Componentes fortemente conexos

Marco A L Barbosa
malbarbo.pro.br

Departamento de Informática
Universidade Estadual de Maringá

Vimos anteriormente uma classe de problemas que consistia em determinar a ordem de execução de tarefas com precedências.

Um aspecto interessante na resolução desses problemas foi a modelagem:

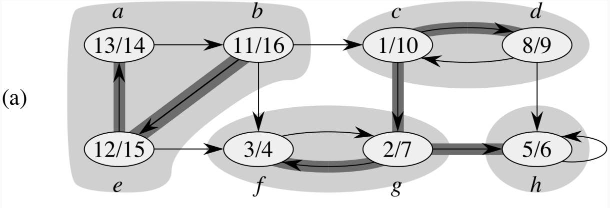
- Representar o problema com um modelo (grafo) e definir o que é uma solução (ordenação topológica).

Embora nós tenhamos "construído" o problema da ordenação topológica para resolver o problema de determinar a ordem de execução de tarefas com precedências, muitos outros problemas podem ser resolvidos com a ordenação topológica.

A criação e utilização de modelos é uma ferramente bastante útil na resolução de problemas.

De fato, para muitos problemas o principal passo na sua resolução é a modelagem. Isto porque a modelagem pode ter sido feita em termos de um problema já conhecido e para o qual já existem algoritmos eficientes.

Por isso é importante estudar problemas e algoritmos em grafos! Quanto mais problemas e algoritmos você conhecer, melhor preparado você estará para resolver problemas do mundo real.

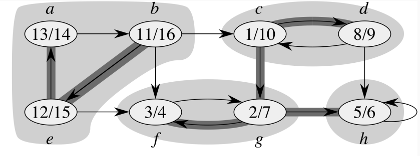


Um **componente fortemente conexo** (SCC) de um grafo orientado $G = (V, E)$ é um conjunto máximo de vértices $C \subseteq V$, tal que, para todo par de vértice u e $v \in C$, existe um caminho de u para v e um caminho de v para u, isto é $u \rightsquigarrow v$ e $v \rightsquigarrow u$.

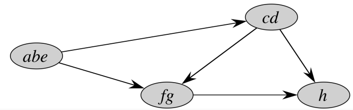
Projete um algoritmo que receba como entrada um grafo orientado e encontre os seus componentes fortemente conexo.

Podemos entender melhor a relação entre os SCC's de um grafo criado um grafo com os SCC's.

(a)



(c)



Que característica interessante podemos observar no grafo de componentes? Ele é acíclico.

O **grafo de componentes** de um grafo $G = (V, E)$ com os SCC's $C_1, C_2, \ldots C_k$ é o grafo $G^{\text{SCC}} = (V^{\text{SCC}}, E^{\text{SCC}})$, onde:

- $V^{\text{SCC}} = \{v_1, v_2, \ldots, v_k\}$ e cada vértice v_i representa o SCC C_i de G.
- $E^{\text{SCC}} = \{(v_i, v_j) |$ existe um vértice $x \in C_i$, um vértice $y \in C_j$ e $(x, y) \in E\}$

Lema 22.13

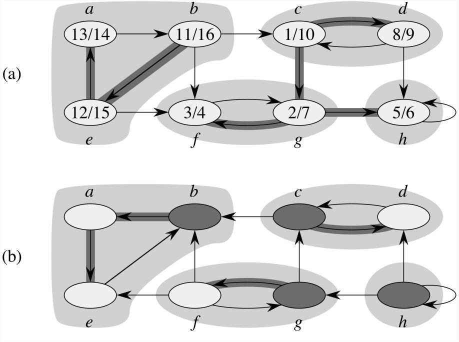
Sejam C e C' SCC distintos em um grafo orientado $G = (V, E)$, seja $u, v \in C$ e seja $u', v' \in C'$. Suponha que exista um caminho $u \rightsquigarrow u'$ em G. Então, não pode existir um caminho $v' \rightsquigarrow v$ em G.

Prova

Por contradição.

Suponha que exista um caminho $v' \rightsquigarrow v$ em G. Então existem caminhos $u \rightsquigarrow u' \rightsquigarrow v'$ e $v' \rightsquigarrow v \rightsquigarrow u$ em G. Portanto, u e v' são acessíveis um a partir do outro, e não podem estar em SCC separados, o que é uma contradição! $\square$

O **grafo transposto** de um grafo $G = (V, E)$ é o grafo $G^T = (V, E^T)$, onde $E^T = \{(u, v) : (v, u) \in E\}$. Ou seja, G^T é G com todas as arestas invertidas.



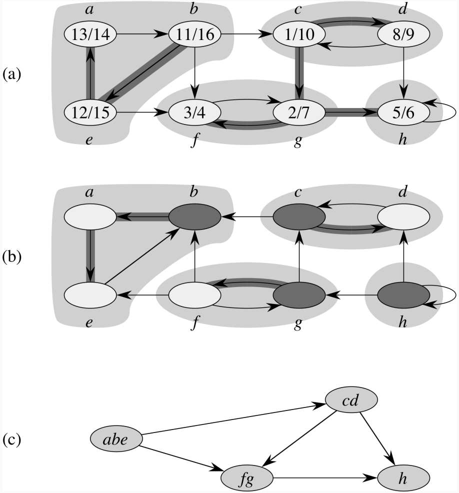
Qual o tempo necessário para calcular G^T a partir de G? $\Theta(V + E)$ para lista de adjacências (veja o exercício 22.1-3). Qual a relação entre os SCC's de G e G^T. São os mesmos. Ou seja, dois vértices u e v são acessíveis um a partir do outro em G se e somente se eles são acessíveis um a partir do outro em G^T.

Procedimento STRONGLY-CONNECTED-COMPONENTS

Usando estas observações e propriedades Kosaraju propos o seguinte algoritmo para encontrar os componentes fortemente conexos de um grafo:

STRONGLY-CONNECTED-COMPONENTS(G)

1 chamar DFS(G) para calcular o
  tempo de término $v.f$ de cada vértice
2 calcular G^T
3 chamar DFS(G^T) mas, no laço principal
  de DFS, considerar os vértices
  em ordem decrescente de $u.f$
4 os vértices de cada árvore DFS formada na
  linha 3 formam um componente
  fortemente conexo



Ideia: ao considerar os vértices na segunda execução do DFS na ordem decrescente dos tempos de términos obtidos na primeira execução do DFS, estamos visitando os vértices do grafo de componentes em ordem topológica.

Procedimento STRONGLY-CONNECTED-COMPONENTS

STRONGLY-CONNECTED-COMPONENTS(G)

1 chamar DFS(G) para calcular o tempo de término $v.f$ de cada vértice
2 calcular G^T
3 chamar DFS(G^T) mas, no laço principal de DFS, considerar os vértices em ordem decrescente de $u.f$
4 os vértices de cada árvore DFS formada na linha 3 formam um componente fortemente conexo

Tempo de execução

- O tempo do DFS das linhas 1 e 3 é $\Theta(V + E)$
- Conforme os vértices são terminados na chamada do DFS da linha 1, os vértices são inseridos na frente de uma lista ligada ($O(1)$), como cada vértice é inserido apenas uma vez, o tempo total de operações de inserções é $\Theta(V)$
- O tempo para calcular o grafo transposto na linha 2 é $\Theta(V + E)$
- Portanto, o tempo de execução do algoritmo é $\Theta(V + E)$

Vamos definir duas questões de notação

- As referências a $u.d$ e $u.f$ referem-se aos valores obtidos na primeira execução do DFS
- Para um conjunto $U \subseteq V$, definimos
  - $d(U) = \min_{u \in U}\{u.d\}$ (tempo de descoberta mais antigo)
  - $f(U) = \max_{u \in U}\{u.f\}$ (tempo de término mais recente)

Lema 22.14

Sejam C e C' SCC distintos em $G = (V, E)$. Suponha que exista uma aresta $(u, v) \in E$, tal que $u \in C$ e $v \in C'$. Então $f(C) > f(C')$.

Prova

Que componente teve o primeiro vértice descoberto, C ou C'? Vamos considerar os dois casos.

No primeiro caso $d(C) < d(C')$. Seja x o primeiro vértice descoberto de C:

- Qual a cor dos vértices em C e C' no momento em que x é descoberto? Branco, porque x foi o primeiro vértice descoberto de C e C'.
- Existe um caminho de vértices branco de x para todos os vértices de C.
- Como $(u, v) \in E$ e $u \in C$ e $v \in C'$, então existe um caminho de vértices brancos $x \rightsquigarrow u \rightarrow v \rightsquigarrow w$ para cada vértice $w \in C'$.
- Portanto, pelo teorema do caminho branco, todos os vértices de C e C' se tornam descendentes de x.
- Pelo Corolário 22.8, x tem o maior tempo de término entre todos os descentes e portanto $x.f = f(C) > f(C')$.

No segundo caso $d(C') < d(C)$. Seja y o primeiro vértice descoberto de C':

- Qual a cor dos vértices em C e C' no momento em que y é descoberto? Branco, porque y foi o primeiro vértice descoberto de C e C'.
- Existe um caminho de vértices branco de y para todos os vértices de C'. Então todos os vértices de C' se tornam descendentes de y e portanto $y.f = f(C')$.
- E os vértices de C serão descendentes de y? Não. Como existe a aresta (u, v) de C para C', não pode existir uma aresta de C' para C (Lema 22.13) e portanto os vértices de C não são acessíveis a partir de y e não serão descendentes de y.
- Logo, no tempo $y.f$ todos os vértices de C ainda são brancos, então para qualquer vértices $w \in C$, $w.f > y.f$ e portanto $f(C) > f(C')$. $\square$

Corolário 22.15

Sejam C e C' SCC distintos em $G = (V, E)$. Suponha que exista uma aresta $(u, v) \in E^T$, tal que $u \in C$ e $v \in C'$, então $f(C) < f(C')$.

Prova

Como $(u, v) \in E^T$, então $(v, u) \in E$. Como os SCC's de G e G^T são os mesmos, o lema 22.14 implica que $f(C) < f(C')$. $\square$

Este corolário é fundamental para entendermos porque o algoritmo STRONGLY-CONNECTED-COMPONENTS(G) funciona corretamente.

Teorema 22.16

STRONGLY-CONNECTED-COMPONENTS(G) calcula corretamente os SCC's de um grafo orientado G.

Prova

Vamos discutir a ideia da prova. Para a prova formal, veja o livro.

A segunda execução do DFS começa por um vértice em uma SCC C tal que $f(C)$ é máximo. Seja x este vértice.

A execução do DFS iniciada em x visita pelo menos todos os vértices de C. Algum outro vértice de alguma outra componente $C' \neq C$ é visitado nessa execução? Pelo corolário 22.15, se existisse alguma aresta de um vértice de C para um vértice de C', então $f(C) < f(C')$, mas como $f(C) > f(C')$, então esta aresta não existe e portando nenhum vértice de outra componente é descoberto. Logo, esta execução do DFS visita apenas os vértices de C, descobrindo este SCC.

A próximo vértice inicial escolhido pelo DFS está em um SCC C' tal que $f(C')$ é máximo em relação a todos os outros SCC (sem considerar C). O DFS visita todos os vértices de C', e as únicas arestas para fora de C' vão para C (mesma argumentação usando o corolário 22.15 do item anterior), cujo os vértices já foram visitados. Ou seja, esta execução do DFS descobre apenas os vértices de C'.

Este processo continua até que todas as componentes tenham sido encontradas.

22.5-1 - Como o número de componentes fortemente conexos de um grafo pode mudar se uma nova aresta é adiciona?

22.5-6 - Dado um grafo orientado $G = (V, E)$, explique como criar um outro grafo $G' = (V, E')$ tal que

a) G' tem os mesmos componentes fortemente conexos de G;

b) G' tem o mesmo grafo de componentes de G; e

c) E' tem o menor número possível de arestas.

Descreva um algoritmo eficiente para calcular G'.

Exercícios

Veja a lista de exercícios e algumas soluções na página da disciplina.

Thomas H. Cormen et al. Introduction to Algorithms. 3rd edition. Capítulo 22.5.