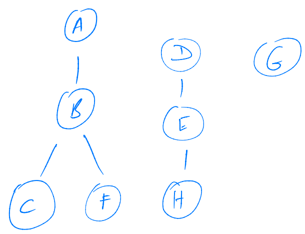
  $[_v\ [_t\ ]_t\ ]_v$ ou $[_t\ ]_t\ [_v\ ]_v$

  ↳ Não pode haver arco de t para v - back edge

Exemplo 3 [Ordenação Topológica]







Ordenação Topológica: G, D, E, H, A, B, F, C

Definição 2 [Ordenação Topológica]

Dado um dag $G=(V,E)$, uma ordenação topológica de G é uma sequência contendo todos os vértices de V tal que se $(u,v) \in E$, u aparece antes de v na ordenação topológica de G.

Algoritmo 2 [Topological Sort]

TopologicalSort(G)

- Compute DFS(G)
  - Calcular a DFS(G)
  - Quando um vértice é terminado inseri-lo no princípio de uma lista ligada
  - Retornar a lista de vértices

[Correcção da Ordenação Topológica]

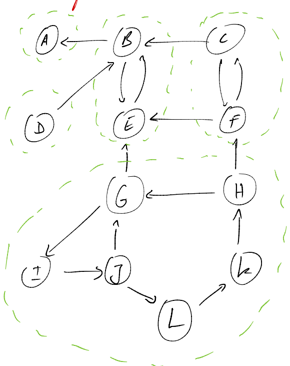
Para provarmos q o algoritmo é correcto basta provarmos que se $(u,v) \in E$ então $u.f > v.f$.

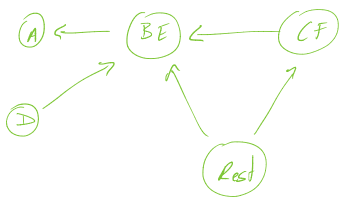
- (u,v) pode ser:
  - tree edge: $u.f > v.f$
  - forward edge: $u.f > v.f$
  - cross edge: $u.f > v.f$

Definição 1 [Componente Fortemente Ligado]

Dado um grafo $G=(V,E)$, um componente fortemente ligado é um conjunto máximo de vértices C tal que para quaisquer $u,v \in C$ temos q $u \leadsto v$ e $v \leadsto u$
(v é atingível a partir de u e u é atingível a partir de v).

Exemplo 1 [SCCs]





Observação 1: Seja $G=(V,E)$ um grafo dirigido, C um SCC em G e u o primeiro nó de C a ser encontrado numa DFS de G, então todos os nós de C são descendentes de u na árvore DFS.

Mas u pode ter mais descendentes.

Por exemplo, se começarmos a DFS no nó L?

Lema 1. Seja $G=(V,E)$ um grafo dirigido e sejam C e C' dois SCCs em G. Dados dois vértices $u,v \in C$ e dois vértices $u',v' \in C'$, se $u \leadsto u'$ então $v' \not\leadsto v$.

Prova.

Suponhamos q existe um caminho entre v' e v e q portanto $v' \leadsto v$.

Então: $u \leadsto u'$, $u' \leadsto v'$, $v' \leadsto v$

Isto significa q C e C' formam juntos um SCC, o q contradiz a hipótese de que C e C' são SCCs.

- Seja $G=(V,E)$ um grafo dirigido e C um SCC em G, definimos:

$$d(C) = \min\{v.d \mid v \in C\}$$
$$f(C) = \max\{v.f \mid v \in C\}$$

Lema 2. Sejam $G=(V,E)$ um grafo dirigido, C e C' dois SCCs em G e (u,v) um arco de C p/ C', então $f(C) > f(C')$

Prova:

- Consideramos 2 casos separadamente:
  a) $d(C) > d(C')$
  b) $d(C') > d(C)$

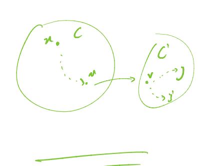
a) $d(C) < d(C')$

- Seja x o 1º nó a ser descoberto em C e y o 1º nó a ser descoberto em C'. Da hipótese, concluímos que $d(x) > d(y)$.
- Concluímos também que:
  - $x \leadsto u \wedge u \leadsto v \wedge v \leadsto y$
  - $x \leadsto y$

- Quando x é descoberto existe um caminho branco a ligar x a todos os nós de C', de onde concluímos q todos os nós de C' são descendentes de x na árvore DFS e q portanto têm menor tempo de fim. Fundamental:

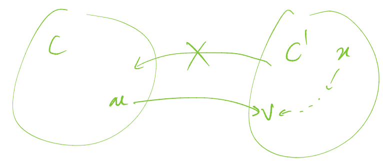
  $\forall v \in C'.\ f(v) < f(x)$

- $f(C') < f(C)$



2) $d(C') < d(C)$

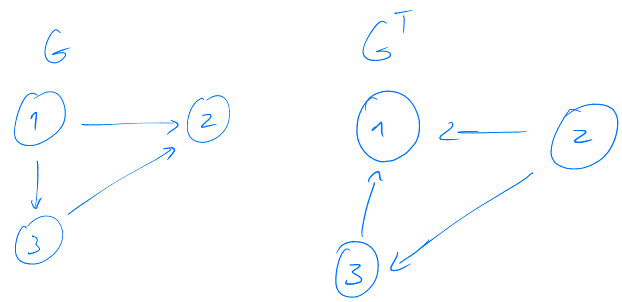
· Pelo lema 1, concluímos q não existe um arco (y, x) em G c/ $y \in C'$ e $x \in C$. Assim sendo, quando o primeiro vértice de C' é descoberto, não existe nenhum caminho branco a ligar esse vértice aos vértices de C. Concluímos, pelo teorema do Caminho Branco, que nenhum nó de C é descendente de nós de C' na árvore DFS. Assim, $f(C') < f(C)$.



Definição 2 [Grafo Transposto]

Seja $G=(V,E)$ um grafo dirigido, o grafo transposto de G, designado por G^T, é definido como se segue: $G^T = (V, E^T)$, onde

$$E^T = \{(v,u) \mid (u,v) \in E\}$$



· É simplesmente o grafo q se obtém a partir de G trocando o sentido dos arcos.

Lema 3. Sejam C e C' dois SCCs num grafo dirigido $G=(V,E)$, temos que:

$$u \in C \wedge v \in C' \wedge (u,v) \in E^T \Rightarrow f(C') > f(C)$$

Prova:

1. $u \in C$ - (hyp)
2. $v \in C'$ - (hyp)
3. $(u, v) \in E^T$ - (hyp)
4. $(v, u) \in E$ - (3) + def. de grafo transposto
5. $f(C') > f(C)$ - (2) + (1) + (4) + Lema 2

Observação 2. O SCC c/ maior tempo de fim numa DFS no grafo original não tem outgoing edges no grafo transposto.

Lema 4. Se C é um SCC em $G=(V,E)$, então C é tb um SCC no grafo $G^T=(V,E^T)$.

Prova:

· Temos de provar que:

$\forall x,y \in C.\ G^T \vdash x \rightsquigarrow y$

· Sejam x e y dois vértices em C, temos de provar que $G^T \vdash x \rightsquigarrow y$. Como C é um SCC em G, sabemos q $G \vdash y \rightsquigarrow x$. De onde segue que $G^T \vdash x \rightsquigarrow y$.

Lema 5. $(G^T)^T = G$

Prova. Seja $G = (V, E)$, então $(G^T)^T = (V, (E^T)^T)$

$(u,v) \in (E^T)^T \Leftrightarrow (u,v) \in E$

$(u,v) \in (E^T)^T$

$\Leftrightarrow (v,u) \in E^T$

$\Leftrightarrow (u,v) \in E$

Lema 6. G e G^T têm os mesmos componentes fortemente ligados.

Prova. C é um SCC de $G \Leftrightarrow C$ é um SCC de G^T

$\boxed{\Rightarrow}$ Lema 4

$\boxed{\Leftarrow}$

1. C é um SCC de G^T - hyp
2. C é um SCC de $(G^T)^T$ - (1) + Lema 4
3. C é um SCC de G - (2) + Lema 5

Observação 3: Um SCC no grafo original é um SCC no grafo transposto.

Algoritmo 1 [SCCs]

SCCs(G)

- Executar DFS(G) p/ cálculo do tempo de fim
- Calcular G^T
- Executar DFS(G^T) → no ciclo principal considerar os vértices por ordem decrescente de tempo de fim DFS(G)
- O conjunto de vértices de cada árvore DFS corresponde a um SCC.

Complexidade:

$O(|E| + |V|)$

Teorema 1 [Correcção SCCs]

O algoritmo SCCs é correcto.

Prova:

Provamos a correcção do algoritmo por indução no nº de árvores encontradas durante a DFS calculada no grafo transposto.

$\boxed{n=0}$ É vacuosamente verdade.

$\boxed{n=k+1}$

- Admitimos que as k primeiras árvores encontradas são SCCs e temos de provar q a árvore $(k+1)$ tb é um SCC.
- Começamos por observar q no momento em q o primeiro nó da árvore $k+1$ é encontrado, seja x esse nó, há um caminho branco a ligar x a todos os nós do SCC q contém x, seja C_x esse SCC. Pelo Lema 6, sabemos q C_x é um SCC de G e de G^T. Pelo Teorema do Caminho branco, sabemos q todos os nós de C_x são descendentes de x na árvore DFS $k+1$. Resta provar, q esta árvore nenhum nó além dos nós em C_x.
- Suponhamos, por absurdo, q existe um nó y na árvore DFS com raiz em x e q não está em C_x. Isto significa q há um arco (x', y) em E^T a ligar C_x a C_y (o SCC de y). Aplicando o Lema 3, concluímos que $f(C_y) > f(C_x)$, de onde segue q $y.f > x.f$. Isto contradiz a hipótese de q os nós são considerados por ordem decrescente de tempo de fim.

Problema 1 [Grafo semi-ligado]

Um grafo dirigido $G=(V,E)$ diz-se semi-ligado se para cada dois vértices $u,v \in V$
$v \leadsto u$ ou $u \leadsto v$

Proponha um algoritmo para determinar se um grafo é semi-ligado.

1. Determinar as SCCs de G
2. Aplicar o algoritmo de ordenação topológica no grafo das SCCs, G_{SCC}
3. O grafo é ligado sse a ordenação encontrada corresponde a um caminho no grafo G_{SCC}.

- Complexidade: $O(|E|+|V|)$
- Correcção:

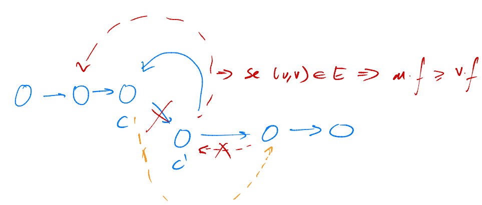
[⇒] Se a ordenação topológica corresponde a um caminho no grafo G_{SCC}, então G é semi-ligado.

- Suponhamos q a ordenação topológica encontra um caminho.
- Consideramos dois vértices quaisquer u e v em G. Se o u e v pertencem ao mesmo SCC então há um caminho a ligá-los. Suponhamos q u e v pertencem a SCCs distintos, C_u e C_v.

  Temos q $C_u \leadsto C_v$ ou $C_v \leadsto C_u$ em G_{SCC}, pelo q $u \leadsto v$ ou $v \leadsto u$ no grafo original.

[⇐] • Suponhamos q o grafo é semi-ligado e a ordenação topológica não corresponde a um caminho no grafo G_{SCC}. Então, concluímos que existem dois componentes C e C' tais que:

$$f(C) < f(C') \quad e \quad C \rightarrow C' \ \ ((C,C') \notin E_{SCC})$$



- Da correcção da ordenação topológica, concluímos que não há qualquer caminho em G_{SCC} a ligar C a C' ou C' a C. Donde concluímos q não há qualquer caminho em G a ligar os vértices de C aos vértices de C'.

Problema 2 [Fonte Única]

07/06 (exame)

Dado um grafo $G=(V,E)$ onde $V=\{v_1,\dots,v_n\}$. Proponha um algoritmo para identificar o vértice v_i de menor índice tal q todos os vértices são atingíveis a partir de v_i.

1. Determinar os SCCs de G
2. Determinar o componente do grafo G_{SCC} q não tem arcos incidentes, seja C esse componente.
3. Retornar o vértice de C com menor índice.

Problema 3 [Ciclos]

10/11

Proponha um algoritmo q, dado um grafo dirigido $G=(V,E)$, determine se existem ciclos e, caso não existam, retorne o caminho mais longo em G.

- Solução 1 (Duas DFSs)
  - DFS 1 ⇒ o grafo não tem ciclos se não forem encontrados arcos p/ trás
  - DFS 2 ⇒ repetimos a DFS começando no nó com maior tempo de fim. ?
- Solução 2
  - DFS 1 ⇒ se houver 1 arco p/ trás, existem ciclos. Calculamos a profundidade do caminho à medida q calculamos a DFS (temos de ter cuidado c/ as cross edges).