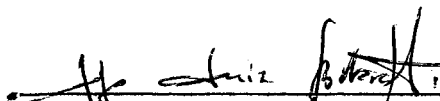


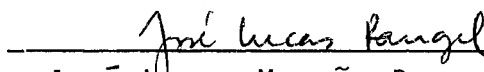
REPRESENTAÇÃO DE GRAFOS EM COMPUTADOR

Heloisa Celia Ramalho Pombo

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS (M.Sc.)

Aprovada por:


Jayme Luiz Szwarcfiter
(Presidente)


José Lucas Mourão Rangel Netto


Paulo Augusto da Silva Veloso

RIO DE JANEIRO, RJ — BRASIL
JANEIRO DE 1979

POMBO, HELOISA CELIA RAMALHO

Representação de Grafos em Computador
[Rio de Janeiro] 1979.

VI, 110p. 29,7cm (COPPE-UFRJ, M.Sc,
Engenharia de Sistemas e Computação, 1979)

Tese - Univ. Fed. Rio de Janeiro. Fac.
Engenharia

1.Representação de grafos I.COPPE/UFRJ
II.Título(série).

AGRADECIMENTOS

Agradeço ao Jayme pela grande compreensão e ajuda.

Agradeço também ao Dinamérico, meu marido, pelo apoio e incentivo ao estudo.

RESUMO

Este trabalho é uma apresentação geral sobre representação de grafos (direcionados ou não), contendo todas as representações encontradas na literatura consultada.

Vários tipos de representações são apresentados: matrizes de adjacência, de incidência, de posicionamento e orientada graficamente; estruturas de adjacência, de incidência e de posicionamento; representação vetorial de arestas, pseudo-árvore, árvore-coárvore, estrutura de lista; representação lista de ramos, seqüência de adjacência, k-fórmula, representação por strings, representações seqüenciais de pseudo-árvores e seqüências ternárias.

Cada representação constitui um capítulo do trabalho, incluindo definição, exemplos, propriedades básicas, comentários e referências. Em cada caso é considerada a possibilidade de representação de grafos direcionados e não direcionados, pseudografos e multigrafos. Compõe ainda o trabalho uma comparação de todas as representações em termos de espaço necessário para armazenamento.

ABSTRACT

This work is a survey on graph representation. It contains all representations found in the consulted literature.

Several types of representations are presented : the adjacency, incidence, position and draw oriented matrices ; the adjacency, incidence and position structures ; the edge vector representation, pseudo-tree, tree-cotree, list structure ; the branch list representation, adjacency sequence, k-formula, string representation, pseudo-tree sequential representations and ternary sequence representations.

Each representation constitutes a chapter in this work, including definition, examples, basic properties, comments and references. In each case we consider the possibility of representing directed and nondirected graphs, pseudographs and multigraphs. We also present a comparison of all representations in terms of storage requirements.

ÍNDICE

	Pag.
I. Introdução e Definições Básicas	1
II. Matriz de Adjacência	6
III. Matriz de Incidência	11
IV. Matriz de Posicionamento	15
V. Representação Orientada Graficamente	21
VI. Estrutura de Adjacência	25
VII. Estrutura de Incidência	37
VIII. Estrutura de Posicionamento	42
IX. Representação Vetorial de Arestas	45
X. Pseudo-árvore	51
XI. Representação de Grafos por Árvores Binárias	60
XII. Árvore-Coárvore	66
XIII. Representação por Estrutura de Lista	71
XIV. Lista de Ramos	75
XV. Seqüência de Adjacência	80
XVI. Representação por k-fórmula	82
XVII. Representação por Strings	87
XVIII. Representações Seqüenciais de Pseudo-árvores	93
XIX. Representações Seqüenciais Ternárias	99
XX. Conclusão	105
Referências Bibliográficas	109

CAPÍTULO I

INTRODUÇÃO E DEFINIÇÕES BÁSICAS

Este trabalho constitui um apanhado geral sobre representação de grafos, compreendendo todas as representações encontradas na literatura consultada, bem como outras ainda inéditas, que estão sendo introduzidas aqui pelo Prof. Jayme Szwarcfiter. Naturalmente, além dessas seria ainda possível definir outras representações e, talvez, algumas tenham já sido publicadas.

O conceito de representação de um grafo (direcionado ou não) que estamos usando pode ser estabelecido da seguinte forma: considere um grafo G_M que modela uma estrutura abstrata, e considere uma estrutura G_R que descreve esse grafo; dizemos que G_R é uma representação de G_M se, dado G_R apenas, podemos reproduzir todas as informações contidas no grafo G_M , de maneira unívoca. Portanto, uma estrutura não é considerada uma representação se não contiver todos os elementos (vértices, arestas) do grafo em questão ou se não definir univocamente o grafo (isto é, se grafos distintos puderem produzir uma mesma estrutura).

Podemos dizer que uma representação de grafo em computador consiste em um conjunto finito de símbolos e um conjunto finito de regras que identificam univocamente o grafo sendo representado, onde os símbolos seriam o resultado da aplicação das regras.

Se duas representações de um mesmo grafo têm diferentes conjuntos de regras, então dizemos que elas são de tipos diferentes. Se elas têm o mesmo conjunto de regras, mas diferentes conjuntos de símbolos, dizemos que elas são versões distintas do mesmo tipo de representação. Elas são iguais se seus conjuntos de regras e de símbolos forem iguais. Por exemplo, a ma-

triz de adjacência e a matriz de incidência de um grafo são representações de tipos diferentes ; duas matrizes de adjacência diferentes são versões distintas do mesmo tipo de representação, enquanto que duas matrizes de adjacência iguais são representações iguais.

A representação deve determinar de forma unívoca o grafo que representa. No entanto, um dado tipo de representação de um grafo pode não estar univocamente determinado, isto é, o conjunto de símbolos pode não ser único, surgindo aí várias versões. O problema de decidir se duas versões distintas de uma mesma representação correspondem ou não a um mesmo grafo é geralmente um problema não trivial, possivelmente NP-completo . Este problema não está sendo abordado por nós aqui.

A idéia deste trabalho surgiu do fato de não se encontrar, na literatura sobre grafos em geral, nenhum estudo exaustivo compreendendo as várias representações existentes para grafos. São apresentadas aqui várias classes de representações - por matrizes, por listas e representações sequenciais - perfazendo um total de 18 representações.

Cada representação constitui um capítulo completo, incluindo definição, exemplos, propriedades, comentários e referências.

Finalmente, apresentamos uma comparação de todas as representações contidas neste trabalho, em termos do espaço necessário para o armazenamento de cada uma.

Definições Básicas Sobre Grafos

Um grafo G é um par $G = (V, X)$ onde V é um conjunto finito cujos elementos são os vértices de G , e X é um conjunto de pares de elementos distintos de V .

Anotamos um par de elementos em X por (u, v) ou uRv , onde $u, v \in V$, e dizemos que os vértices u e v estão relacionados ou formam uma aresta de G .

Se $e = (u, v)$ é uma aresta, dizemos que os vértices u e v são adjacentes, e que o vértice u e a aresta e são incidentes (bem como o vértice v e a aresta e). Se duas arestas distintas possuem um vértice em comum então dizemos que elas são adjacentes.

Se os pares de vértices em X são pares ordenados (isto é, se $(u, v) \neq (v, u)$), então o grafo é direcionado ou é um digrafo; caso contrário o grafo é não direcionado.

Em um digrafo, uma aresta (u, v) é dita de u para v . O grau de entrada de um vértice w é o número de arestas para w , enquanto que o grau de saída de w é o número de arestas de w . Se w tem grau de entrada zero então w é um vértice fonte, e se w tem grau de saída zero então w é um sumidouro.

Em um grafo qualquer, um vértice isolado é um vértice que não está relacionado com nenhum outro vértice.

Um laço é uma aresta que relaciona um vértice a ele mesmo. De acordo com a definição dada, não há laços em grafos. Se forem permitidos laços teremos um pseudografo. Se forem permitidas arestas múltiplas, isto é, mais de uma aresta relacionando o mesmo par de vértices, teremos um multigrafo.

Uma rotulação dos vértices (arestas) de um grafo é uma atribuição de "nomes" ou rótulos aos vértices (arestas), de forma a distingui-los uns dos outros.

Seja $G = (V, X)$ um grafo. Um subgrafo G_1 de G é um grafo $G_1 = (V_1, X_1)$ tal que $V_1 \subset V$ e $X_1 \subset X$.

Seja agora $S \subset V$. O subgrafo induzido $\langle S \rangle$ é o subgrafo maximal de G cujo conjunto de vértices é S .

Um grafo é completo se possui um número máximo de arestas para o dado conjunto de vértices.

Um caminho de um grafo com comprimento n é uma seqüência de vértices v_1, \dots, v_n tais que (v_i, v_{i+1}) é uma aresta do grafo, para $i = 1, \dots, n-1$. O caminho é elementar se todos os vértices forem distintos.

Em um digrafo, um semicaminho é uma seqüência de vértices v_1, \dots, v_n onde (v_i, v_{i+1}) ou (v_{i+1}, v_i) é uma aresta, para $i = 1, \dots, n-1$.

Um ciclo (ciclo elementar) é uma seqüência de vértices v_0, v_1, \dots, v_n onde $v_0 = v_n$, v_1, \dots, v_n é um caminho elementar e $n > 2$ se o grafo for não direcionado ou $n > 1$ se o grafo for direcionado.

Um grafo G não direcionado é conexo se existir um caminho entre quaisquer dois vértices de G . As componentes conexas de G são os subgrafos maximais de G conexos. G é desconexo se não for conexo.

Seja D um digrafo. Sejam u, v vértices quaisquer de D . Se existir um caminho de u para v e de v para u , para todo u, v vértices de D , então D é fortemente conexo. Se existir um caminho de u para v ou de v para u , para todo u, v , então D é unilateralmente conexo. Se existir um semicaminho de u para v , para todo u, v , então D é fracamente conexo. D é desconexo se não for sequer fracamente conexo.

Uma árvore é um grafo não direcionado acíclico conexo. Uma árvore enraizada direcionada (árvore enraizada) é um digrafo acíclico no qual exatamente um vértice (a raiz) tem grau de entrada zero, enquanto todos os demais têm grau de entrada igual a 1.

Sejam v, w vértices de uma árvore. Se existir uma aresta de v para w então v é o pai de w , e w é o filho de v ; e se u é outro vértice tal que v é pai de u então u e w são irmãos.

Uma árvore geradora de um grafo não direcionado G é um subgrafo S de G tal que S é uma árvore que contém todos os vértices de G . A coárvore é o menor subgrafo de G que contém todas as arestas não pertencentes a S .

Um desenho de G é uma figura tal que os vértices correspondem a pontos, e as arestas correspondem a linhas unindo os pontos. Um cruzamento no desenho de G é uma interseção das linhas do desenho em pontos diferentes do vértices.

Um desenho plano é um desenho de G num plano sem cruzamentos. Um grafo é planar quando admite desenho plano.

O número de cruzamento de um grafo é o menor número de cruzamentos dentre todos os possíveis desenhos de G . Por exemplo, o número de cruzamento de um grafo planar é zero.

CAPÍTULO II

MATRIZ DE ADJACÊNCIAII-1. Definição da Representação

Seja G um grafo cujos N vértices estão rotulados .
 A matriz de adjacência de G é a matriz $N \times N$

$$A(G) = (a_{ij}), \quad i, j = 1, \dots, N$$

definida por :

$$a_{ij} = \begin{cases} 1, & \text{se o vértice } v_i \text{ é adjacente ao vértice } v_j \text{ (isto é, se } v_i R v_j \text{) ,} \\ 0, & \text{caso contrário .} \end{cases}$$

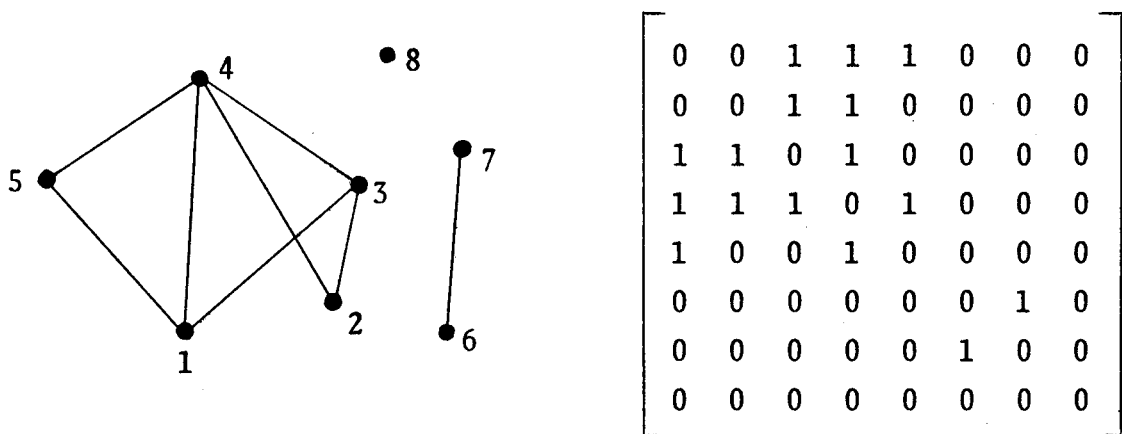
II-2. Exemplo

Figura II-1. Um grafo e uma representação por matriz de adjacência.

II-3. Propriedades Básicas e Comentários

Uma vez que em um grafo não há laços, a diagonal principal é toda nula (isto é, $a_{ii} = 0, i = 1, \dots, N$).

Como a relação entre as arestas de um grafo é si-

métrica, temos $a_{ij} = a_{ji}$ para todo $i, j = 1, \dots, N$, e portanto a matriz $A(G)$ é sempre simétrica.

Um vértice isolado não está relacionado com nenhum outro vértice. Assim, v_i é vértice isolado se e somente se $a_{ij} = a_{ji} = 0$, para todo $j = 1, \dots, N$.

Seja X o conjunto de arestas de G . Seja M o número de elementos de X (isto é, $\text{Card } X = M$). Ponhamos

$$A_1 = \{ (i, j) \mid a_{ij} = 1 \} .$$

Então $v_i R v_j \implies (i, j) \in A_1, (j, i) \in A_1$

e portanto

$$\text{Card } A_1 = 2 \text{ Card } X = 2M .$$

Seja agora V o conjunto de vértices de G , e $S \subset V$. Então $a_{ij} \in A(\langle S \rangle) \iff v_i, v_j \in S$. Isto sugere um algoritmo bastante simples para a obtenção do subgrafo de G induzido por S , $\langle S \rangle$.

Ponhamos $A^n(G) = (b_{ij})$ para n inteiro positivo . Então cada b_{ij} é o número de caminhos distintos de v_i a v_j de comprimento n .

Sendo $A(G)$ uma matriz $N \times N$, o espaço necessário para armazenar a representação consiste de N^2 informações binárias (0 ou 1). Naturalmente, é possível agrupar vários elementos da matriz em uma palavra de computador, através da associação de um bit a cada elemento da matriz, economizando assim espaço em memória. Além disso, uma vez que a matriz de adjacência de um grafo é simétrica, é possível reduzir ainda mais o espaço gasto, armazenando $A(G)$ como uma matriz triangular.

É possível redefinir a matriz de adjacência de forma a permitir que seja usada também para representar multigrafos e/ou pseudografos, fazendo $a_{ij} = m$, onde m é o número de relações entre v_i e v_j . A figura II-2 a seguir contém um exemplo.

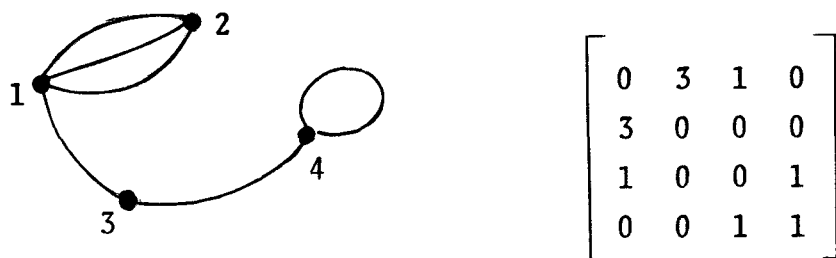


Figura II-2. Um pseudomultigrafo e uma representação por matriz de adjacência.

Observamos que a matriz ainda permanece simétrica.

A definição da matriz de adjacência $A(D)$ de um digrafo D pode ser a mesma definição que para grafos não direcionados. Exemplo :

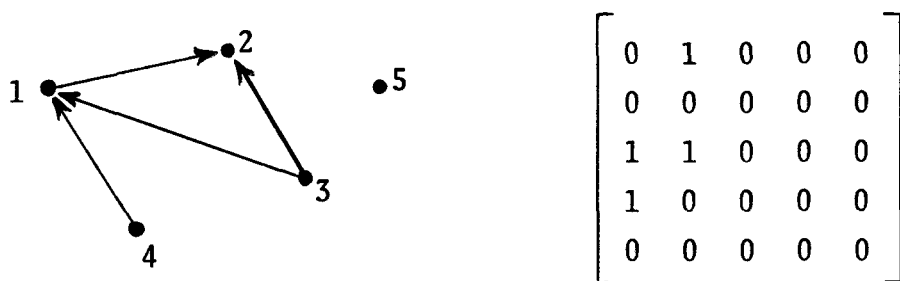


Figura II-3. Um digrafo e uma representação por matriz de adjacência.

Uma vez que a relação entre as arestas não é em geral simétrica, a matriz de adjacência não é em geral simétrica.

Pode ser facilmente observado que, se a i -ésima linha da matriz for toda nula, então o vértice v_i é um sumidouro, e, se a j -ésima coluna for toda nula, então o vértice v_j é um vértice fonte. Também, como um vértice isolado é tanto um vértice fonte como um sumidouro, a mesma propriedade quanto a vértices isolados de um grafo se aplica a digrafos.

Definamos A_1 como para grafos não direcionados, isto é, $A_1 = \{ (i,j) \mid a_{ij} = 1 \}$. Temos agora

$$v_i R v_j \implies (i,j) \in A_1$$

e portanto

$$\text{Card } A_1 = \text{Card } X = M.$$

É possível armazenar um grafo direcionado como uma matriz triangular, redefinindo a matriz de adjacência como se segue :

$$A(D) = (a_{ij}) , \text{ onde}$$

$$a_{ij} = \begin{cases} 1, & \text{se } v_i R v_j \text{ mas } v_j \cancel{R} v_i , \\ -1, & \text{se } v_j R v_i \text{ mas } v_i \cancel{R} v_j , \\ *, & \text{se } v_i R v_j \text{ e } v_j R v_i , \\ 0, & \text{se } v_i \cancel{R} v_j \text{ e } v_j \cancel{R} v_i . \end{cases}$$

Como dado a_{ij} podemos determinar a_{ji} , para todo $i = 1, \dots, N$, é possível armazenar a matriz $A(D)$ definida acima como uma matriz triangular, sem perda de informação, conforme pode ser visto na figura II-4.

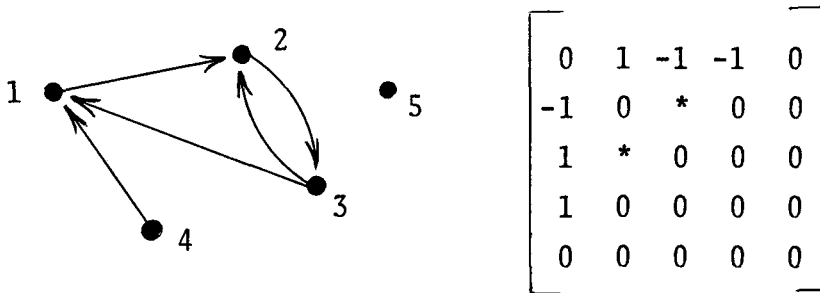


Figura II-4. Um digrafo e uma representação por matriz de adjacência triangular.

De um modo geral, este tipo de representação para grafos (ou digrafos) é boa para manipulações. Mas a maior vantagem deve-se ao fato da matriz de adjacência apresentar várias propriedades teóricas interessantes, que são úteis no projeto de algoritmos relacionados a estruturas de grafos. Contudo, a quantidade de espaço em memória necessária para armazená-la chega a ser, em alguns casos, proibitiva. Mesmo usando a forma compactada de armazenamento mencionada e considerando-a como uma matriz triangular, o espaço requisitado continua a ser da ordem de N^2 .

Outra desvantagem da representação é o fato da dificuldade que surge, em alguns casos, para a inclusão ou exclusão de vértices no grafo, uma vez que isso pode acarretar uma mudan-

ça de dimensão da matriz. Contudo, incluir ou excluir arestas é uma operação simples.

Observamos também que esta matriz pode ser usada para armazenar informação adicional sobre o grafo, por exemplo, cada a_{ij} representando a distância entre os vértices v_i e v_j do grafo.

II-4. Referências

A idéia desta representação aparece freqüentemente na literatura sobre grafos e computação em geral, por exemplo em Harary⁶, Snow & Scoins¹⁶, Scoins¹⁴, Wells¹⁸, Berge², Berz-tiss³, Obruca¹⁰, Szwarcfiter¹⁷.

CAPÍTULO III

MATRIZ DE INCIDÊNCIAIII-1. Definição da Representação

Seja G um grafo cujos N vértices e M arestas estão rotulados. A matriz de incidência de G é a matriz $N \times M$

$$B(G) = (b_{ij}) , i=1, \dots, N , j=1, \dots, M$$

definida por :

$$b_{ij} = \begin{cases} 1, & \text{se o vértice } v_i \text{ é incidente à aresta } e_j \text{ (isto é, se } e_j = (v_i, v_k) \text{ para algum } k) , \\ 0, & \text{caso contrário .} \end{cases}$$

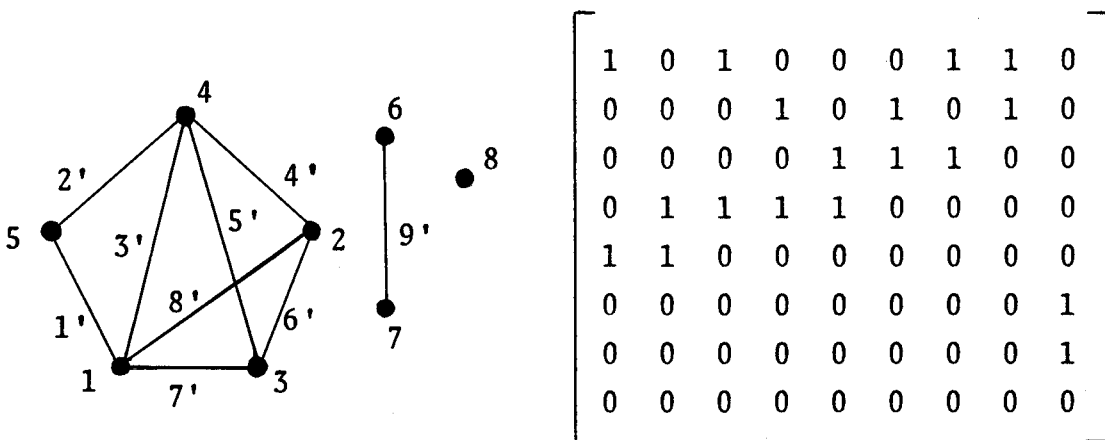
III-2. Exemplo

Figura III-1. Um grafo e uma representação por matriz de incidência.

III-3. Propriedades Básicas e Comentários

Um vértice isolado não é adjacente a nenhuma aresta, portanto,

$$v_i \text{ é isolado } \iff b_{ij} = 0 \text{ para todo } j=1, \dots, M .$$

Uma vez que qualquer aresta é incidente a exatamente dois vértices, toda coluna da matriz tem exatamente dois elementos cujo valor é 1. Assim, a quantidade de 1's na matriz é o dobro do número de colunas, isto é, o dobro do número de arestas do grafo, como na matriz de adjacência. Deste fato conclui-se que pode ser suprimida uma linha da matriz de incidência, pois ela pode ser determinada a partir das restantes $N-1$ linhas, resultando daí uma matriz $N-1 \times M$. No entanto, tal eliminação produz uma economia muito pequena de espaço, se comparada com o decorrente aumento de operações para a manipulação do grafo.

O espaço necessário para armazenar o grafo corresponde a $M \times N$ operações binárias. Da mesma forma que na matriz de adjacência, é também possível agora agrupar elementos armazenando-os sob a forma de bits de palavra.

Também é possível redefinir a matriz de incidência de forma a representar multigrafos e/ou pseudografos, fazendo

$$b_{ij} = \begin{cases} 1, & \text{se o vértice } v_i \text{ é incidente à aresta} \\ & e_j \text{ (isto é, se } e_j = (v_i, v_k) \text{ para al} \\ & \text{gum } k) , \\ 2, & \text{se existe um laço no vértice } v_i \text{ (is-} \\ & \text{to é, se } e_j = (v_i, v_i)), \\ 0, & \text{caso contrário .} \end{cases}$$

A figura III-2 a seguir mostra um exemplo da aplicação dessa redefinição da matriz de incidência.

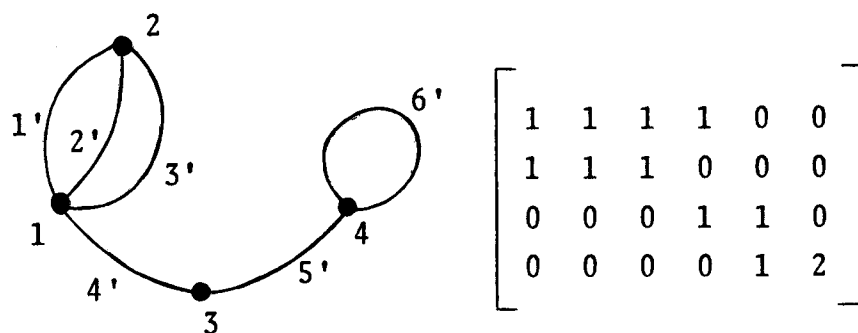


Figura III-2. Um pseudomultigrafo e uma representação por matriz de incidência.

Desta forma permanece válida a afirmação da soma dos elementos em cada coluna da matriz ser 2. Obviamente, o número de laços no pseudografo é o número de elementos com valor 2 na matriz. Para permitir que haja mais de um laço no mesmo vértice, a definição da representação pode ser alterada.

A representação de grafos direcionados é possível com uma alteração na definição. A matriz de incidência de um grafo direcionado D pode ser definida por

$$B(D) = (b_{ij}), \text{ com}$$

$$b_{ij} = \begin{cases} 1, & \text{se a aresta } e_j \text{ começa no vértice } v_i, \\ -1, & \text{se a aresta } e_j \text{ termina no vértice } v_i, \\ 0, & \text{se a aresta } e_j \text{ não é incidente ao vértice } v_i. \end{cases}$$

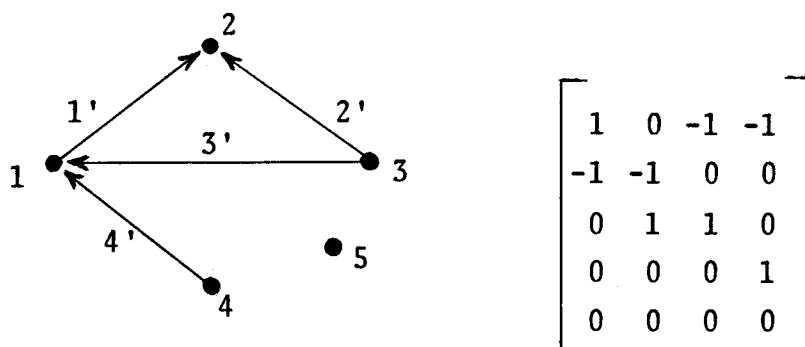


Figura III-3. Um digrafo e uma representação por matriz de incidência.

Observamos que continua válida a afirmação de que , se v_i é um vértice isolado do digrafo, então a i -ésima linha da matriz é toda nula.

Uma vez que cada aresta possui um vértice fonte e um sumidouro, cada coluna tem exatamente um elemento com valor -1 e um elemento com valor $+1$.

De um modo geral, em um grafo com N vértices e M arestas temos $M \gg N$. Uma vez que o número de elementos com valor 1 é o mesmo nas matrizes de adjacência e de incidência, a densidade de 1 's é maior na primeira (proporção de 1 's em relação ao número total de elementos). Portanto, pode-se dizer que a matriz de incidência consome mais memória do que a matriz de adjacência.

Como na representação por matriz de adjacência, existe também aqui o problema da dificuldade das operações de inserção e remoção. Na presente representação essas operações podem ser difíceis (em alguns casos) tanto para vértices como para arestas.

III-4. Referências

Esta representação aparece freqüentemente na literatura. Ver, por exemplo, Harary⁶, Scoins¹⁴, Wells¹⁸, Berge², Obruca¹⁰, Szwarcfiter¹⁷.

CAPÍTULO IV

MATRIZ DE POSICIONAMENTOIV-1. Definição da Representação

Seja D um digrafo, cujo conjunto de vértices é V e cujo conjunto de arestas é X . Seja M o número de arestas de D .

A matriz de posicionamento de D é a matriz $M \times M$

$$C(D) = (c_{ij}) , i, j = 1, \dots, M ,$$

cujos elementos c_{ij} procuram refletir a posição relativa das arestas i e j , e são definidos da maneira como se segue.

Associe um rótulo a cada aresta de X . Associe também uma ordem a esses rótulos.

Se $v_i R v_j$ denota a aresta \underline{e} , onde $v_i, v_j \in V$ e $\underline{e} \in X$, então defina

$$\begin{aligned} \text{in}(\underline{e}) &= v_i , \\ \text{fin}(\underline{e}) &= v_j . \end{aligned}$$

Suponha que a ordenação das arestas rotuladas é e_1, \dots, e_M . Para facilitar a notação, vamos denotar cada aresta e_i pelo seu subscrito \underline{i} . Os elementos da matriz de posicionamento podem ser obtidos da seguinte maneira :

Para $i < j$,

$$c_{ij} = \begin{cases} 1, & \text{se } \text{in}(i) = \text{in}(j), \\ -1, & \text{se } \text{in}(i) = \text{fin}(j), \\ 0, & \text{caso contrário;} \end{cases}$$

$$c_{ji} = \begin{cases} -1, & \text{se } \text{fin}(i) = \text{in}(j), \\ 1, & \text{se } \text{fin}(i) = \text{fin}(j), \\ 0, & \text{caso contrário.} \end{cases}$$

Esta definição corresponde à seguinte posição relativa de duas arestas i e j , com $i < j$:

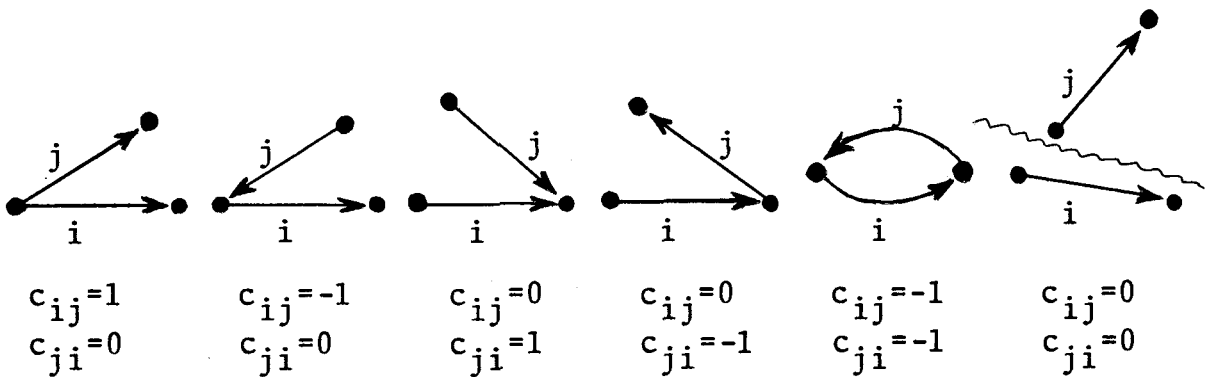


Figura IV-1. Posições relativas de duas arestas e seus elementos correspondentes na matriz de posicionamento.

IV-2. Exemplo

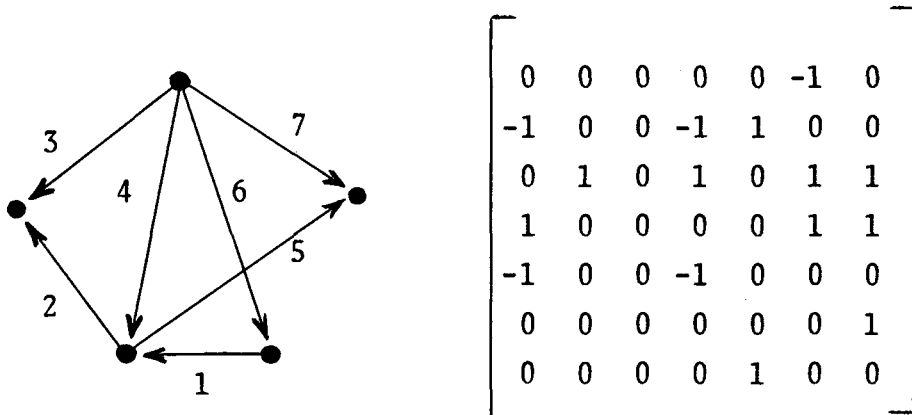


Figura IV-2. Um digrafo e uma representação por matriz de posicionamento.

IV-3. Propriedades Básicas e Comentários

É trivial ver que, dado um grafo direcionado D , existe uma e apenas uma representação de D por matriz de posicionamento, para cada ordenação de suas arestas considerada. Por outro lado, essa representação determina unicamente um digrafo D , obviamente o mesmo D que gerou a representação.

Esta representação refere-se diretamente às arestas do digrafo, e não aos seus vértices — não havendo, portanto, uma maneira de se representar vértices isolados. Este problema pode ser contornado através da associação, a cada vértice isolado, de um laço, e representando o pseudografo assim obtido da maneira descrita mais adiante. Contudo, isso implica em uma modificação na estrutura do digrafo, o que nem sempre é recomen-dável.

Seja e_i uma aresta do digrafo D . Então $e_i=(v',v'')$, onde v',v'' são vértices de D . Para $j = 1, \dots, M$, temos então :

$$\text{Grau de entrada de } v' = \text{Card} \{ c_{ij} \mid c_{ij}=1 \},$$

$$\begin{aligned} \text{Grau de saída de } v' = 1 + \text{Card} \{ c_{ij} \mid i < j \text{ e } c_{ij}=1 \} + \\ + \text{Card} \{ c_{ji} \mid j < i \text{ e } c_{ji}=1 \}, \end{aligned}$$

$$\begin{aligned} \text{Grau de entrada de } v'' = 1 + \text{Card} \{ c_{ji} \mid i < j \text{ e } c_{ji}=1 \} + \\ + \text{Card} \{ c_{ij} \mid j < i \text{ e } c_{ij}=1 \}, \end{aligned}$$

$$\text{Grau de saída de } v'' = \text{Card} \{ c_{ji} \mid c_{ji}=-1 \}.$$

Diremos que um grafo é n-percorrível se existir um caminho orientado de comprimento n . Obviamente, qualquer digrafo que possua ao menos uma aresta é 1-percorrível. Um digrafo é 1-percorrível, e não n -percorrível para qualquer $n > 1$ (isto significa que todo vértice do digrafo tem grau de saída zero ou grau de entrada zero) se e somente se sua matriz de posicionamento não possui nenhum elemento com valor -1 .

Seja v_1 um vértice do digrafo. O número de elementos com valor 1 gerados por este vértice é dado por

$$\begin{aligned} & \frac{\text{gr.saída } v_1 \cdot (\text{gr.saída } v_1 - 1)}{2} + \\ & + \frac{\text{gr.entrada } v_1 \cdot (\text{gr.entrada } v_1 - 1)}{2} . \end{aligned}$$

O número total de 1's na matriz é a soma dos 1's gerados por cada vértice do digrafo, ou seja,

$$\begin{aligned} & \sum_{i=1}^N \left[\frac{\text{gr.saída } v_i \cdot (\text{gr.saída } v_i - 1)}{2} + \frac{\text{gr.entrada } v_i \cdot (\text{gr.entrada } v_i - 1)}{2} \right] = \\ & = \frac{1}{2} \sum_{i=1}^N \left[(\text{gr.saída } v_i)^2 + (\text{gr.entrada } v_i)^2 - (\text{gr.saída } v_i + \text{gr.entrada } v_i) \right] = \\ & = -M + \frac{1}{2} \sum_{i=1}^N \left[(\text{gr.saída } v_i)^2 + (\text{gr.entrada } v_i)^2 \right] , \end{aligned}$$

onde N é o número de vértices e M o número de arestas.

O número de -1's gerados por um vértice v_1 é
 $(\text{gr.saída } v_1) \cdot (\text{gr.entrada } v_1)$.

Logo, o número total de -1's na matriz é dado por

$$\sum_{i=1}^N (\text{gr.saída } v_i) \cdot (\text{gr.entrada } v_i) .$$

Concluimos, assim, que o número total de elementos não nulos na matriz é dado pela soma dos dois resultados anteriores, ou seja,

$$\begin{aligned} & -M + \frac{1}{2} \sum_{i=1}^N \left[(\text{gr.saída } v_i)^2 + (\text{gr.entrada } v_i)^2 + 2 \cdot (\text{gr.saída } v_i) \cdot (\text{gr.entrada } v_i) \right] \\ & = -M + \frac{1}{2} \sum_{i=1}^N (\text{gr.saída } v_i + \text{gr.entrada } v_i)^2 . \end{aligned}$$

Chamando $\text{grau}(v_i) = \text{gr.saída } v_i + \text{gr.entrada } v_i$, o número total de elementos não nulos na matriz será dado por

$$-M + \frac{1}{2} \sum_{i=1}^N \text{grau}^2(v_i) .$$

O grafo de linha de um grafo G , anotado $L(G)$, pode ser definido da seguinte forma: os vértices de $L(G)$ são as arestas de G , e dois vértices de $L(G)$ são adjacentes se as arestas correspondentes de G o forem. Podemos ver que a matriz de posicionamento de D pode representar a matriz de adjacência de $L(D)$, no sentido de que c_{ij} ou c_{ji} será não nulo sse os elementos correspondentes da matriz de adjacência de $L(D)$ forem não nulos.

Intercambiando os valores dos elementos c_{ij} e c_{ji} na matriz de posicionamento de um digrafo D , obteremos uma outra matriz, que é a matriz de posicionamento do digrafo obtido a partir de D pela inversão do sentido de todas as suas arestas.

Para representar pseudografos e/ou multigrafos direcionados, podemos usar a mesma definição, adicionando estas convenções sobre as novas situações :

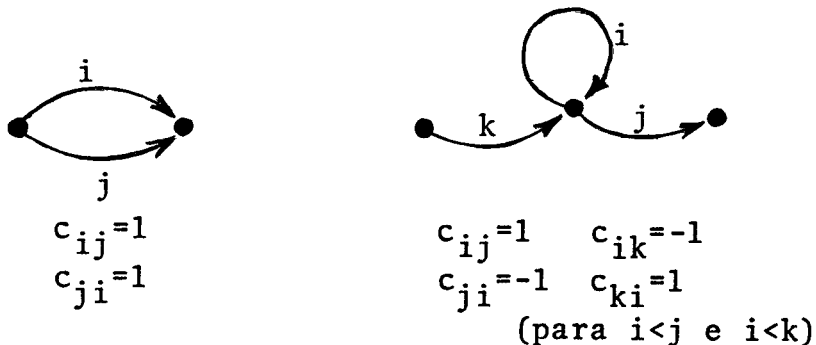


Figura IV-3. Elementos da matriz de posicionamento correspondentes a laços e arestas múltiplas.

Devemos, no entanto, tomar o cuidado de ter sempre $i < j$ e $i < k$, e de ter no máximo um laço em cada vértice (a menos

que seja usada a diagonal principal para armazenar informação sobre o número de laços). A figura IV-4 mostra um exemplo da representação de um pseudomultigrafo direcionado por matriz de posicionamento.

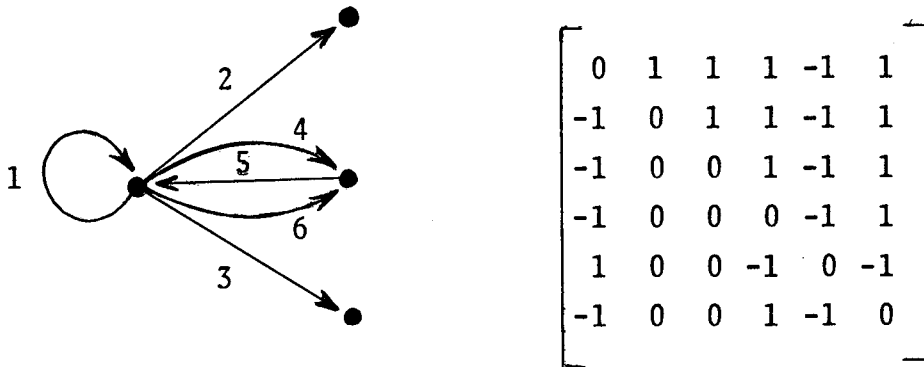


Figura IV-4. Um pseudomultigrafo direcionado e uma representação por matriz de posicionamento.

A matriz de posicionamento requer o armazenamento de M^2 informações, onde M é o número de arestas do digrafo. Como, na prática, M pode ser muito maior do que N , o espaço de armazenamento necessário é, em geral, muito grande. Para diminuir a quantidade de espaço gasto, podem ser usadas técnicas de compactação, mas ainda assim o espaço necessário será proporcional a M^2 .

IV-4. Referências

Esta representação aparece em Szwarcfiter¹⁷, e foi a única representação encontrada na literatura consultada que utilizava uma matriz $M \times M$, onde M é o número de arestas do digrafo (uma outra representação por matriz $M \times M$, a matriz orientada graficamente apresentada no capítulo V, consiste em uma extensão desta). Wells¹⁸ apresenta uma outra matriz $M \times M$ que não foi considerada por nós uma representação por não identificar unicamente o grafo.

CAPÍTULO V

REPRESENTAÇÃO ORIENTADA GRAFICAMENTE

De um modo geral, as representações para grafos (direcionados ou não) apresentam as seguintes características :

- (i) As representações dependem de uma rotulação dos vértices ou das arestas, ou de ambos. Desta forma, diferentes rotulações do mesmo grafo correspondem a diferentes representações.
- (ii) Grafos isomorfos, rotulados da mesma maneira , têm o mesmo formato de representação. No entanto, um mesmo grafo pode ser desenhado de diversas maneiras distintas. Às vezes pode ser in-teressante poder distinguir, pela representação do grafo, qual o aspecto gráfico que está representado.

A presente representação é uma tentativa de solução deste último problema. Podemos dizer que, praticamente, para cada aspecto gráfico de um digrafo, obtém-se uma representação diferente. É quase impossível termos uma representação para cada desenho diferente do digrafo pois desenhos diferentes podem corresponder a um mesmo aspecto gráfico.

V-1. Definição da Representação

A representação orientada graficamente é uma extensão da representação por matriz de posicionamento (capítulo IV).

Seja D um digrafo, cujo conjunto de vértices é V e cujo conjunto de arestas é X . Seja M o número de arestas de D .

A matriz orientada graficamente é a matriz $M \times M$

$$W(D) = (w_{ij}) , i, j = 1, \dots, M .$$

Suponhamos que as M arestas do digrafo estão rotuladas e_1, \dots, e_M .

Se e_i e e_j são duas arestas tais que $i < j$, temos : se e_i e e_j são adjacentes, então os elementos w_{ij} e w_{ji} são definidos como para a matriz de posicionamento; caso contrário, se e_i e e_j se interceptam em algum ponto que não nos extremos, $w_{ij} = w_{ji} = "*" ;$ se e_i e e_j não se interceptam, $w_{ij} = w_{ji} = 0$. Também temos $w_{ii} = 0$, para $i = 1, \dots, M$.

Assim, para $i < j$ temos as seguintes situações possíveis :

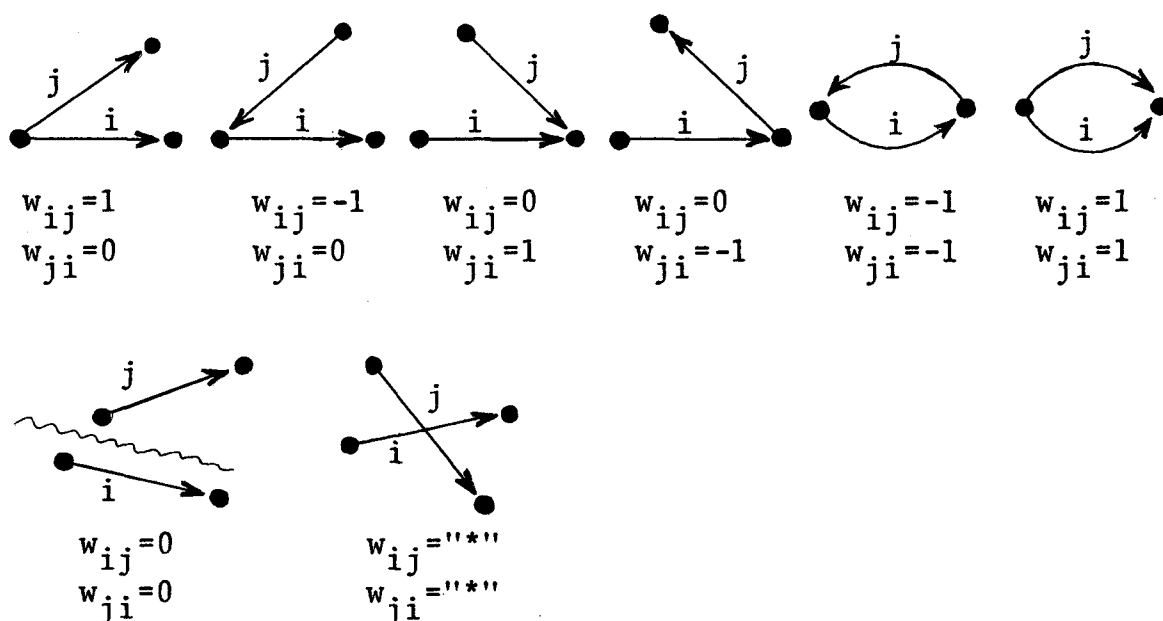


Figura V-1. Posições relativas de duas arestas e seus elementos correspondentes na matriz orientada graficamente.

V-2. Exemplos

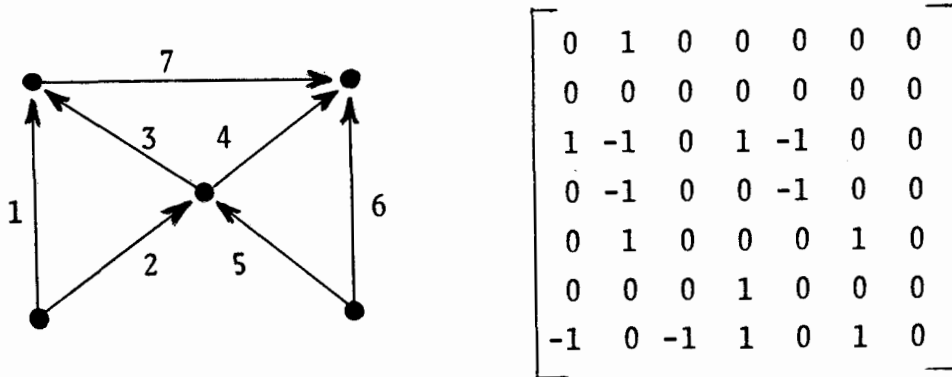


Figura V-2. Um digrafo e uma representação por matriz orientada graficamente.

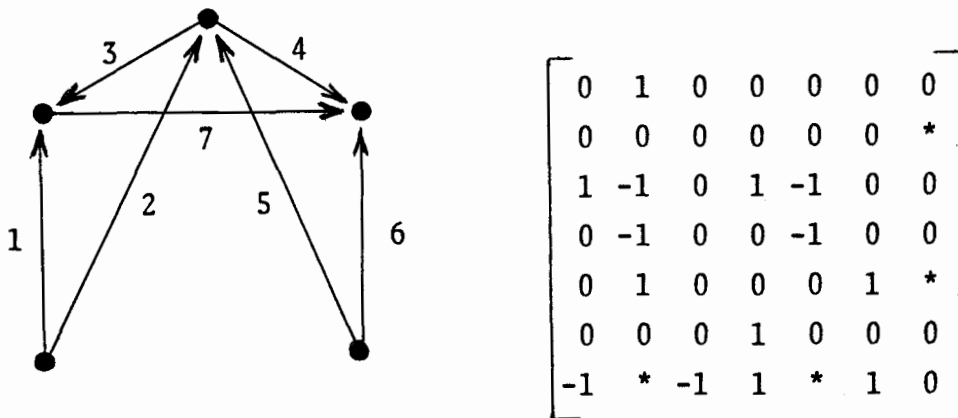


Figura V-3. Um digrafo isomorfo ao digrafo da figura V-2 e uma representação por matriz orientada graficamente.

V-3. Propriedades Básicas e Comentários

Podemos observar que os dois digrafos das figuras V-2 e V-3 são isomorfos e correspondem a diferentes aspectos gráficos. A representação orientada graficamente fornece matrizes diferentes para cada um deles. No entanto, se considerarmos o digrafo da figura V-4, veremos que ele é isomorfo aos dois anteriores e que pode ter uma representação idêntica à do digrafo da figura V-2. Isso, contudo, não invalida a idéia desta representação.

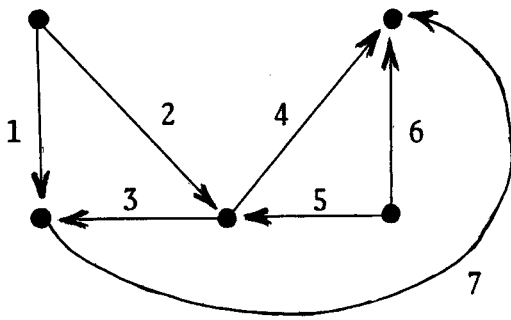


Figura V-4. Um digrafo isomorfo aos digrafos das figuras V-2 e V-3.

O espaço requisitado por esta representação corresponde ao armazenamento de M^2 informações quaternárias. Mas, de um modo geral, a matriz deverá ter um grande número de elementos nulos, sendo portanto possível construir uma "estrutura graficamente orientada" que seria basicamente uma representação da matriz orientada graficamente como matriz esparsa.

Se a matriz orientada graficamente não possuir nenhum elemento "*", então o digrafo correspondente será planar. Não obstante a sua trivialidade, esta é a única representação apresentada neste trabalho através da qual obtemos alguma informação sobre planaridade apenas olhando a representação.

Para saber qual a representação orientada graficamente que possui o menor número de "*"s, dentre as diversas representações correspondentes aos diversos aspectos gráficos de um digrafo, teríamos que saber o nº de cruzamento do digrafo. No entanto, até o presente esse número não foi determinado nem mesmo para grafos completos, tendo sido obtidas apenas cotas superiores.

Sendo a presente representação uma pequena extensão da matriz de posicionamento, as propriedades desta última permanecem válidas para a representação orientada graficamente. Veja o capítulo IV para obter maiores detalhes.

V-4. Referências

Na literatura consultada, esta representação aparece apenas em Szwarcfiter¹⁷, que a definiu.

CAPÍTULO VI

ESTRUTURA DE ADJACÊNCIAVI-1. Definição da Representação

Tomemos a matriz de adjacência (capítulo II) de um grafo G e representemos essa matriz através da construção, para cada vértice do grafo, da lista dos vértices adjacentes ao vértice considerado. Isto significa que, para cada linha da matriz, vamos ligar todos os vértices cujos elementos correspondentes nessa linha têm valor 1. Cada uma dessas listas é dita a lista de adjacência do vértice ao qual se refere, e o conjunto de todas as listas de adjacência, uma para cada vértice do grafo, compõe a estrutura de adjacência.

Assim, se v_i é um vértice do grafo G ,

lista de adjacência de $v_i = \{ \text{vértices } v_j \mid v_i R v_j \}$,

estrutura de adjacência de $G =$

$= \{ \text{listas de adjacência de } v_i \mid v_i \text{ é vértice de } G \} .$

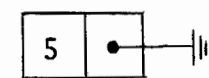
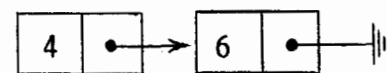
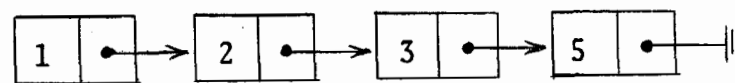
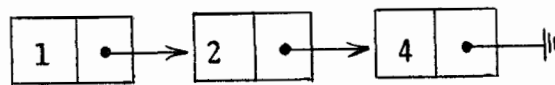
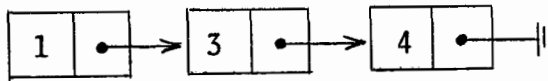
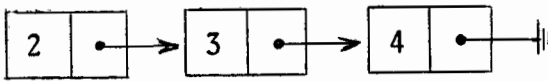
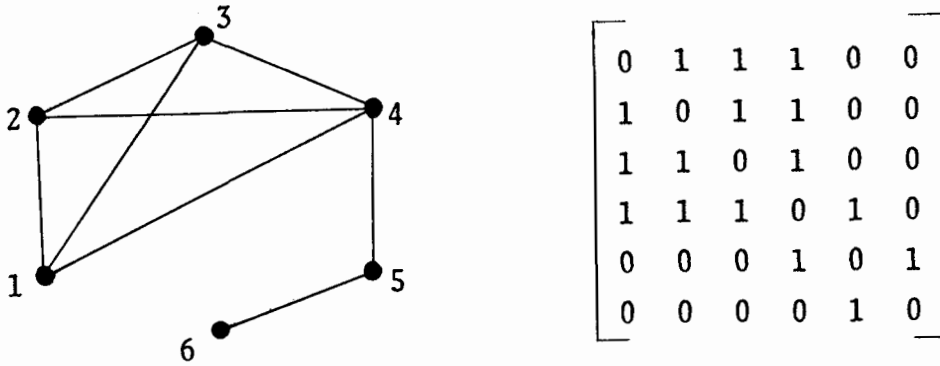
VI-2. Exemplo

Figura VI-1. Um grafo, uma matriz de adjacência e a correspondente estrutura de adjacência.

VI-3. Propriedades Básicas e Comentários

Observamos que esta estrutura pode ser obtida a partir da matriz de adjacência por uma transformação bem definida. E reciprocamente, dada uma estrutura de adjacência, pode-se obter de maneira unívoca a matriz de adjacência. Todas as propriedades válidas para a matriz de adjacência têm um equivalente para a estrutura de adjacência.

Assim, um vértice isolado do grafo G é representado, na matriz de adjacência, por uma linha nula, e na estrutura de adjacência por uma lista de adjacência vazia. Para termos manipulação eficiente da estrutura nessa situação, é conveniente adicionar um nó extra a cada lista de adjacência, chamado de cabeça de linha, apontando para o primeiro vértice da lista de adjacência correspondente. A figura VI-2 mostra um grafo representado por uma estrutura de adjacência com cabeças de linha.

Para implementação em computador, o conjunto das cabeças de linha pode ser representado por um vetor com N elementos (onde N é o número de vértices do grafo), e as listas de adjacência por um conjunto de listas ligadas compartilhando a mesma área de espaço disponível.

Olhando a figura anterior, verificamos que somente manipulações na mesma linha podem ser feitas facilmente. Se forem necessárias também manipulações ao longo de uma coluna, torna-se útil ligar os nós que correspondem a um mesmo vértice em todas as listas de adjacência. A estrutura resultante corresponde à representação geral ligada de matrizes esparsas. A figura VI-3 mostra um exemplo desta estrutura, correspondente ao grafo da figura VI-2.

Caso seja necessário armazenar uma informação adicional para cada vértice, podemos acrescentar um campo a mais em cada nó da estrutura, apontando para a informação correspondente.

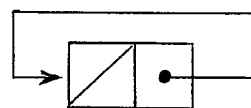
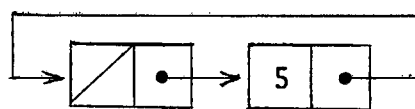
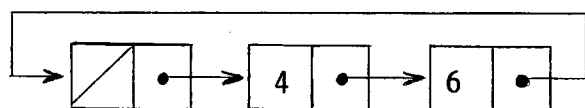
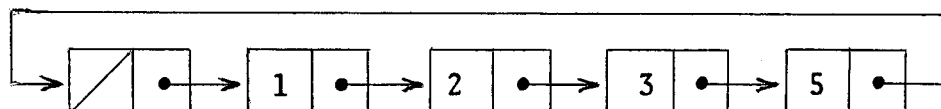
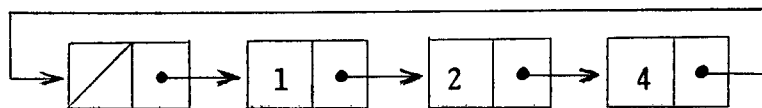
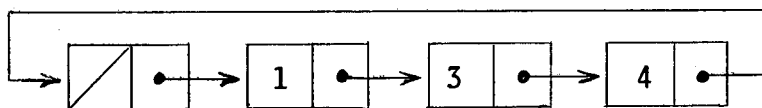
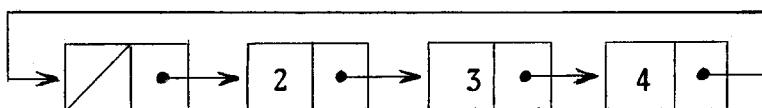
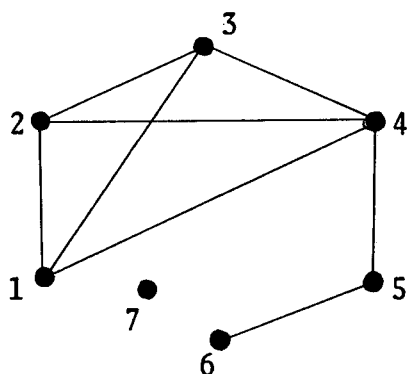


Figura VI-2. Um grafo e uma estrutura de adjacência com cabeças de linha.

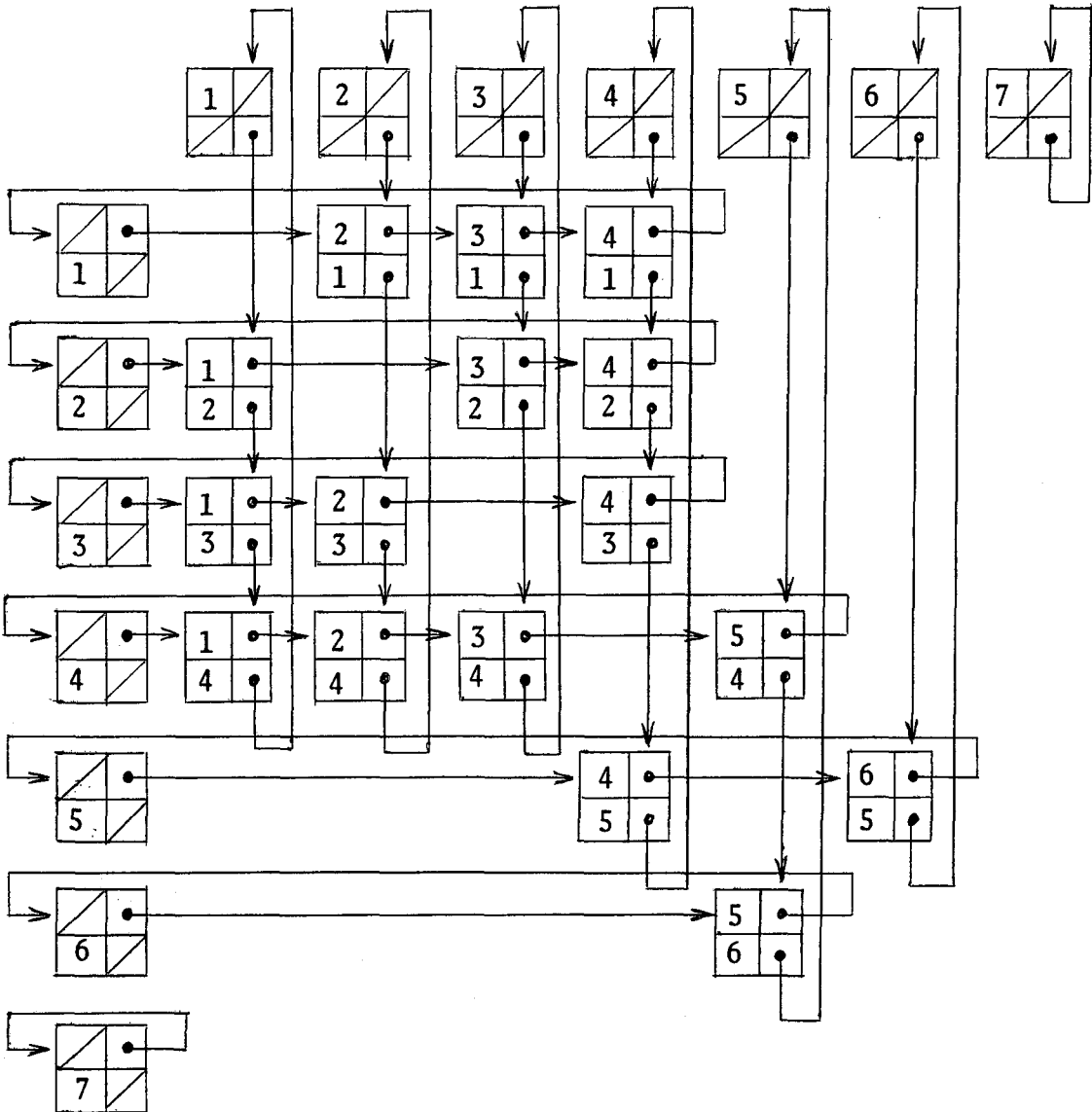


Figura VI-3. Uma estrutura de adjacência com cabeças de linha e cabeças de coluna.

Podemos observar que, se v_i e v_j são vértices do grafo com $v_i R v_j$, existe um elemento com valor 1 na matriz de adjacência correspondendo ao elemento (i,j) e correspondendo ao elemento (j,i) , pois a matriz é simétrica. Eliminando os elementos (i,j) tais que $i > j$, a estrutura correspondente ao grafo da figura VI-3 seria a mostrada na figura VI-4.

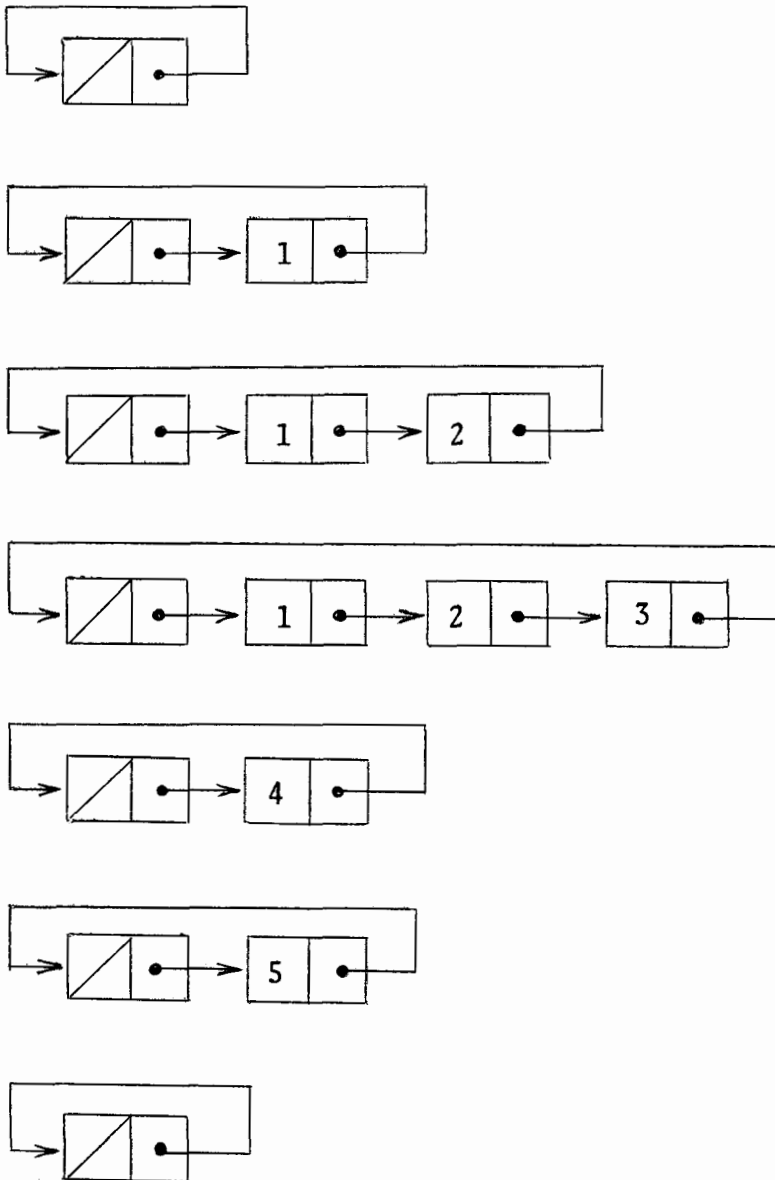


Figura VI-4. Uma estrutura de adjacência triangular com cabeças de linha.

Para verificar agora a existência de uma relação $v_i R v_j$, é necessário procurar pelo elemento \underline{n} na lista correspondente a \underline{m} , onde $n = \min \{i, j\}$ e $m = \max \{i, j\}$.

É fácil ver que a lista de adjacência correspondente ao primeiro vértice é sempre vazia, e portanto pode ser suprimida. Também é possível definir outras maneiras de eliminar uma das duas relações repetidas, resultando em estruturas diferentes.

Observamos ainda que, se adicionarmos cabeças de coluna a este tipo de estrutura, então a procura por uma dada relação $v_i R v_j$ pode ser feita selecionando-se o elemento \underline{n} na linha \underline{m} ou o elemento \underline{m} na coluna \underline{n} , onde $n = \min \{i, j\}$ e $m = \max \{i, j\}$. Neste caso a linha correspondente ao primeiro vértice e a coluna correspondente ao último vértice serão sempre listas vazias.

A simplificação introduzida pela eliminação de uma das relações repetidas de cada aresta geralmente dificulta a manipulação do grafo, aumentando o tempo de processamento. Portanto, às vezes é vantajoso aumentar o número de nós na estrutura, acrescentando um apontador cruzado entre cada par de nós representando uma mesma aresta. A estrutura correspondente à figura VI-2 seria agora a mostrada na figura VI-5.

Esta mesma idéia pode também ser aplicada à estrutura ligada por linha e por coluna, com a adição de um campo extra contendo um elo.

Uma vez que a estrutura de adjacência é encarada aqui como uma maneira particular de representação da matriz de adjacência como uma matriz esparsa, outras representações de matrizes esparsas podem ser consideradas para a obtenção de outras representações diferentes para grafos.

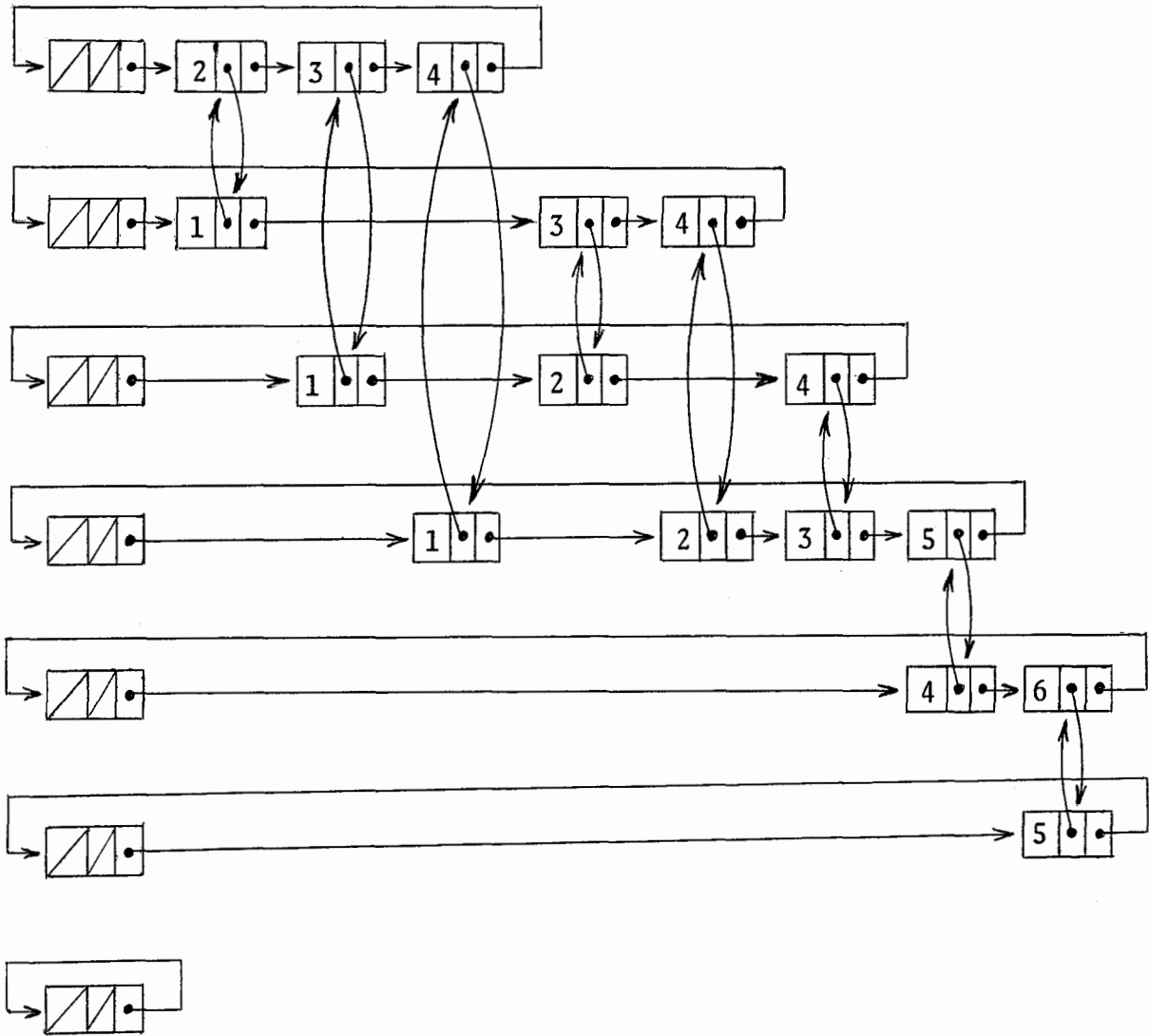


Figura VI-5. Uma estrutura de adjacência com apontadores cruzados e cabeças de linha.

É possível redefinir esta representação de forma a poder ser usada em pseudografos e/ou multigrafos. Podemos adicionar a cada nó da estrutura um campo que informaria quantas arestas existem entre os vértices correspondentes. A informação referente aos laços existentes em um vértice pode ser armazenada em um dos campos não utilizados nas cabeças de linha, como mostra o exemplo da figura VI-6.

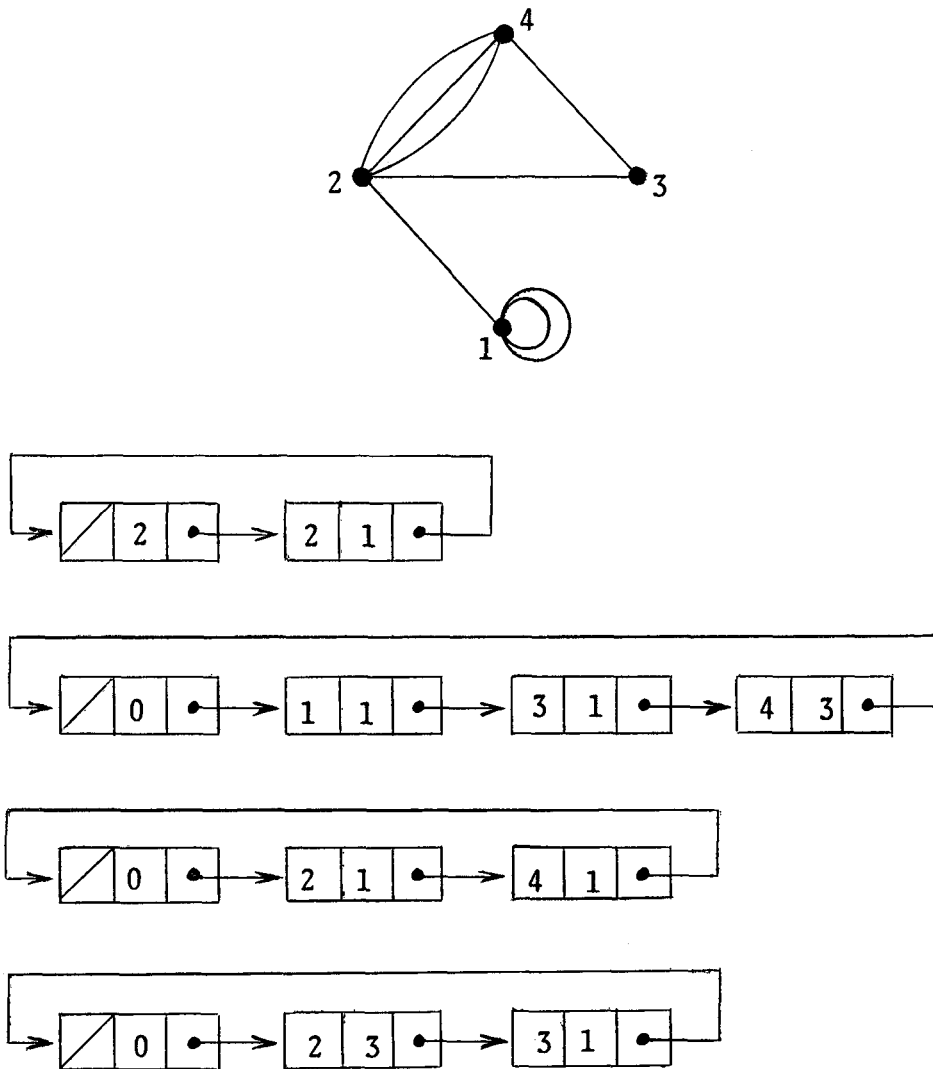
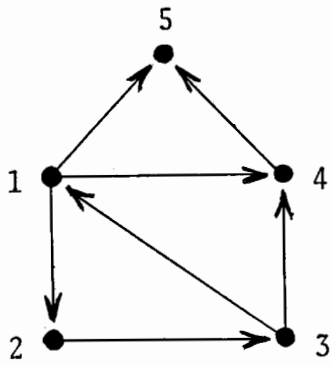


Figura VI-6. Um pseudomultigrafo e uma estrutura de adjacência com cabeças de linha.

A quantidade de espaço necessário para armazenar esta representação pode ser estimada da seguinte maneira: Suponha mos um grafo com N vértices e M arestas. Como a matriz de adjacência possui exatamente $2M$ elementos com valor 1, a estrutura de adjacência, conforme inicialmente definida, terá exatamente $2M$ nós. A adição de nós cabeça implica em um aumento de N ou $2N$ nós, conforme sejam usadas apenas cabeças de linha ou também cabeças de coluna. O espaço total requerido depende da versão escolhida (dentre as apresentadas aqui). A expressão geral do espaço total, contudo, é $c_1N + c_2M$, onde c_1, c_2 são inteiros com $c_1 \geq 0$, $c_2 > 0$. Podemos ver, portanto, que em qualquer caso o espaço é proporcional (linear) a M e N . Apesar dos conjuntos de matrizes de adjacência e de estruturas de adjacência serem isomorfos, há uma importante diferença entre eles, em termos de memória: no primeiro a requisição de espaço é quadrática, e no segundo linear. Aliás, esta é a grande vantagem desta representação sobre a representação por matriz. Outra vantagem é que as operações de inserção e remoção, tanto de vértices como de arestas, podem ser feitas facilmente. Além disso, como a estrutura representa apenas os pares de vértices que estão efetivamente relacionados, a quantidade de itens a serem manipulados em uma operação é geralmente menor, o que resulta, em alguns casos, em uma economia substancial de tempo.

A mesma definição de lista de adjacência e estrutura de adjacência apresentada para grafos não direcionados pode ser aplicada para grafos direcionados. A figura VI-7 é um exemplo.



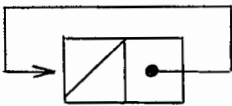
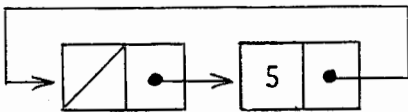
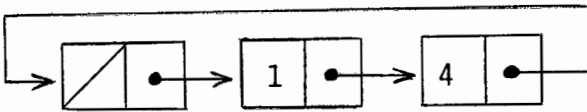
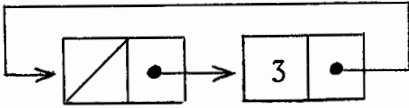
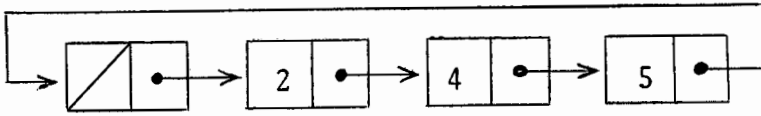
$$\begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$


Figura VI-7. Um digrafo, uma matriz de adjacência e a correspondente estrutura de adjacência, com cabeças de linha.

VI-4. Referências

Esta representação aparece freqüentemente na literatura sobre Computação, mas não na literatura teórica sobre grafos. Os conceitos de lista de adjacência e estrutura de adjacência são encontrados em Hopcroft & Tarjan⁸; a idéia de apontadores cruzados entre vértices relacionados, em Hopcroft & Tarjan⁷; a representação como lista ligada de matrizes esparsas, em Knuth⁹; a idéia da representação é ainda mencionada em Scoins¹⁴. Da maneira como foi definida aqui, esta representação pode ser encontrada em Szwarcfiter¹⁷.

CAPÍTULO VII

ESTRUTURA DE INCIDÊNCIAVII-1. Definição da Representação

Tomemos a matriz de incidência (capítulo III) de um grafo e representemos essa matriz através da construção, para cada vértice do grafo, da lista das arestas que são adjacentes ao vértice considerado. Isto significa que, para cada linha da matriz, vamos ligar todas as arestas cujos elementos correspondentes nessa linha têm valor 1. Cada uma dessas listas é dita a lista de incidência do vértice ao qual se refere, e o conjunto de todas as listas de incidência, uma para cada vértice do grafo, compõe a estrutura de incidência.

Assim, se v_i é um vértice do grafo G ,

$$\begin{aligned} \text{lista de incidência de } v_i &= \\ &= \{ \text{arestas } e \mid e=(v_i, v_j) \text{ para algum vértice } v_j \text{ de } G \}, \end{aligned}$$

estrutura de incidência de G =

$$= \{ \text{listas de incidência de } v_i \mid v_i \text{ é vértice de } G \}.$$

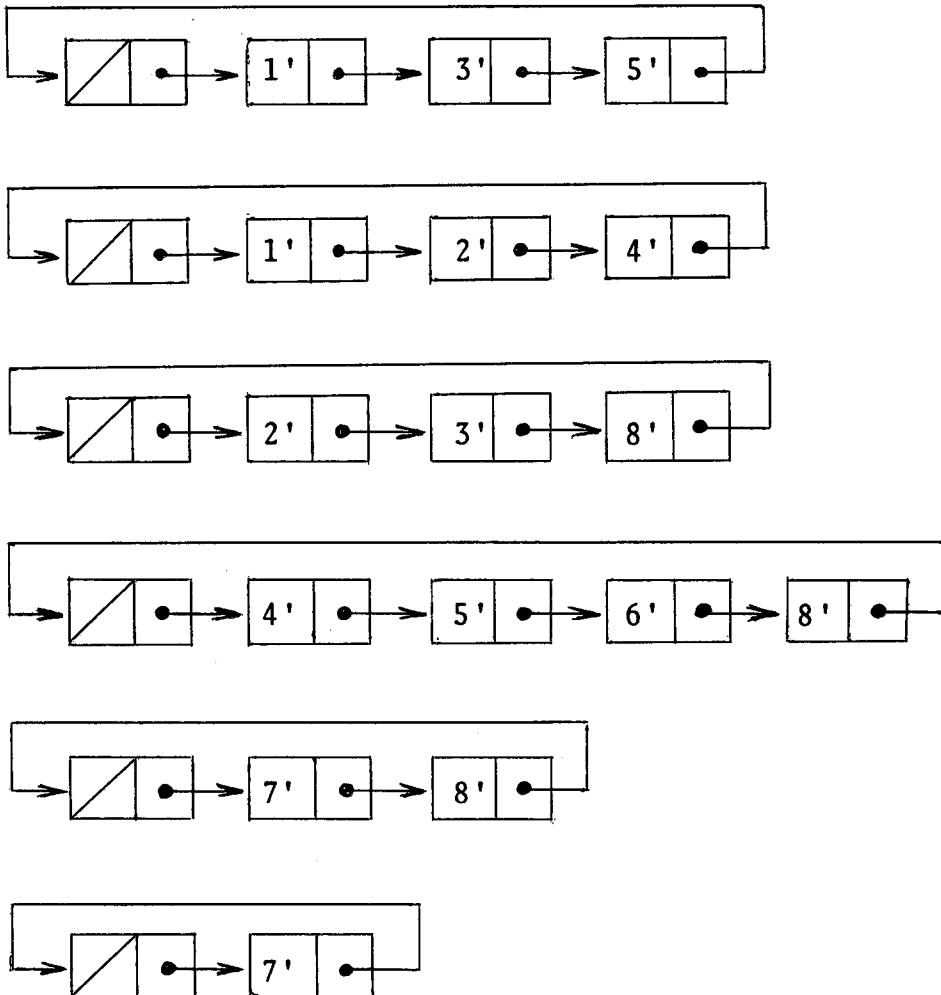
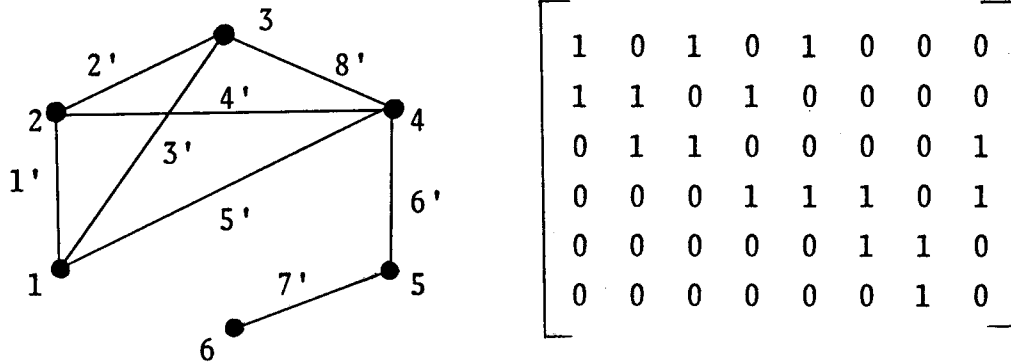
VII-2. Exemplo

Figura VII-1. Um grafo, uma matriz de incidência e a correspondente estrutura de incidência, com cabeças de linha.

VII-3. Propriedades Básicas e Comentários

Se adicionarmos a esta representação uma ligação por coluna, obteremos rapidamente o conjunto dos vértices incidentes a uma determinada aresta.

Da mesma forma que a estrutura de adjacência, a estrutura de incidência precisa de $2M$ nós para armazenar informações, além dos nós cabeça, sendo cada aresta representada duas vezes.

Se for necessário um apontador cruzado entre os pares de arestas idênticas em uma estrutura com cabeças de linha e cabeças de coluna, não será preciso adicionar nenhum campo extra, pois a ligação por coluna corresponde exatamente aos vértices incidentes à mesma aresta do grafo. Neste caso, as listas de coluna não serão circulares; cada uma delas será uma lista duplamente ligada composta de dois nós e de uma cabeça de coluna que aponta para o primeiro nó. A figura VII-2 é um exemplo.

Com o objetivo de tornar mais rápidas as manipulações, é vantajoso acrescentar um novo campo em cada nó, contendo o rótulo do vértice correspondente àquela linha. Isto torna a estrutura de incidência bastante poderosa e fornece meios diretos para a obtenção de algumas características básicas do grafo, como :

- dado um vértice, encontrar todas as arestas incidentes a ele ;
- dada uma aresta, encontrar todos os vértices incidentes a ela ;
- dados dois vértices, saber se estão ou não relacionados ;
- dadas duas arestas, saber se são ou não incidentes a um mesmo vértice ;
- etc.

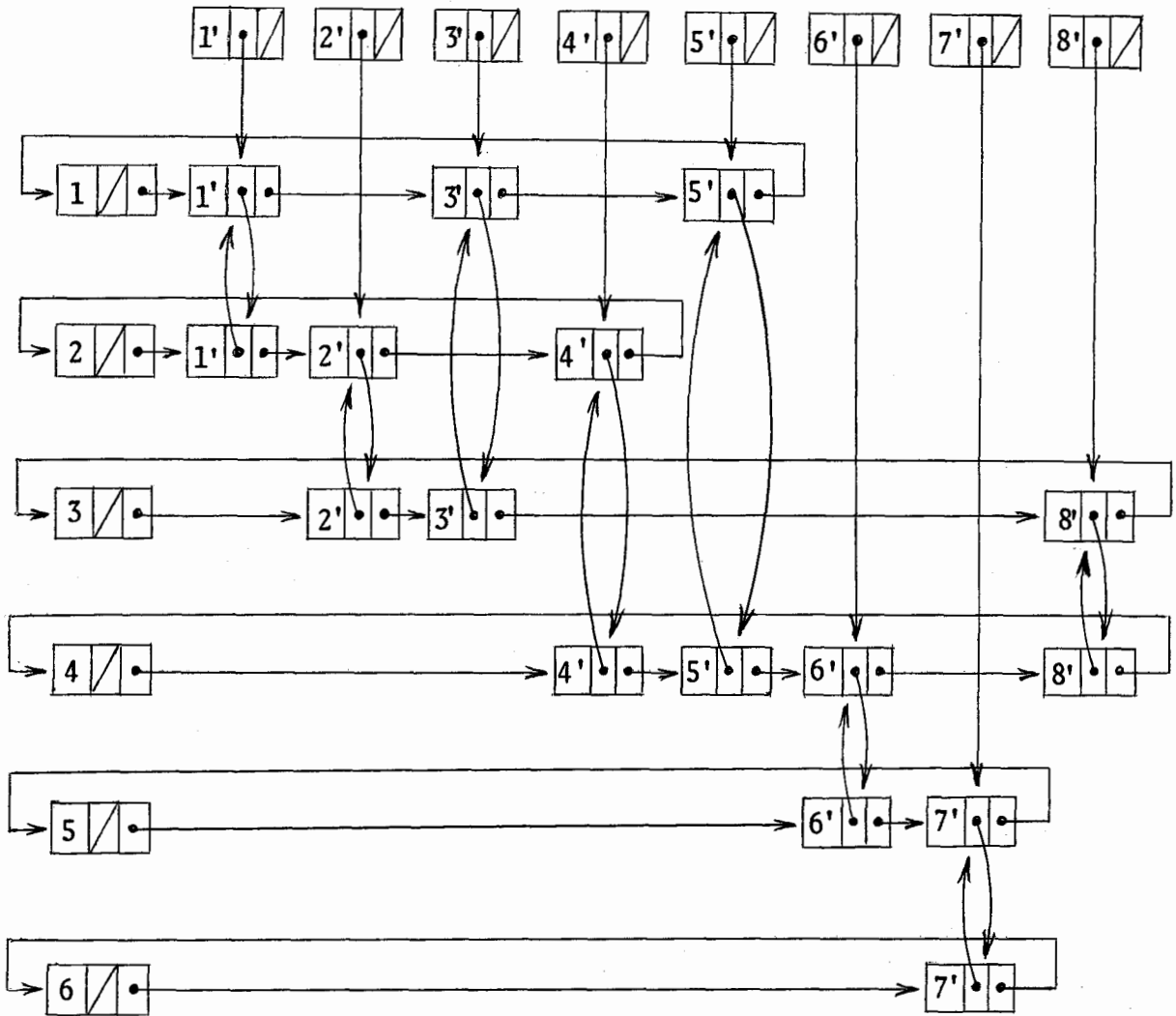


Figura VII-2. Uma estrutura de incidência com cabeças de linha e cabeças de coluna.

A estrutura de incidência é uma espécie de representação por lista da matriz de incidência, da mesma forma que a estrutura de adjacência para a matriz de adjacência. Portanto, propriedades análogas às que já foram deduzidas podem ser obtidas para a presente representação.

VII-4. Referências

Da maneira como foi definida, esta representação pode ser encontrada em Szwarcfiter¹⁷.

CAPÍTULO VIII

ESTRUTURA DE POSICIONAMENTOVIII-1. Definição da Representação

De forma análoga ao que foi feito para as estruturas de adjacência e de incidência, podemos definir a estrutura de posicionamento, considerando a matriz de posicionamento como matriz esparsa e representando-a pela técnica de representação de matrizes esparsas por listas.

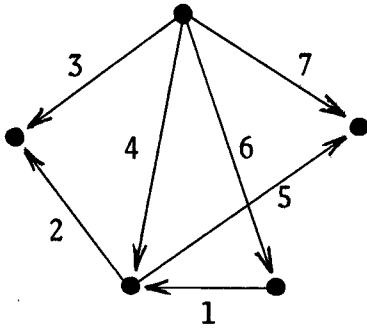
Neste caso, é conveniente considerar a representação de matrizes esparsas por listas utilizando tanto cabeças de linha como cabeças de coluna, uma vez que a determinação completa de uma aresta e_i necessita da linha i e da coluna i .

Também neste caso devemos adicionar um campo de "tag" em cada nó, informando se o nó corresponde a um elemento '+1' ou '-1' na matriz. Na prática, esse "tag" pode ser o sinal de algum dos outros campos do nó.

Seja N o número de vértices e M o número de arestas. O número de nós cabeça nesta representação é $2M$. De acordo com o que foi apresentado para a matriz de posicionamento (capítulo IV), o número de nós na estrutura de posicionamento é

$$\begin{aligned} & -M + \frac{1}{2} \sum_{i=1}^N \text{grau}^2(v_i) + 2M = \\ & = M + \frac{1}{2} \sum_{i=1}^N \text{grau}^2(v_i) . \end{aligned}$$

Uma vez que esta representação é, na verdade, uma outra forma de representação da matriz de posicionamento, as propriedades válidas para esta podem em geral ser traduzidas para aquela.

VIII-2. Exemplo

0	0	0	0	0	-1	0
-1	0	0	-1	1	0	0
0	1	0	1	0	1	1
1	0	0	0	0	1	1
-1	0	0	-1	0	0	0
0	0	0	0	0	0	1
0	0	0	0	1	0	0

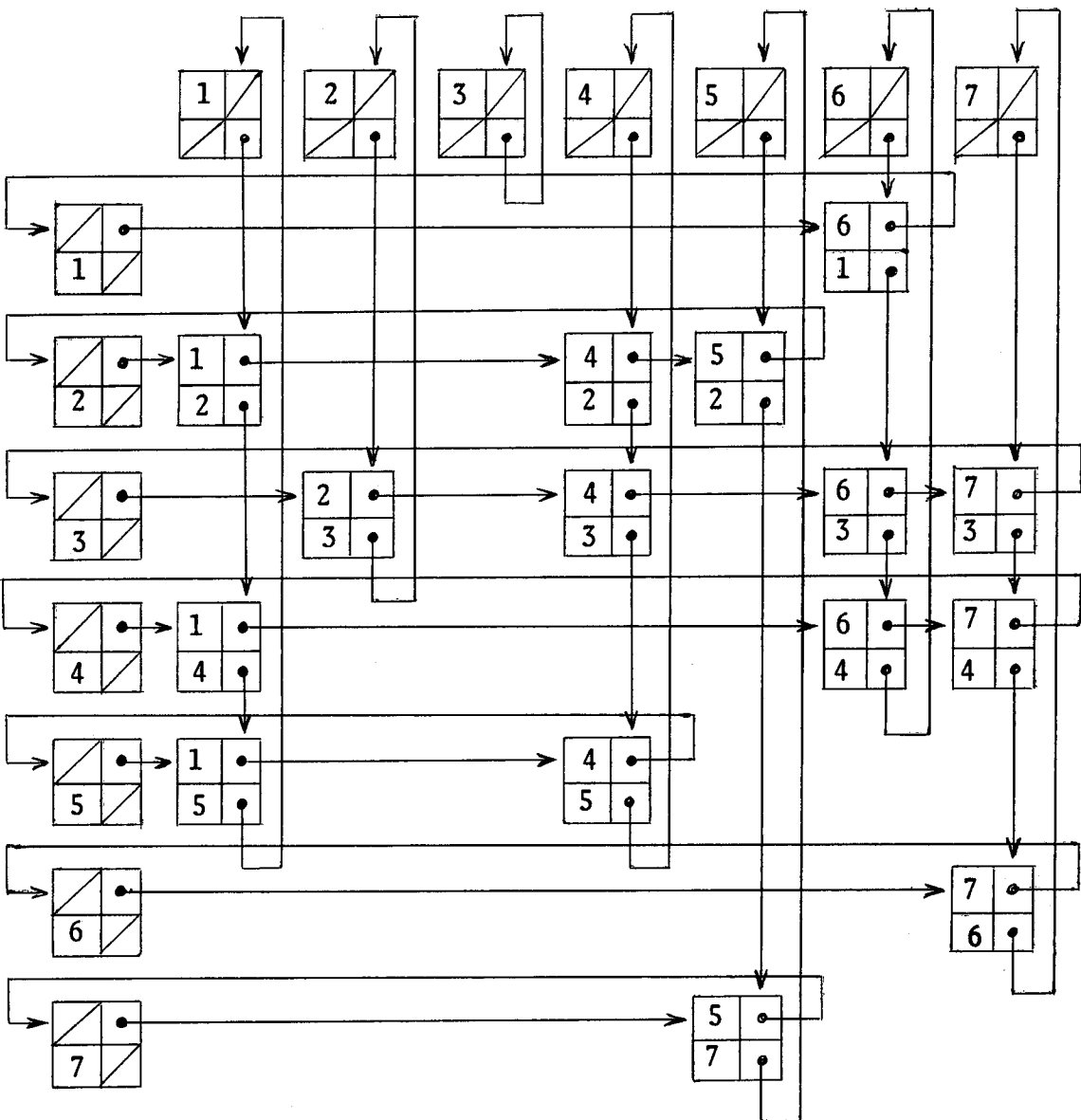


Figura VIII-1. Um digrafo, uma matriz de posicionamento e a correspondente estrutura de posicionamento, com cabeças de linha e cabeças de coluna.

VIII-3. Referências

Esta representação pode ser encontrada em
Szwarcfiter¹⁷.

CAPÍTULO IX

REPRESENTAÇÃO VETORIAL DE ARESTASIX-1. Definição da Representação

Seja D um digrafo, e sejam V o seu conjunto de vértices e X o seu conjunto de arestas.

Se $v_i R v_j$ denota a aresta e ($v_i, v_j \in V$, $e \in X$), então

$$\text{in}(e) = v_i ,$$

$$\text{fin}(e) = v_j .$$

Para cada vértice v do digrafo D , considere os conjuntos :

$$\text{in}^{-1}(v) = \{ e \in X \mid \text{in}(e) = v \} ,$$

$$\text{fin}^{-1}(v) = \{ e \in X \mid \text{fin}(e) = v \} .$$

Defina uma ordenação (qualquer) sobre as arestas de D em cada um dos conjuntos $\text{in}^{-1}(v)$, $\text{fin}^{-1}(v)$, para todo $v \in V$.

Para cada aresta em X , a representação será composta de um nó com o formato

ALINK	ATAG	BLINK	BTAG
-------	------	-------	------

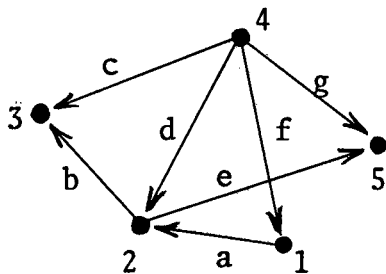
Figura IX-1. Formato de um nó da representação vetorial de arestas.

onde ALINK e BLINK são apontadores para outros nós (arestas), e ATAG e BTAG são "tags" (1 bit) referindo-se à natureza desses apontadores.

Para uma aresta \underline{e} , esses campos podem ser definidos da seguinte forma :

- Tome o vértice v_1 tal que $v_1 = \text{in}(e)$.
- Se \underline{e} for a última aresta (quanto à ordenação) do conjunto $\text{in}^{-1}(v_1)$, então $\text{ATAG}(e) = "-"$. Nesse caso, se $\text{fin}^{-1}(v_1) = \emptyset$ então $\text{ALINK}(e) = \Lambda$; se $\text{fin}^{-1}(v_1) \neq \emptyset$ então, sendo e' a primeira aresta de $\text{fin}^{-1}(v_1)$, $\text{ALINK}(e) = e'$.
- Se \underline{e} não for a última aresta de $\text{in}^{-1}(v_1)$, então, sendo e'' a aresta que sucede \underline{e} , $\text{ATAG}(e) = "+"$ e $\text{ALINK}(e) = e''$.
- Os campos BLINK e BTAG são definidos analogamente, substituindo-se ALINK por BLINK , ATAG por BTAG e in por fin no procedimento acima.

IX-2. Exemplo



$$\begin{array}{ll}
 \text{in}^{-1}(1) = \{a\} & \text{fin}^{-1}(1) = \{f\} \\
 \text{in}^{-1}(2) = \{b, e\} & \text{fin}^{-1}(2) = \{a, d\} \\
 \text{in}^{-1}(3) = \emptyset & \text{fin}^{-1}(3) = \{b, c\} \\
 \text{in}^{-1}(4) = \{c, d, f, g\} & \text{fin}^{-1}(4) = \emptyset \\
 \text{in}^{-1}(5) = \emptyset & \text{fin}^{-1}(5) = \{e, g\}
 \end{array}$$

ARESTA	ALINK	ATAG	BLINK	BTAG
a	f	-	d	+
b	e	+	c	+
c	d	+	Λ	-
d	f	+	b	-
e	a	-	g	+
f	g	+	a	-
g	Λ	-	Λ	-

Figura IX-2. Um digrafo, uma representação vetorial de arestas, e os conjuntos $\text{in}^{-1}(v)$ e $\text{fin}^{-1}(v)$ considerados para a representação.

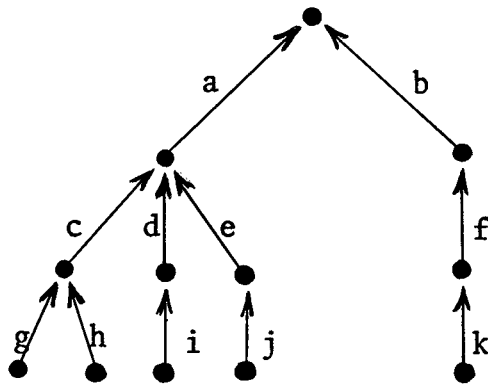
IX-3. Propriedades Básicas e Comentários.

Dado um grafo direcionado D , é trivial ver que, para cada ordenação dos rótulos de suas arestas, existe uma e apenas uma representação vetorial de arestas para esse grafo. Por outro lado, uma representação vetorial de arestas determina unicamente um digrafo, obviamente o mesmo digrafo que havia gerado a representação.

Também pode ser observado que, se para o digrafo D , intercambiarmos ATAG com BTAG e ALINK com BLINK, obteremos um outro digrafo D' cujo conjunto de vértices é o mesmo de D , mas cujo conjunto de arestas corresponde às relações inversas das arestas de D , significando que todas as arestas tiveram suas direções trocadas.

A representação vetorial de arestas para grafos direcionados foi criada por Knuth⁹, a partir da idéia de se generalizar a representação de árvores binárias com alinhave pela direita. Um aspecto interessante da representação é que, no caso do grafo direcionado em questão ser uma árvore, com todas as arestas orientadas de filho para pai, e com a ordenação das arestas sendo sempre da esquerda para a direita, se adicionarmos uma aresta da raiz da árvore para um novo vértice, a representação vetorial de arestas será equivalente à representação da árvore com alinhave pela direita. A figura IX-3 é um exemplo desse fato.

Neste caso, temos $ATAG(e) = "-"$, para toda aresta $e \in X$. Este fato é importante no sentido de que permite o desenho de um sistema para manipulação de digrafos que utilizaria algoritmos eficientes para árvores se reconhecesse que o digrafo em questão era uma árvore.



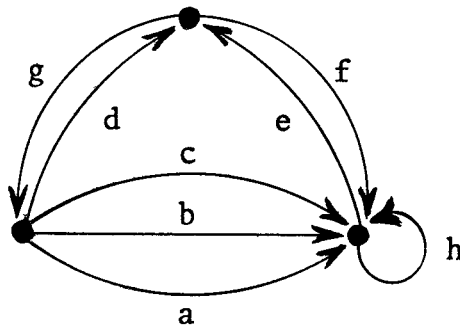
ARESTA	ALINK	ATAG	BLINK	BTAG
a	c	-	b	+
b	f	-	Λ	-
c	g	-	d	+
d	i	-	e	+
e	j	-	a	-
f	k	-	b	-
g	Λ	-	h	+
h	Λ	-	c	-
i	Λ	-	c	-
j	Λ	-	e	-
k	Λ	-	f	-

Figura IX-3. Uma árvore e uma representação vetorial de arestas.

Esta representação refere-se diretamente apenas às arestas do digrafo, e não aos seus vértices — não havendo, portanto, uma maneira de se representar vértices isolados. Este problema pode ser contornado através da associação, a cada vértice isolado, de um laço. Contudo, isso implica em uma modificação na estrutura do digrafo, o que nem sempre é recomendável.

O espaço requisitado por esta representação é diretamente proporcional a M , onde M é o número de arestas do digrafo. Na prática, como os campos de "tag" podem ser os sinais dos apontadores, a memória requisitada é $2M$.

Observa-se que esta mesma estrutura pode ser aplicada a pseudografos e/ou multigrafos direcionados sem qualquer alteração. A figura IX-4 é um exemplo.



ARESTA	ALINK	ATAG	BLINK	BTAG
a	b	+	b	+
b	c	+	c	+
c	d	+	e	+
d	g	-	f	+
e	g	+	h	+
f	h	+	e	-
g	d	-	a	-
h	a	-	f	-

Figura IX-4. Um pseudomultigrafo direcionado e uma representação vetorial de arestas.

IX-4. Referências

Esta representação foi definida e introduzida por Knuth⁹, que procurou generalizar a idéia da representação de árvores com alinhave pela direita. A representação vetorial de arestas aparece também em Szwarcfiter¹⁷.

CAPÍTULO X

PSEUDO-ÁRVOREX-1. Definições Preliminares

As seguintes duas transformações auxiliares serão usadas na representação por pseudo-árvore.

Seja T uma árvore ordenada enraizada. Seja i um nó de T . A transformação $\alpha_{i,j}(T)$ aplica a árvore T na árvore T' , copiando T e inserindo o nó j como o filho mais à direita do nó i . A figura X-1 mostra um exemplo desta transformação.

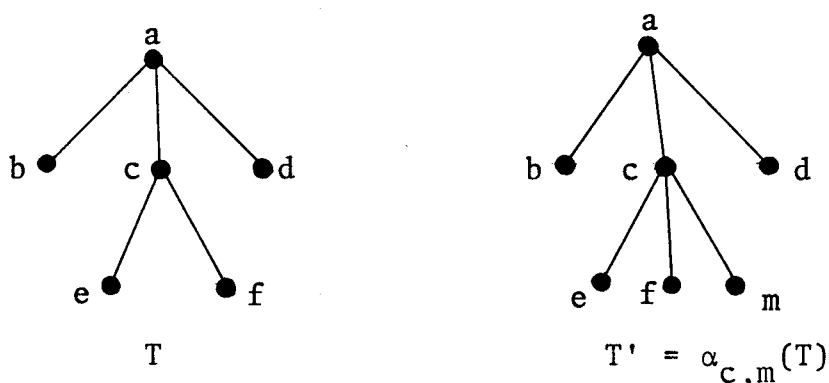


Figura X-1. Exemplo da transformação α .

Sejam T_i , T_j árvores ordenadas enraizadas. Seja k um nó terminal de T_j . A transformação $\beta_k(T_i, T_j)$ aplica essas duas árvores T_i , T_j em uma outra árvore T' , substituindo a folha k de T_j pela árvore T_i . A figura X-2 mostra um exemplo dessa transformação.

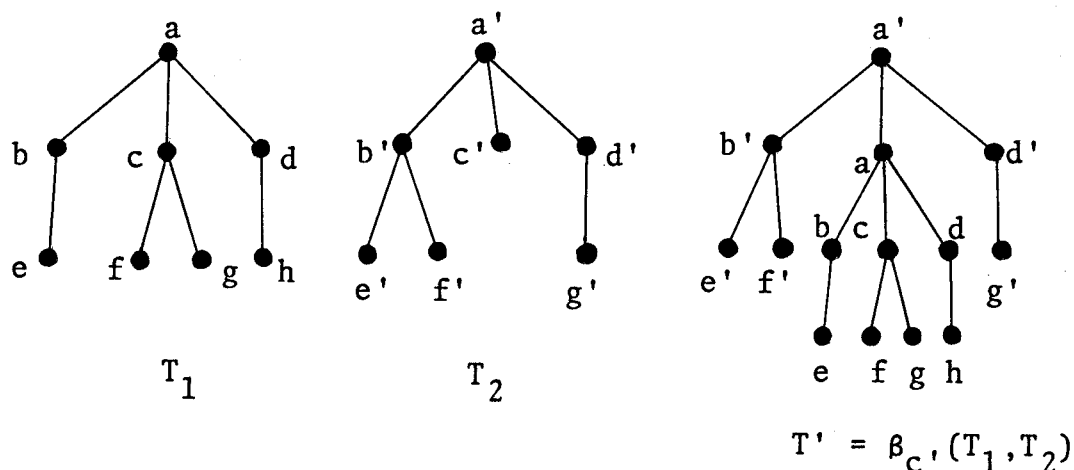


Figura X-2. Exemplo da transformação β .

X-2. Definição de Pseudo-Árvore

Seja T uma árvore ordenada enraizada, onde todos os seus nós carregam informação. Se \underline{i} denota um nó da árvore, então a informação contida em \underline{i} é denotada por $\text{INFO}(\underline{i})$.

Uma estrutura pseudo-árvore pode ser definida recursivamente da seguinte maneira :

- (i) Uma árvore ordenada enraizada T , como descrita acima, onde

$$i \neq j \implies \text{INFO}(i) \neq \text{INFO}(j)$$

é uma pseudo-árvore.

- (ii) Se T é uma pseudo-árvore, e \underline{i} é um nó terminal de T tal que $\text{INFO}(\underline{i}) \neq \text{INFO}(\underline{j})$ para todo nó $\underline{j} \neq \underline{i}$, então, para qualquer nó \underline{k} com qualquer informação, $\alpha_{\underline{i}, \underline{k}}(T)$ é uma pseudo-árvore.

- (iii) Apenas as estruturas definidas em (i) e (ii) são pseudo-árvores.

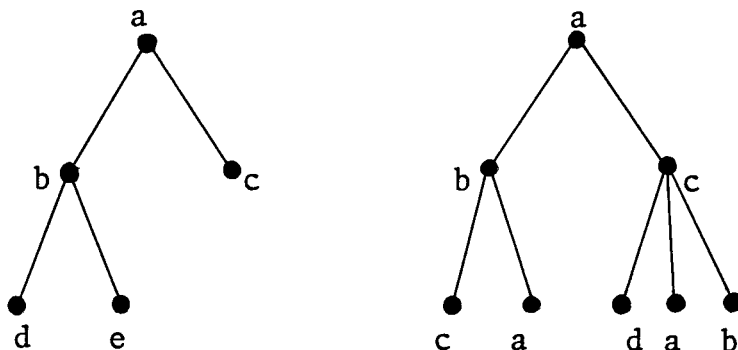


Figura X-3. Exemplos de pseudo-árvores.

X-3. Definição da Representação

A representação por pseudo-árvore de um digrafo D é uma seqüência ordenada de árvores T_1, \dots, T_k , que são obtidas através da execução, para cada vértice v_i de D , da expansão de v_i , que pode ser definida da seguinte maneira :

- (i) Se $i < j$, a expansão de v_i precede a expansão de v_j .
- (ii) Se o vértice v_i já foi expandido (como descendente de algum outro vértice), então não se faz nada.
- (iii) Se o vértice v_i ainda não foi expandido, seja k o número de árvores já construídas. Constrói-se uma árvore T_{k+1} composta somente do nó v_i . Após isso tomam-se todos os vértices v_j tais que $v_i R v_j$ e aplica-se a transformação $\alpha_{v_i, v_j}^{(T_{k+1})}$, de forma que se $j < j'$ então $\alpha_{v_i, v_j}^{(T_{k+1})}$ precede $\alpha_{v_i, v_{j'}}^{(T_{k+1})}$. Agora, para todos os vértices v_j definidos aqui, que ainda não tenham sido expandidos, repete-se este passo.

Seja k o número final de árvores obtidas. Após todos os vértices do digrafo terem sido expandidos, são executadas as operações T_i , $1 \leq i \leq k$, definidas nos passos seguintes:

- (i) Se $i > j$, a operação T_i precede a operação T_j .
- (ii) Seja p o maior inteiro tal que a informação \underline{r} da raiz de T_p ocorre em T_i . Substitua então T_i por $\beta_{\underline{r}}(T_p, T_i)$, onde \underline{r} é a ocorrência mais acima e à esquerda de \underline{r} em T_i . (Observe que necessariamente sempre teremos \underline{r} como terminal de T_i , de modo que a transformação $\beta_{\underline{r}}(T_p, T_i)$ pode ser aplicada.)

X-4. Exemplos

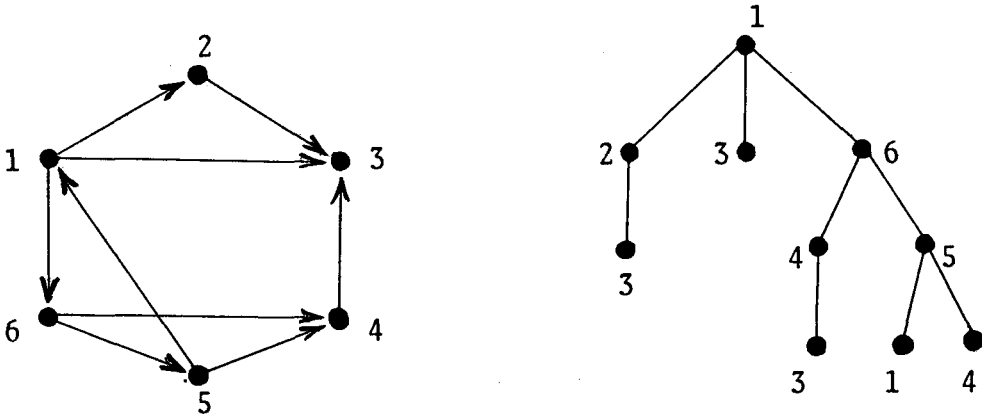


Figura X-4. Um digrafo e uma representação por pseudo-árvore.

No exemplo da figura X-4, não houve necessidade de executar os passos correspondentes às operações T_i pois a floresta resultante consistia já de uma única árvore.

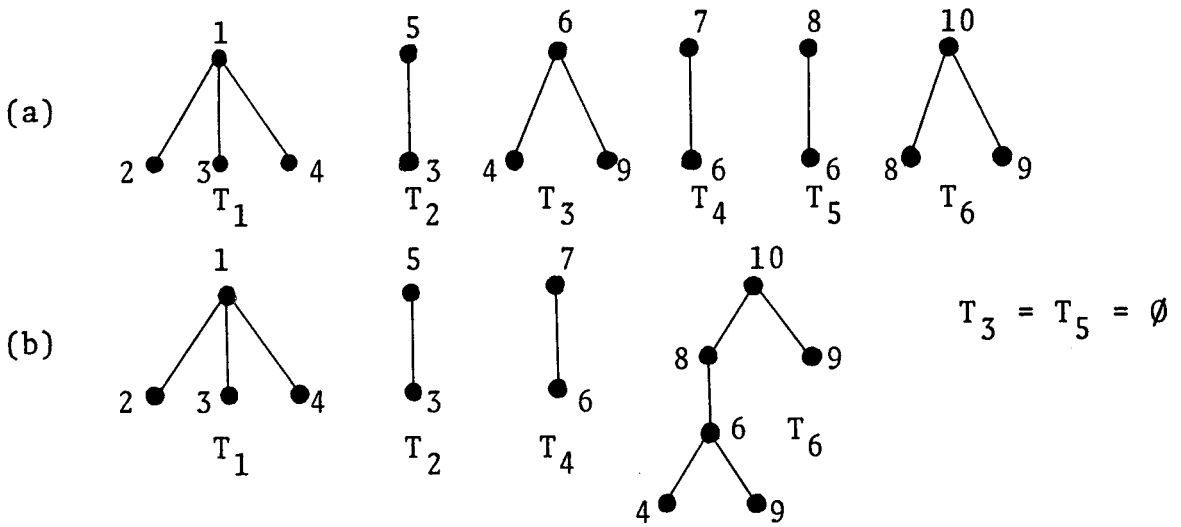
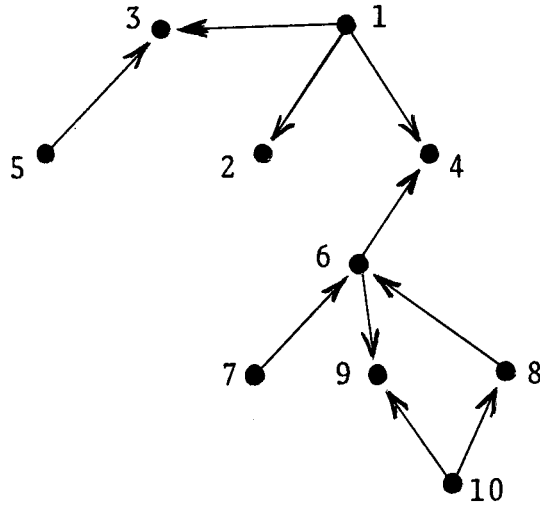


Figura X-5. Um digrafo e uma representação pseudo-árvore. (a) Árvores inicialmente obtidas após a expansão dos vértices. (b) Representação obtida após as operações T_i .

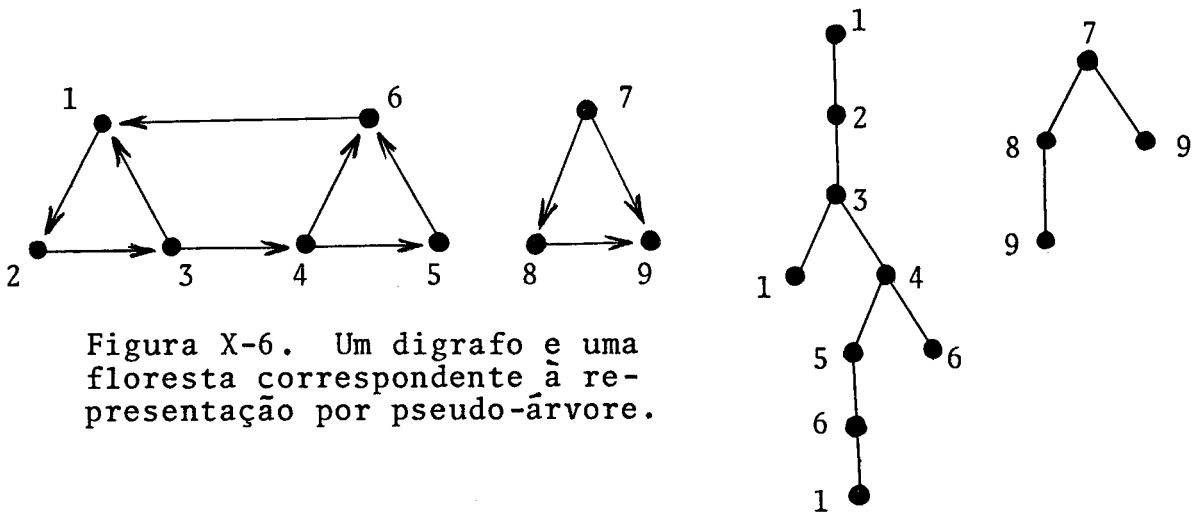


Figura X-6. Um digrafo e uma floresta correspondente à representação por pseudo-árvore.

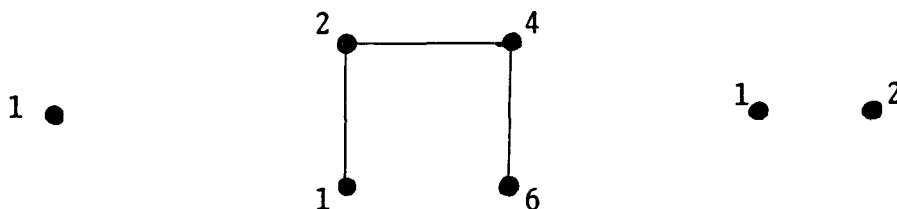
No exemplo da figura X-6, a representação consiste de 2 árvores. Não houve necessidade de executar as operações T_i .

X-5. Grafo da Pseudo-Árvore

Vamos definir agora o grafo da pseudo-árvore de um digrafo D , anotado $PTG(D)$.

O conjunto de vértices de $PTG(D)$ é o conjunto das árvores não vazias da representação por pseudo-árvore de D . O conjunto de arestas de $PTG(D)$ é formado pelos pares de árvores com elementos comuns, ou seja: para $v_i, v_j \in PTG(D)$ correspondendo a T_i, T_j da representação por pseudo-árvore,

$$v_i R v_j \iff T_i \text{ e } T_j \text{ têm elemento em comum.}$$



Grafos das pseudo-árvores correspondentes aos exemplos das figuras X-4, X-5 e X-6, respectivamente.

X-6. Propriedades Básicas e Comentários

Se D é um grafo direcionado fortemente conexo, então a sua representação por pseudo-árvore consiste de uma única árvore, qualquer que seja a rotulação dos vértices e a ordenação dos rótulos.

Dado um grafo não direcionado G , é sempre possível construir uma representação por pseudo-árvore de G , atribuindo direções quaisquer às suas arestas. Nesse caso, é interessante procurar a ordenação dos rótulos dos vértices que minimize o número de árvores na representação, conseqüentemente minimizando o número de nós.

O número total de arestas na representação é igual a M , onde M é o número de arestas do digrafo. O número total de nós é $M+B$, onde B é o número de pseudo-árvores na representação.

O grafo da pseudo-árvore fornece uma idéia sobre o número de componentes conexas do digrafo. Um digrafo D é desconexo se e somente se $PTG(D)$ é desconexo. Se atribuirmos direções e rótulos ordenados a $PTG(D)$, e obtivermos $PTG(PTG(D))$, veremos que este consiste de um conjunto de vértices isolados cuja cardinalidade é igual ao número de componentes conexas do digrafo D .

Observamos que a representação por pseudo-árvore é um conjunto T de árvores no qual dois ou mais nós de uma mesma árvore podem ter informações equivalentes. Se tivermos n nós com informações equivalentes, então ao menos $n-1$ deles serão nós terminais. Além disso, todos os sumidouros de um digrafo correspondem a nós terminais nas árvores.

Observamos também que, se eliminarmos o passo (ii) da definição da expansão de um vértice v_i , então todo digrafo contendo ciclos geraria árvores infinitas.

Se D for uma árvore, com seus vértices rotulados e ordenados adequadamente, então a representação por pseudo-árvore será uma árvore isomorfa a D . Esta propriedade é interessante no sentido de que permite a concepção de um sistema que trabalharia com algoritmos eficientes para árvores, caso a estrutura correspondesse a uma árvore, ou com algoritmos para digrafos, caso contrário — sem alterar a representação da estrutura de dados.

Observamos ainda que a própria definição da representação por pseudo-árvore, por ser uma definição construtiva, fornece um algoritmo para a obtenção da floresta em questão.

Para a representação da estrutura da pseudo-árvore em computador, a floresta é transformada em uma árvore binária. Para um nó A, o formato correspondente seria :

LTAG	LLINK	INFO	RLINK	RTAG
------	-------	------	-------	------

Figura X-8. Formato de um nó da representação por pseudo-árvore.

onde LTAG="+" se LLINK = primeiro filho de A,
 LTAG="-" se LLINK = nó principal da família A,
 RTAG="+" se RLINK = próximo irmão de A com respeito ao pai desta lista de irmãos,
 RTAG="-" se RLINK = raiz da próxima árvore na floresta.

O conjunto de nós com informações equivalentes à de A (informações repetidas) é dito a família A ; o nó principal desta família é o nó que carrega informação sobre os descendentes de A (isto é, o nó que foi expandido). De acordo com esta definição, o nó principal de uma família é único. Como exemplo, a representação em computador da pseudo-árvore da figura X-4 seria a mostrada na figura X-9.

Observamos que apenas vértices no primeiro nível de uma árvore podem ter RTAG="-". Caso RTAG="+" para todos os vértices exceto a raiz, a floresta consiste de uma única árvore. O digrafo é uma árvore enraizada se e somente se LTAG="+" para todos os nós.

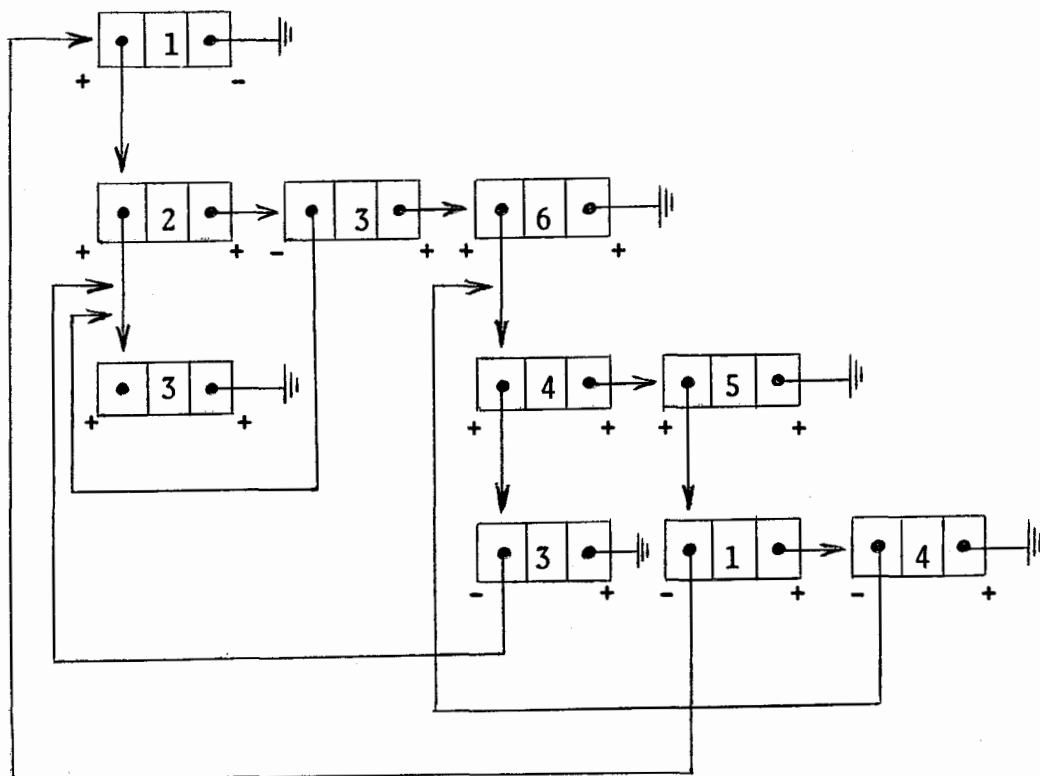


Figura X-9. Representação em computador da pseudo-árvore da figura X-4.

O número de nós na estrutura, como já foi dito, é $M+B$. Portanto, o espaço necessário em memória é proporcional a $M+B$ sendo que, no exemplo fornecido, é exatamente $3(M+B)$, incluindo o campo de informação. É ainda conveniente acrescentar um nó extra à estrutura fazendo o papel de raiz da pseudo-árvore. Esta conveniência torna-se mais clara quando a estrutura corresponde a uma floresta.

X-7. Referências

A representação por pseudo-árvore aparece em Szwarcfiter¹⁷.

CAPÍTULO XI

REPRESENTAÇÃO DE GRAFOS POR ÁRVORES BINÁRIASXI-1. Definição da Representação

Sabemos que podemos representar florestas de árvores por meio de árvores binárias cujos elos apontam para o primeiro filho e o próximo irmão de cada nó (Knuth⁹).

Verificaremos agora que uma classe mais geral de grafos direcionados também podem ser representados por árvores binárias.

Seja D um grafo direcionado, cujo conjunto de vértices é V e cujo conjunto de arestas é X . Seja v um vértice de D .

Chamemos de $R(v)$ a lista de adjacência de v , isto é,

$$R(v) = \{ u \mid (v,u) \in X \} .$$

Chamemos agora de D_R a estrutura obtida após a aplicação da transformação para árvore binária, que pode ser descrita da seguinte maneira :

Para cada vértice x do digrafo D , associe um nó c_x em D_R .

Considere agora todos os vértices w tais que existe uma aresta de w para x , isto é, tais que $R(w) = \{x,y,\dots,z\}$, onde y,\dots,z são vértices de D , e ligue os nós c_x, c_y, \dots, c_z como um conjunto. Ligue cada um dos nós c_w ao conjunto (lista) representando $R(w)$ (estes são os elos próximo irmão e primeiro filho de Knuth⁹).

Nos exemplos que se seguem, esta transformação foi mostrada por linhas pontilhadas para denotar elos de conjuntos

e por linhas cheias para denotar elos do tipo $w \rightarrow R(w)$. Por meio desses exemplos verificamos que em alguns casos a transformação para árvore binária pode ser aplicada com sucesso (figuras XI-1 e XI-2) e, em outros, não (figura XI-3).

XI-2. Exemplos

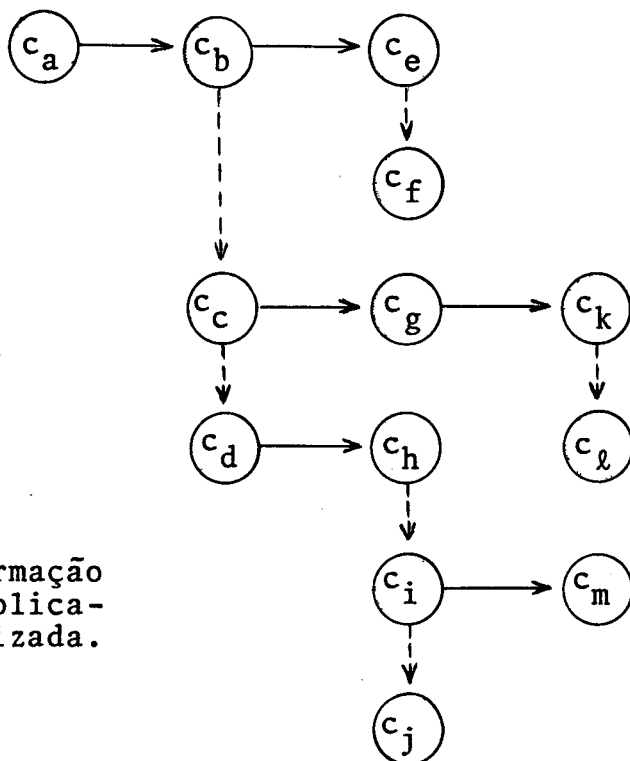
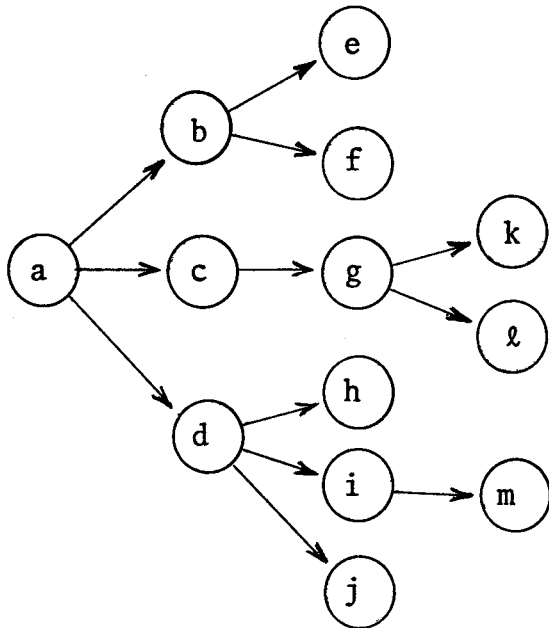


Figura XI-1. Transformação para árvore binária aplicada em uma árvore enraizada.

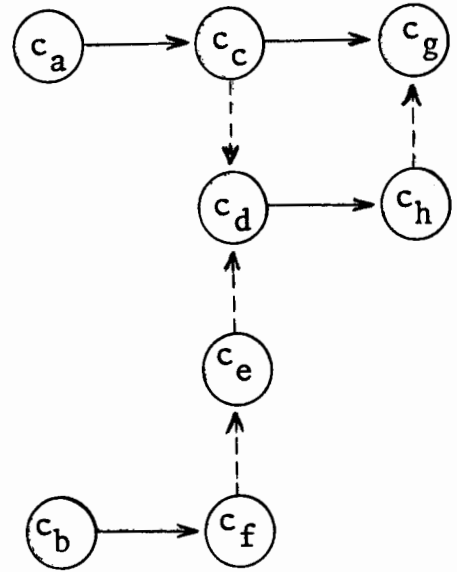
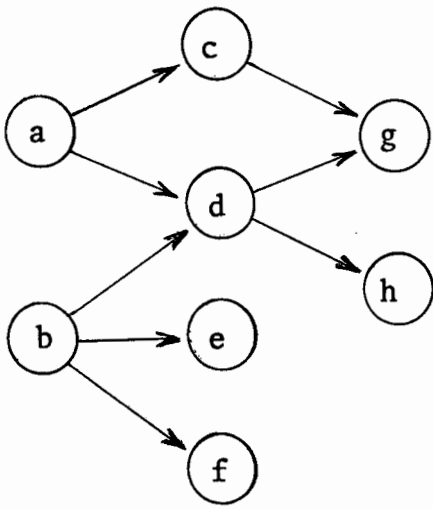


Figura XI-2. Transformação para árvore binária aplicada em um grafo acíclico.

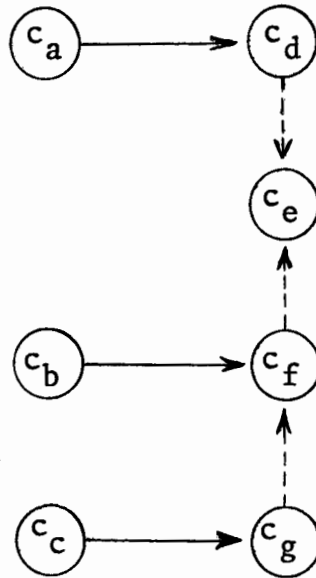
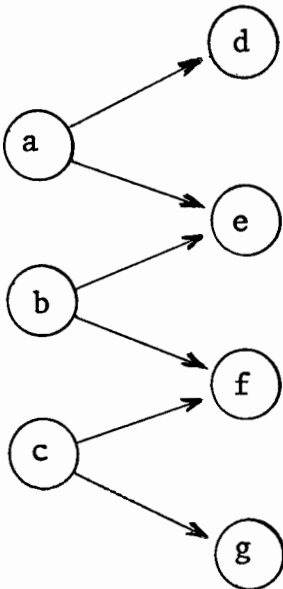


Figura IX-3. Aplicação inválida da transformação para árvore binária em uma árvore não enraizada.

XI-3. Propriedades Básicas e Comentários

Pudemos observar que nem todos os digrafos são representáveis por árvores binárias. Para caracterizar quais os digrafos que são representáveis por árvores binárias, vamos considerar o seguinte.

Seja $\mathcal{R} = \{ R(x) \mid x \in V \}$. Seja \mathcal{R}^* o conjunto obtido pela união de \mathcal{R} com todas as interseções finitas de elementos de \mathcal{R} . A relação " \subseteq " é uma relação de ordem parcial em \mathcal{R}^* . Temos assim o digrafo induzido $D^* = (\mathcal{R}^*, \subseteq)$.

Pfaltz^{12,13} provou o seguinte teorema: "O digrafo D é representável por árvore binária se e somente se $D^* = (\mathcal{R}^*, \subseteq)$ é uma árvore".

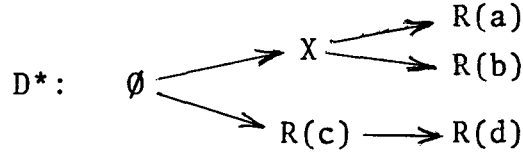
Observe que, como $\emptyset \subseteq R(x)$ para todo vértice x , \emptyset é a raiz de D^* .

A figura XI-4 contém exemplos dessa caracterização dos digrafos das figuras XI-2 e XI-3. Podemos observar que no primeiro caso D^* é uma árvore e o digrafo pode ser representado por árvore binária, e no segundo caso D^* não é uma árvore e o digrafo não pode ser representado por árvore binária.

Há, porém, um meio efetivo de contornarmos esta restrição e tornarmos todos os digrafos representáveis pela técnica da árvore binária: adicionando vértices auxiliares (extra), que fariam com que o novo \mathcal{R}^* fosse uma árvore. As rotinas que operam em D_R seriam então modificadas de forma a ignorar os nós auxiliares, como se não fizessem parte da estrutura.

$$\mathcal{R}^* = \{ R(a) = \{c, d\}, R(b) = \{d, e, f\}, R(c) = \{g\}, \\ R(d) = \{g, h\}, R(e) = R(f) = R(g) = R(h) = \emptyset, \\ X = R(a) \quad R(b) = \{d\} \}$$

(a)



$$\mathcal{R}^* = \{ R(a) = \{d, e\}, R(b) = \{e, f\}, R(c) = \{f, g\}, \\ R(d) = R(e) = R(f) = R(g) = \emptyset, X = R(a) \quad R(b) = \{e\} \\ Y = R(b) \quad R(c) = \{f\} \}$$

(b)

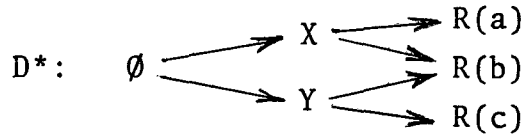


Figura XI-4. (a) Digrafo induzido pelo digrafo da figura XI-2. (b) Digrafo induzido pelo digrafo da figura XI-3.

O digrafo da figura XI-3 pode agora ser representado por uma árvore binária, com a adição do vértice x , conforme mostrado na figura XI-5.

Assim, graças à adição de vértices auxiliares e a um pequeno aumento da complexidade das rotinas, podemos representar qualquer digrafo com um número fixo de campos por nó.

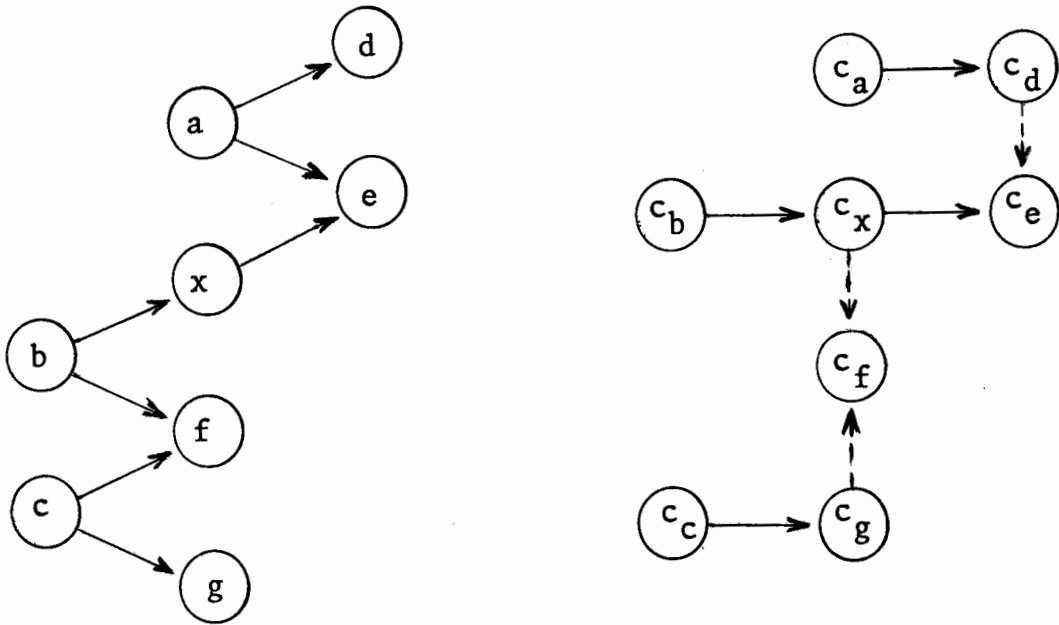


Figura XI-5. Modificação no digrafo da figura XI-3 de forma a permitir a aplicação válida da transformação para árvore binária.

Para a representação de grafos não direcionados, podemos empregar uma representação simétrica, sendo que neste caso seriam necessários 4 campos de elo por nó (em vez de 2).

XI-4. Referências

Outros detalhes e aspectos teóricos desta representação podem ser encontrados em Pfaltz^{12,13}, que introduziu esta representação.

CAPÍTULO XII

ÁRVORE-COÁRVOREXII-1. Definição da Representação

Dado um grafo G conexo, G pode ser determinado por uma árvore geradora qualquer e a correspondente coárvore.

Podemos assim definir uma representação para um grafo conexo G , que consiste de uma árvore geradora de G e a correspondente coárvore. Inicialmente, obtemos uma árvore geradora (enraizada), e, para cada nó da árvore, associamos a lista dos vértices que são adjacentes ao vértice correspondente a esse nó mas que não fazem parte da árvore (dita a lista associada). A união de todas essas listas é a coárvore.

A representação da árvore em computador é feita através da sua árvore binária correspondente, tendo então cada nó da árvore o seguinte formato :

LLINK	INFO	ALINK	RLINK
-------	------	-------	-------

Figura XII-1. Formato de um nó da representação árvore-coárvore.

Para um nó pertencente à árvore, LLINK e RLINK são os apontadores esquerdo e direito usuais da árvore binária, ALINK aponta para a lista associada a esse nó e o campo INFO indica o vértice correspondente do grafo.

Para um nó pertencente a uma das listas associadas, poderíamos ter RLINK apontando para o próximo nó da lista, ALINK apontando para o nó da árvore que corresponde ao nó considerado, e LLINK contendo um código especial para indicar que este nó pertence a uma lista associada e não à árvore geradora.

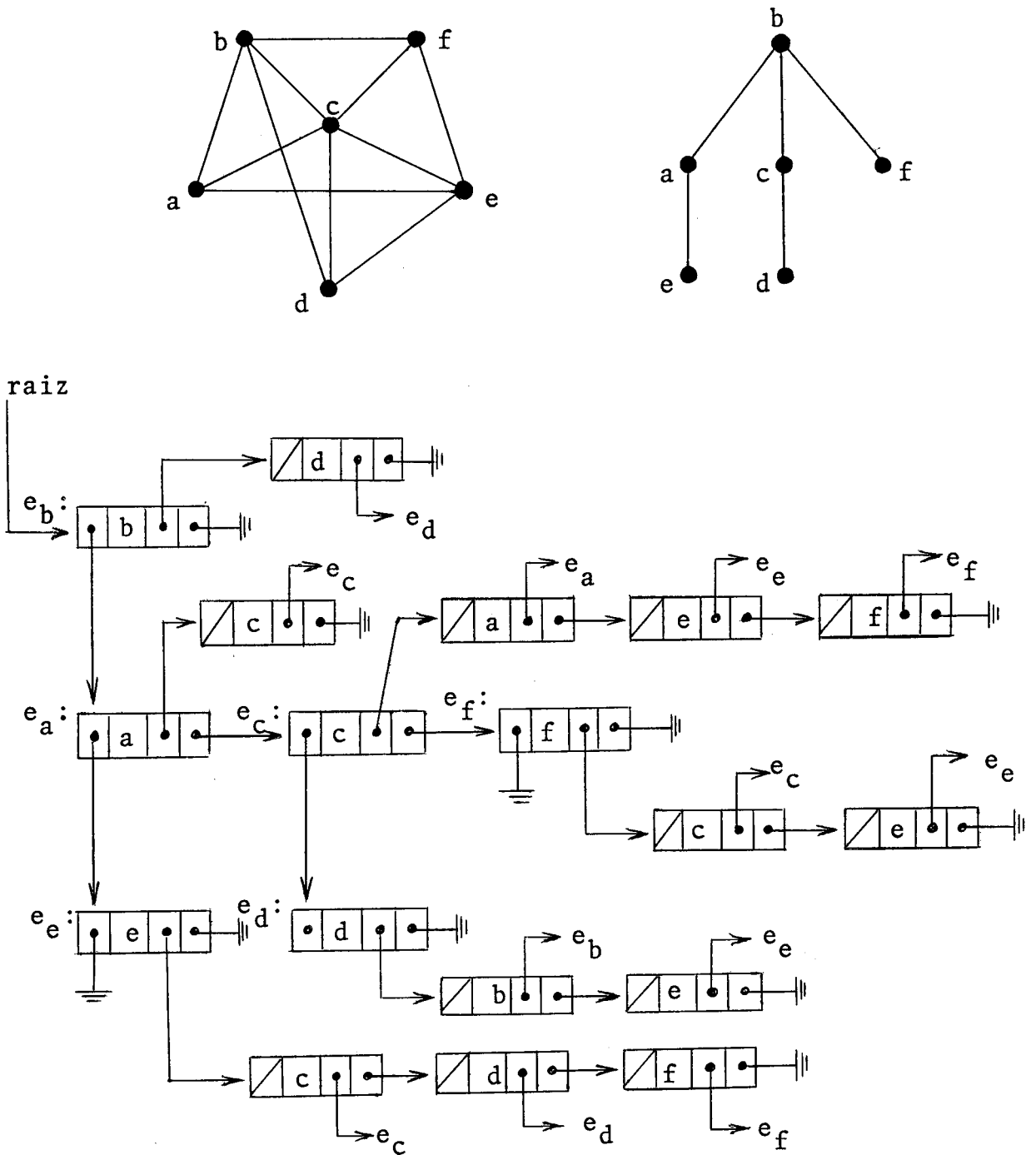
XII-2. Exemplo

Figura XII-2. Um grafo, uma árvore geradora e a correspondente representação árvore-co-árvore.

Neste exemplo, as arestas do grafo não pertencentes à árvore geradora são

$(b,d), (c,a), (c,f), (c,e), (d,e), (e,f)$.

XII-3. Propriedades Básicas e Comentários

Através desta representação é fácil reconhecer se um determinado grafo é ou não uma árvore ; o grafo será uma árvore se e somente se todos os ALINK apontarem para listas vazias.

A árvore geradora de um grafo não é única. Não é imediato determinar qual árvore geradora é a mais apropriada para a representação do grafo.

Se o grafo for desconexo, é possível obter uma árvore geradora para cada componente conexa do grafo. Como a floresta geradora assim obtida pode ser transformada em uma árvore binária, esta mesma estrutura de representação pode ser usada para representar grafos desconexos.

Com esta mesma definição podemos representar pseudografos e/ou multigrafos. A figura XII-3 é um exemplo.

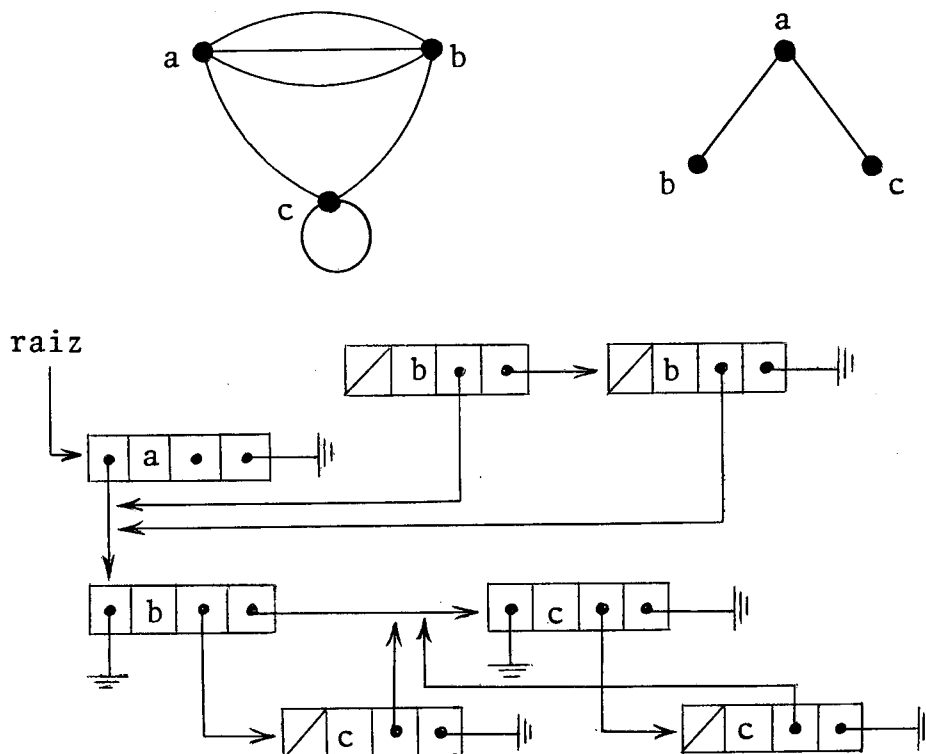


Figura XII-3. Um pseudomultigrafo, uma árvore geradora e a correspondente representação árvore-coárvore.

É vantajoso usar aqui a representação alinhavada de árvores binárias, pois assim podemos encontrar facilmente todos os vértices adjacentes a um determinado vértice, o que constitui um dos problemas básicos da manipulação de grafos. Observe que se tivermos um trecho da estrutura semelhante ao desenho da figura XII-4, os vértices do grafo adjacentes ao vértice X serão $Z_1, Z_2, \dots, Z_n, W_1, W_2, \dots, W_m, Y$:

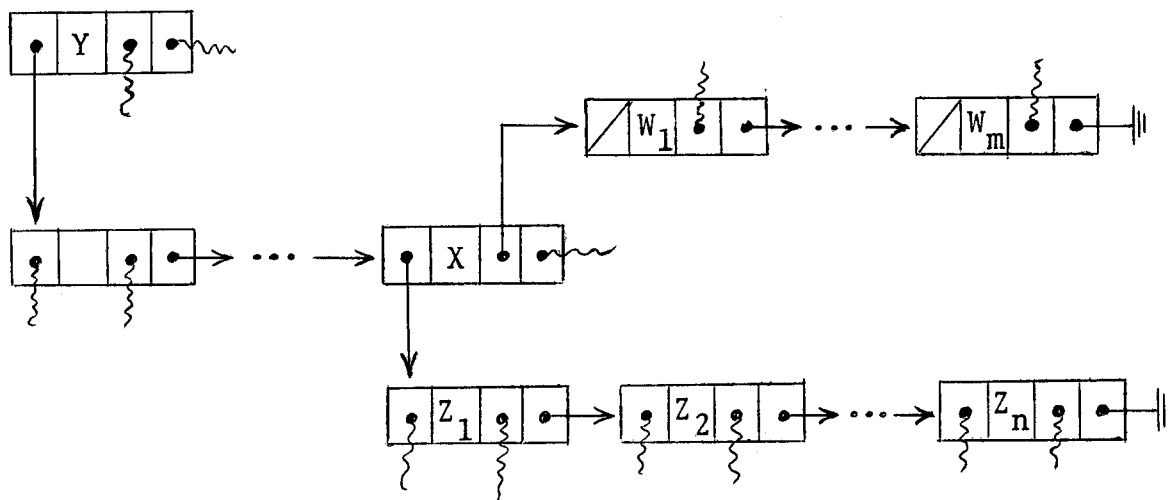


Figura XII-4. Trecho de uma representação árvore-co-árvore.

Uma outra maneira de se obter um resultado equivalente seria usando o campo RLINK do último nó de cada lista associada para armazenar um apontador para o nó da árvore correspondente ao pai do nó ao qual pertence a lista associada (com sinal negativo, por exemplo). Desta forma, se o grafo for uma árvore, a sua representação árvore-co-árvore corresponderá à representação usual de árvores com 3 apontadores (primeiro filho, próximo irmão e pai).

A inserção de novas arestas na estrutura não constitui problema, pois apenas as listas associadas são afetadas. Contudo, a remoção de uma aresta não é trivial, pois, se a aresta

ta fizer parte da árvore geradora, esta operação partirá a árvore em duas, sem necessariamente partir o grafo em dois. Logo, esta representação não se adapta a esse tipo de operação. A remoção de um vértice, que pode ser encarada como a remoção de todas as arestas incidentes a esse vértice, apresenta os mesmos problemas. A inserção de novos vértices não apresenta problemas.

Para um grafo com N vértices e M arestas, observamos que a árvore geradora tem N nós e as listas associadas $2(M-N+1)$, pois a árvore pode representar $N-1$ arestas e as restantes $M-(N-1)$ são representadas duas vezes cada uma. Isto nos dá um total de $N + 2(M-N+1) = 2M-N+2$ nós. Como cada nó possui 4 campos, temos um total de $4(2M-N+2)$ informações a serem armazenadas.

O uso de uma árvore geradora de um grafo como ferramenta para obter uma representação desse grafo também nos sugere um outro tipo de representação, diferente da árvore-coárvore, que seria o seguinte: a nova representação consistiria de um conjunto ordenado de árvores rotuladas, tal que a primeira árvore é uma árvore geradora do grafo, a segunda é uma árvore geradora da coárvore, a terceira é uma árvore geradora da coárvore da coárvore, e assim por diante.

XII-4. Referências

A idéia de uma possível representação de um grafo por meio de sua árvore geradora e de uma tabela adicional representando a coárvore foi sugerida por Scoins¹⁴. A representação árvore-coárvore pode ser encontrada em Szwarcfiter¹⁷.

CAPÍTULO XIII

REPRESENTAÇÃO POR ESTRUTURA DE LISTAXIII-1. Definição da Representação

Talvez a representação mais natural para digrafos , por estruturas ligadas, seja a representação por estrutura de lista.

Há várias maneiras de se representar um digrafo através de estruturas de listas ; a que apresentamos aqui é apenas uma das possibilidades.

Cada nó da estrutura apresenta o seguinte formato:

TAG	CAMPO-A	CAMPO-B
-----	---------	---------

Figura XIII-1. Formato de um nó da representação por estrutura de lista.

onde TAG = "+", se o CAMPO-A corresponde a um vértice do digrafo,

TAG = "-", se o CAMPO-A é um apontador para o primeiro nó da sublista à qual pertence este nó,

CAMPO-B = apontador para o próximo nó da sublista à qual pertence este nó.

A sublista correspondente a cada vértice do digrafo contém os vértices adjacentes a ele.

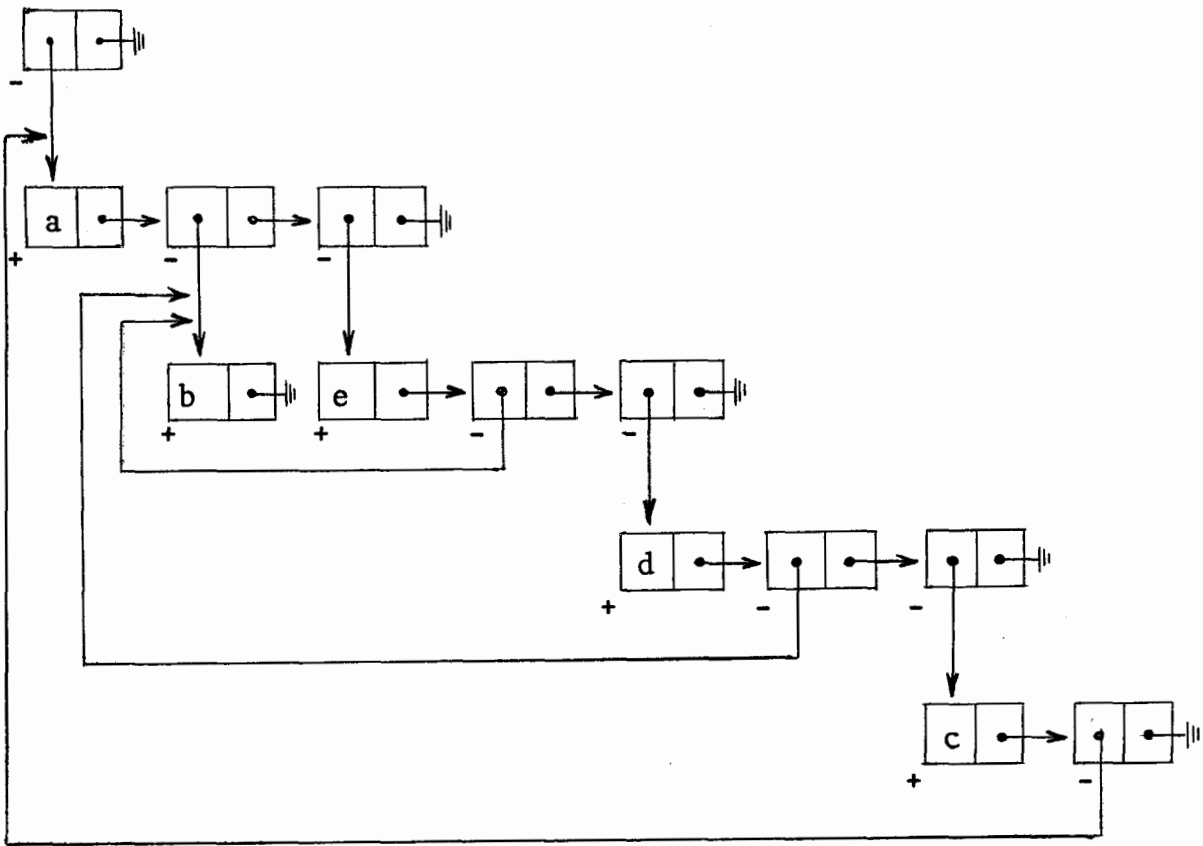
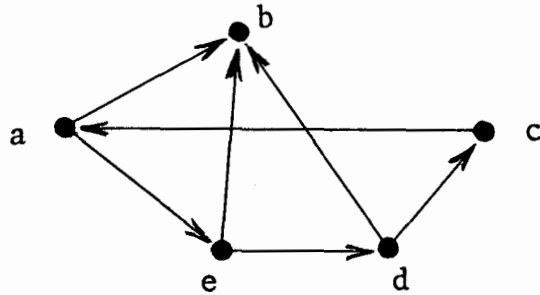
XIII-2. Exemplo

Figura XIII-2. Um digrafo e uma representação por estrutura de lista.

XIII-3. Propriedades Básicas e Comentários

O campo de informação da estrutura, que corresponde ao CAMPO-A com TAG="+", pode ser armazenado em uma lista à parte. Nesse caso, o CAMPO-A conteria um apontador para essa lista, ao invés de conter a própria informação. Isso é interessante no caso das informações serem compostas por dados longos ou de comprimento variável.

A adição de novos vértices à estrutura não representa grande problema. A inserção ou remoção de uma aresta (v_i, v_j) , dados os endereços dos vértices v_i e v_j , também pode ser feita facilmente. No caso de remoção da aresta, elimina-se, pelo reajuste de apontadores, o nó que pertence à sublista de v_i e que aponta para v_j , sem contudo eliminar a sublista de v_j pois pode haver outras arestas terminando em v_j . De um modo geral, podem ser usadas técnicas de "garbage collection" em caso de remoção de arestas.

Esta representação por estrutura de lista possui trechos circulares apenas se o digrafo contém ciclos. Uma mesma sublista pode ser compartilhada por mais de uma sublista. Caso isso não fosse possível, poderíamos ter listas infinitas.

Uma observação importante é que a lista só contém os vértices atingíveis pelo primeiro vértice da lista (que, no caso do exemplo apresentado na figura XIII-2, é o vértice a). Se o vértice inicial escolhido não atingir todos os vértices do digrafo, representa-se a parte restante do digrafo por uma outra estrutura de lista, e assim por diante, até o digrafo ser totalmente representado; as estruturas assim obtidas são então ligadas pelo CAMPO-B de suas cabeças de lista. Podemos observar que haverá no máximo tantas estruturas quantas forem as componentes fortemente conexas do digrafo.

Esta mesma definição pode ser aplicada a pseudografos e/ou multigrafos direcionados, como mostra a figura XIII-3 a seguir.

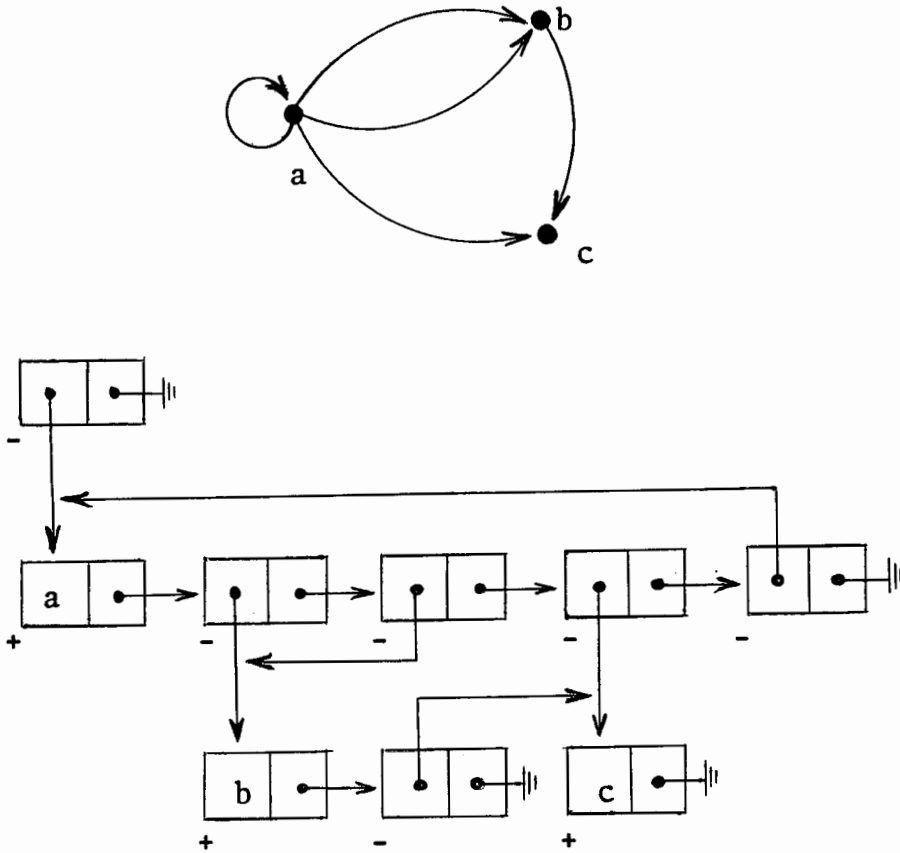


Figura XIII-3. Um pseudomultigrafo direcionado e uma representação por estrutura de lista.

Podemos ver que na estrutura há um nó para cada vértice e para cada aresta, e mais um nó para cada estrutura de lista obtida. Nesta representação particular, cada nó é composto por 2 campos, e portanto o espaço necessário para armazenar um digrafo com N vértices, M arestas e L estruturas de lista é $2(M+N+L)$.

XIII-4. Referências

Como referência para este tipo de representação, veja Berztiss³, Knuth⁹ e Szwarcfiter¹⁷.

CAPÍTULO XIV

LISTA DE RAMOSXIV-1. Definição da Representação

A representação lista de ramos é uma forma seqüencial compacta de representar grafos.

Considere um grafo G , com seus N vértices rotulados com inteiros consecutivos $1, \dots, N$. Considere agora todos os pares de vértices relacionados, representando uma relação entre dois vértices i e j por (i, j) , sendo que, para cada par (i, j) de vértices relacionados, considere apenas uma das representações (i, j) ou (j, i) . Dessa maneira, obtêm-se exatamente M pares de vértices, onde M é o número de arestas do grafo.

Para cada vértice v_i , considere a seqüência

$$Ad(i) = j, k, \dots, 1,$$

onde $(i, j), (i, k), \dots, (i, 1)$ são todos os pares começando por i .

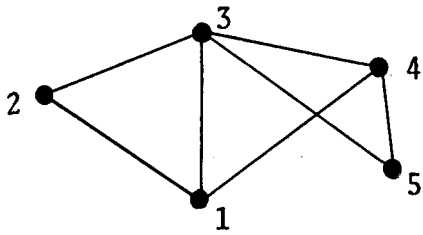
A lista de ramos é definida pela reunião de todas essas seqüências $Ad(1), \dots, Ad(N)$, nesta ordem.

A lista de entradas é definida pela seguinte seqüência de inteiros:

$$1, 1 + |Ad(1)|, 1 + |Ad(1)| + |Ad(2)|, \dots, 1 + \sum_{j=1}^{N-1} |Ad(j)|,$$

onde $|Ad(j)|$ denota o número de elementos da seqüência $Ad(j)$.

A representação por lista de ramos compõe-se da lista de ramos e da lista de entradas.

XIV-2. Exemplo

3,3,1,4,1,5,3 (lista de ramos)

1,2,4,5,7 (lista de entradas)

Figura XIV-1. Um grafo e uma representação por lista de ramos.

Na figura XIV-1 acima, os pares de vértices considerados e as correspondentes sequências $Ad(i)$ são :

(1,3), (2,3), (2,1), (3,4), (4,1), (4,5), (5,3)

$Ad(1) = 3$	$ Ad(1) = 1$
$Ad(2) = 3,1$	$ Ad(2) = 2$
$Ad(3) = 4$	$ Ad(3) = 1$
$Ad(4) = 1,5$	$ Ad(4) = 2$
$Ad(5) = 3$	$ Ad(5) = 1$

XIV-3. Propriedades Básicas e Comentários

A partir da definição da representação, pode-se ver que, dado um grafo, para cada rotulação dos vértices e para cada maneira de se representar as arestas, tem-se uma única representação por lista de ramos. Por outro lado, as duas listas da representação obtida determinam unicamente o grafo que gerou a representação.

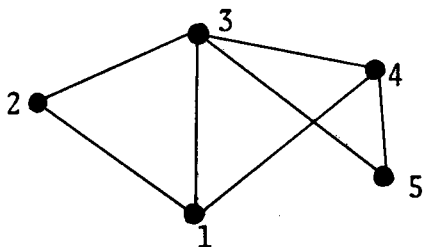
Chamemos R , a lista de ramos, e
 E , a lista de entradas ,

e denotemos o k -ésimo elemento dessas listas por $R(k)$ e $E(k)$, respectivamente.

Para encontrar todos os vértices adjacentes a um determinado vértice v_i , precisamos inicialmente tomar todos os e-

lementos de R começando por $R(E(i))$ e considerando $E(i+1) - E(i)$ elementos. Se $i = N$ (onde N é o número de vértices do grafo), consideram-se os elementos de R começando por $R(E(i))$ e indo até o final da lista. (Ou, para evitar essa mudança de procedimento no caso $i = N$, pode-se inserir um $(N+1)$ -ésimo elemento em E , com valor $M+1$.) Este procedimento só encontra os vértices v_j tais que o par (i,j) aparece na representação escolhida para as arestas. Para encontrar também todos os vértices v_j tais que o par (j,i) aparece na representação escolhida para as arestas, tem-se que procurar pelo elemento i dentre todos os vértices adjacentes a v_j , da maneira descrita acima.

Entretanto, para se obter um resultado melhor em termos de rapidez, pode-se fixar uma regra para a formação das seqüências $Ad(i)$, estabelecendo, por exemplo, que a seqüência seja composta por todos os j tais que o vértice v_i está relacionado ao vértice v_j e tais que $i < j$. Neste caso, sempre teremos $|Ad(N)| = 0$.



Representação:

2,3,4,3,4,5,5

1,4,5,7,8

Figura XIV-2. Um grafo e uma representação por lista de ramos.

Com relação à figura XIV-2 acima, as seqüências consideradas na obtenção da representação foram :

$$Ad(1) = 2,3,4$$

$$Ad(2) = 3$$

$$Ad(3) = 4,5$$

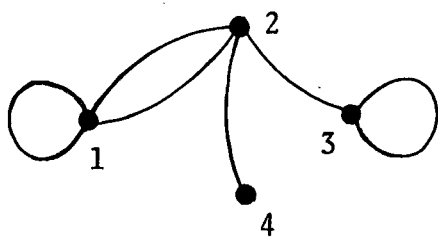
$$Ad(4) = 5$$

$$Ad(5) = \text{seqüência vazia}$$

Agora, para encontrar os vértices adjacentes a um dado vértice v_i , precisamos procurar em todas as sublistas começando por $R(E(j))$ para todos os j tais que $j \leq i$ - e não mais para todos os j , como antes.

A lista de ramos compõe-se de um inteiro para cada aresta, e a lista de entradas de um inteiro para cada vértice. Portanto a representação por lista de ramos necessita de $M + N$ posições de armazenamento.

Com esta mesma definição, sem qualquer mudança, podemos representar pseudografos e/ou multigrafos, conforme pode ser visto na figura XIV-3 abaixo.

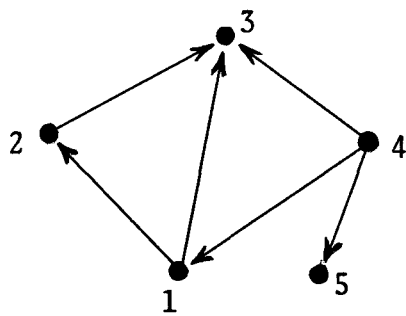


1,2,2,3,4,3 (lista de ramos)

1,4,6,7 (lista de entradas)

Figura XIV-3. Um pseudomultigrafo e uma representação por lista de ramos.

Ao definirmos a representação por lista de ramos para um grafo não direcionado, representamos uma relação entre dois vértices v_i, v_j por apenas um par (i, j) - isto é, na realidade associamos o grafo não direcionado a um grafo direcionado. Concluimos daí que esta representação é mais natural para grafos direcionados do que para grafos não direcionados. Esta representação é também mais eficiente para digrafos, no sentido de que podem-se obter todos os vértices adjacentes a um vértice v_i simplesmente consultando a sublista começando por $R(E(i))$ e considerando $E(i+1) - E(i)$ elementos (ou terminando no final da lista). A figura XIV-4 a seguir é um exemplo deste fato.



2,3,3,1,3,5 (lista de ramos)

1,3,4,4,7 (lista de entradas)

Figura XIV-4. Um digrafo e uma representação por lista de ramos.

Observamos que foi aplicada exatamente a mesma definição utilizada para grafos não direcionados.

Note que $Ad(i)$ contém todos os elementos j tais que $v_i R v_j$. Assim, se v_i é sumidouro então $Ad(i)$ é vazia ; e se v_i é vértice fonte então i não aparece em nenhum lugar na lista de ramos. Portanto, um vértice v_i é isolada se e somente se $Ad(i)$ é vazia e i não consta da lista de ramos. As seqüências $Ad(i)$ correspondentes à figura XIV-4 são as seguintes :

$$Ad(1) = 2,3$$

$$Ad(2) = 3$$

$$Ad(3) = \text{seqüência vazia}$$

$$Ad(4) = 1,3,5$$

$$Ad(5) = \text{seqüência vazia}$$

Este tipo de representação pode ser classificada como seqüencial. De um modo geral, as representações seqüenciais se caracterizam por requerer pouco armazenamento, mas também por não permitir na prática modificações na estrutura (inserção ou remoção de vértices ou arestas). É claro que este comentário é válido também para qualquer outra representação seqüencial presente neste trabalho.

XIV-4. Referências

A representação por lista de ramos pode ser encontrada em Obruca¹⁰, que a definiu apenas para digrafos, e em Szwarcfiter¹⁷.

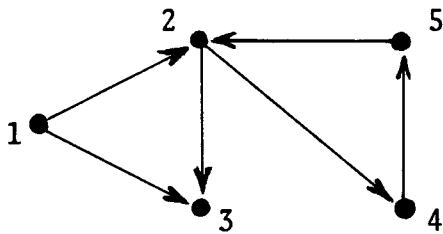
CAPÍTULO XV

SEQUÊNCIA DE ADJACÊNCIAXV-1. Definição da Representação

Seja D um digrafo, tendo seus N vértices rotulados com inteiros $1, \dots, N$.

Para cada vértice v_i do digrafo, considere a seqüência de todos os rótulos j dos vértices v_j tais que $v_i R v_j$. Se k é o primeiro rótulo que aparece nessa seqüência, represente tal rótulo por $-k$. Represente uma seqüência vazia por -0 .

A concatenação das seqüências assim obtidas, em ordem dos rótulos dos vértices que as geraram, é chamada a seqüência de adjacência e constitui uma representação para digrafos.

XV-2. Exemplo

Seqüência de adjacência :

$-2, 3, -3, 4, -0, -5, -2$

Figura XV-1. Um digrafo e uma representação por seqüência de adjacência.

XV-3. Propriedades Básicas e Comentários

Para encontrar os vértices adjacentes a um dado vértice v_i , temos que encontrar o i -ésimo inteiro negativo da seqüência e, a partir desse elemento (inclusive), considerar todos

os rótulos até o próximo inteiro negativo (exclusive) ou até o final da seqüência. Se o i -ésimo inteiro negativo for um -0 , então não existe no digrafo nenhum vértice v_j tal que $v_i R v_j$.

Esta representação necessita de $M+S$ posições de armazenamento, onde M é o número de arestas do digrafo e S é o número de sumidouros.

Devido ao fato desta representação ser ruim para manipulações, mas bastante boa em termos de gasto de memória, ela pode ser usada para armazenar o digrafo em uma memória auxiliar durante o tempo em que não está sendo manipulado.

XV-4. Referências

A representação por seqüência de adjacência foi definida por Szwarcfiter¹⁷.

CAPÍTULO XVI

REPRESENTAÇÃO POR K-FÓRMULAXVI-1. Definição da Representação

A representação por k-fórmula é uma representação sequencial para grafos direcionados, cujo objetivo é conseguir um conjunto de uma ou mais fórmulas lineares que identifiquem univocamente o digrafo.

Seja D um digrafo cujos vértices estão rotulados. A construção das fórmulas para a representação de D baseia-se no operador "*", dito k-operador, para representar as arestas do digrafo. Assim, $*ab$ representa a aresta (a,b) , onde a,b são vértices. Quando há mais de uma aresta originando-se em um mesmo vértice, empregam-se tantos k-operadores quantas forem as arestas. No digrafo da figura XVI-1, por exemplo, as fórmulas que representam as arestas que se originam nos vértices \underline{a} , \underline{b} e \underline{c} são, respectivamente,

$**abc$, $**bac$, $*cd$.

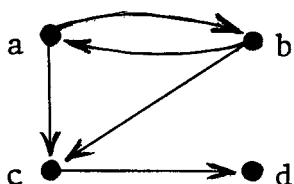


Figura XVI-1. Um digrafo.

Substituindo, na fórmula do vértice \underline{a} , os rótulos \underline{b} e \underline{c} pelas fórmulas dos vértices \underline{b} e \underline{c} , respectivamente, obtém-se

$**a**bac*cd$.

Esta última fórmula é uma k-fórmula, e representa o digrafo da figura XVI-1.

A construção da representação por k -fórmula, que foi apresentada intuitivamente, pode ser formalizada da maneira apresentada a seguir.

Uma k -fórmula pode ser definida recursivamente por :

- (i) Um rótulo de um vértice é uma k -fórmula.
- (ii) Se α e β são k -fórmulas, então $*\alpha\beta$ é uma k -fórmula.

O vértice inicial de uma k -fórmula é o vértice cujo rótulo aparece mais à esquerda na k -fórmula, a qual é dita então uma k -fórmula desse vértice.

O algoritmo para a obtenção das k -fórmulas que representam um digrafo é o seguinte :

- (1) Para todo vértice isolado \underline{a} do digrafo, construa a k -fórmula \underline{a} .
- (2) Para toda aresta (a,b) do digrafo construa a k -fórmula $*ab$.
- (3) Combine as k -fórmulas obtidas de acordo com a seguinte regra de substituição : se existir uma k -fórmula de um vértice e se existir também uma outra k -fórmula que contenha um rótulo desse vértice, substitua tal rótulo pela k -fórmula do vértice. Aplique esta regra de substituição até que ela não possa mais ser aplicada, cuidando sempre para que a k -fórmula resultante de cada substituição seja a mais longa possível.
- (4) Sejam k_1, k_2, \dots, k_n as k -fórmulas obtidas após a execução do passo (3), e sejam a_1, \dots, a_n os vértices iniciais de k_1, \dots, k_n , respectivamente. Se alguma k_i contiver, como subfórmula, uma k -fórmula de um vértice \underline{b} onde a_i aparece, e se o rótulo \underline{b} ocorrer em uma das demais k -fórmulas $k_1, \dots, k_{i-1}, k_{i+1}, \dots, k_n$, extraia a k -fórmula de \underline{b} da k -fórmula k_i , inserindo \underline{b} em seu \underline{lu}

gar, e substituindo a k -fórmula de \underline{b} na outra fórmula. Retorne então ao passo (3).

XVI-2. Exemplo



Figura XVI-2. Um digrafo e uma representação por k -fórmula.

No exemplo da figura XVI-2, o passo (1) não se aplica pois não há vértices isolados.

Após a execução do passo (2), obtemos

$\{ *ab , *bc , *cb \} .$

Após a execução do passo (3), obtemos

$\{ *ab , *c*bc \} .$

Na execução do passo (4), verificamos que, na fórmula $*c*bc$, a subfórmula $*bc$ contém \underline{c} e que a fórmula $*ab$ contém \underline{b} . Portanto, extraímos a subfórmula $*bc$, colocando \underline{b} em seu lugar (a fórmula $*c*bc$ fica agora $*cb$), e substituímos agora $*cb$ em $*bc$, obtendo $*b*cb$.

Retornando ao passo (3), o conjunto de k -fórmulas é transformado em

$\{ *a*b*cb \} ,$

que é a representação final do digrafo.

XVI-3. Propriedades Básicas e Comentários

O número total de k -operadores que precedem todas as ocorrências de um determinado rótulo \underline{a} é igual ao número de arestas que se originam no vértice \underline{a} .

Se o rótulo \underline{a} é precedido por \underline{n} k -operadores, então \underline{a} é o vértice inicial de uma k -fórmula

$$\underbrace{** \dots *}_n a \alpha_1 \alpha_2 \dots \alpha_n ,$$

onde $\alpha_1, \dots, \alpha_n$ são k -fórmulas. Os vértices terminais das arestas que se originam em \underline{a} são os vértices iniciais dessas k -fórmulas $\alpha_1, \dots, \alpha_n$.

A representação de pseudografos e/ou multigrafos direcionados pode ser feita usando-se esta mesma definição, conforme pode ser visto no exemplo da figura XVI-3.

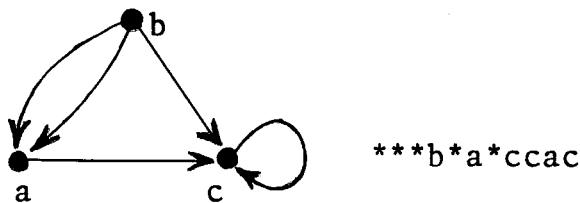


Figura XVI-3. Um digrafo e uma representação por k -fórmula.

Seja M o número de arestas do digrafo. Então o número total de k -operadores na representação é M , e o número total de rótulos de vértices é igual a $M + K$, onde K é o número final de k -fórmulas na representação do digrafo. Portanto, a representação por k -fórmula desse digrafo requer espaço proporcional a $2M + K$.

O vértice inicial de uma k -fórmula atinge todos os outros vértices que aparecem nessa k -fórmula. Podemos assim concluir que haverá no máximo tantas k -fórmulas quantas forem

as componentes fortemente conexas do digrafo.

XVI-4. Referências

Outros detalhes desta representação podem ser encontrados em Berztiss³. A representação por k-fórmula aparece ainda em Szwarcfiter¹⁷.

CAPÍTULO XVII

REPRESENTAÇÃO POR STRINGS

A representação de grafos por strings, sob a forma de fórmulas lineares, constitui uma abordagem bastante geral para a representação de grafos e digrafos. Tais strings são de manipulação relativamente fácil por não apresentarem parênteses ou outros operadores hierárquicos.

XVII-1. Definição da Representação

Seja D um digrafo cujos vértices estão rotulados . Seja v_i um vértice de D . Se v_i tem grau de saída m , chamaremos de fórmula de saída de v_i ao conjunto das m arestas que partem de v_i .

Para a representação de uma fórmula de saída, usaremos a seguinte notação prefixa :

$$(m, v_i) v_{j_1}, v_{j_2}, \dots, v_{j_m} ,$$

onde $(v_i, v_{j_1}), (v_i, v_{j_2}), \dots, (v_i, v_{j_m})$ são as m arestas que partem de v_i .

O simétrico da fórmula de saída de v_i é a sua fórmula de entrada :

$$[p, v_i] v_{k_1}, v_{k_2}, \dots, v_{k_m} .$$

Cada fórmula é composta de um operador (que consiste do rótulo do vértice e de um numeral que representa o grau apropriado do vértice) e dos vértices relacionados ao vértice considerado ; vírgulas, parênteses e colchetes são meros delimitadores.

A construção das fórmulas de entrada e de saída dos vértices de um digrafo é elementar, qualquer que seja a forma na qual o digrafo é fornecido. É óbvio também que as fórmulas de entrada ou de saída dos vértices de um digrafo o representam de forma unívoca e que qualquer outra representação pode ser derivada dessas fórmulas.

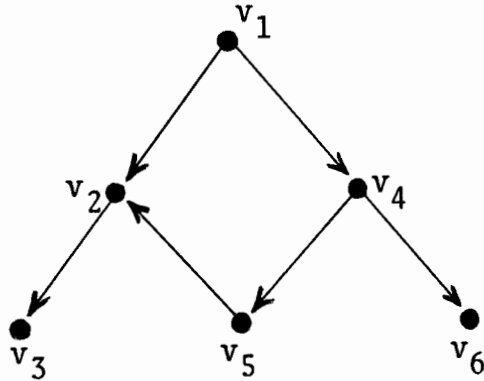
É interessante notar que as fórmulas de entrada e de saída de um vértice são uma representação abreviada respectivamente de uma coluna e linha da matriz de adjacência.

Pode-se, entretanto, encontrar uma representação mais compacta para um digrafo. De fato, dada uma fórmula de saída de um vértice v_i surge a idéia de substituir um rótulo v_j contido na fórmula de v_i pela fórmula de v_j , e assim por diante, até que as várias fórmulas dos vértices do digrafo tenham sido combinadas em uma (ou poucas), representando o digrafo todo. Tal substituição pode ser feita de maneira aleatória ou de maneira ordenada. Para fazer substituições de maneira ordenada, duas maneiras possíveis seriam :

- 1- seguindo um caminho orientado ao qual pertence um determinado vértice (substituição descendente) ;
- 2- seguindo o caminho na direção oposta (substituição ascendente).

Tais substituições podem ser feitas em fórmulas de entrada ou de saída, e podem ou não ser condicionadas a certas regras. Além disso, pode ser usada a notação posfixa ao invés da prefixa.

Por motivos de simplicidade de notação, não se deve misturar notação prefixa e posfixa, ao passo que, por motivos de simplicidade de algoritmos, deve-se limitar a mistura de substituição ascendente com descendente ao mínimo possível.

XVII-2. Exemplo

Fórmulas de entrada :

$$\begin{array}{lll}
 [0, v_1] & [2, v_2] & v_1, v_5 \\
 [1, v_4] & v_1 & [1, v_5] & v_4 \\
 & & [1, v_6] & v_4
 \end{array}$$

Fórmulas de saída :

$$\begin{array}{lll}
 (2, v_1) & v_2, v_4 & (1, v_2) & v_3 & (0, v_3) \\
 (2, v_4) & v_5, v_6 & (1, v_5) & v_2 & (0, v_6)
 \end{array}$$

Figura XVII-1. Um digrafo e as fórmulas de entrada e de saída para cada vértice.

Com relação ao exemplo da figura XVII-1, as fórmulas de saída obtidas a partir da fórmula de saída de v_1 por substituição descendente, até a representação final do digrafo, são:

$$\begin{array}{l}
 (2, v_1) \quad v_2, v_4 \\
 (2, v_1) \quad (1, v_2) \quad v_3, v_4 \\
 (2, v_1) \quad (1, v_2) \quad (0, v_3) \quad v_4 \\
 (2, v_1) \quad (1, v_2) \quad (0, v_3) \quad (2, v_4) \quad v_5, v_6 \\
 (2, v_1) \quad (1, v_2) \quad (0, v_3) \quad (2, v_4) \quad (1, v_5) \quad v_2, v_6 \\
 (2, v_1) \quad (1, v_2) \quad (0, v_3) \quad (2, v_4) \quad (1, v_5) \quad v_2 \quad (0, v_6) \quad .
 \end{array}$$

O vértice v_1 escolhido para se iniciar a substituição é tal que v_1 atinge todos os outros vértices do digrafo; se assim não o fosse, em vez de uma fórmula obteríamos um conjunto de fórmulas representando o digrafo.

XVII-3. Propriedades Básicas e Comentários

Se D é um digrafo tal que D possui exatamente um vértice com grau de entrada (saída) igual a zero, e todos os demais com grau de entrada igual a 1, então as fórmulas que o representam não têm rótulos — e portanto cada operador pode ser representado apenas pelo numeral que dá o grau do vértice ao qual se refere. Isso corresponderia a uma representação da árvore em questão por pré-ordem com graus. A fórmula de saída do digrafo obtido a partir do digrafo da figura XVII-1 pela supressão da aresta (v_5, v_2) pode então ser expressa por

$$2, 1, 0, 2, 0, 0 .$$

Para a representação de grafos não direcionados, pode-se atribuir uma orientação para cada aresta e representar o grafo direcionado assim obtido; ou pode-se considerar cada aresta do grafo como duas arestas simétricas de sentidos opostos e representar o grafo direcionado assim obtido. A figura XVII-2 mostra um exemplo de um grafo não direcionado sendo representado de várias maneiras diferentes.

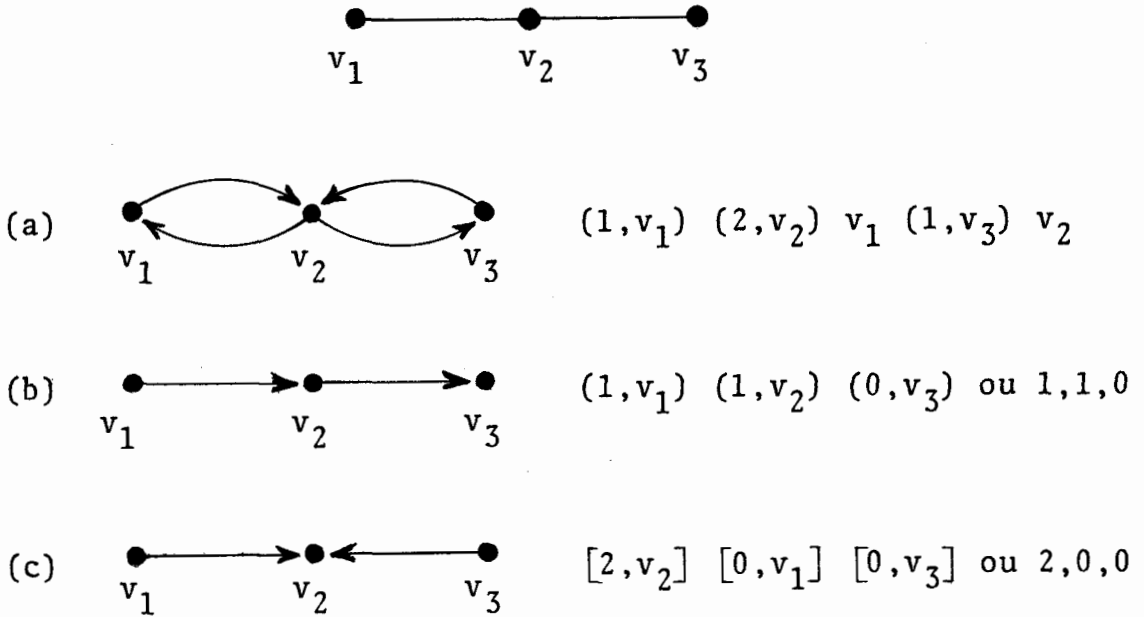


Figura XVII-2. Um grafo não direcionado e três maneiras de representá-lo por strings: (a) considerando dupla orientação das arestas; (b) atribuindo uma direção a cada aresta; (c) atribuindo uma orientação para cada aresta, de maneira diferente de (b).

A representação de pseudografos e/ou multigrafos pode ser feita sem alterações na definição, conforme pode ser visto na figura XVII-3.

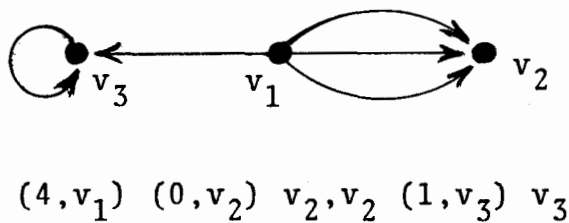


Figura XVII-3. Um pseudomultigrafo direcionado e uma representação por string (fórmula de saída).

Para armazenar esta representação, são necessárias $N+M+F$ posições de armazenamento, onde N é o número de vértices do digrafo, M é o número de arestas e F é o número de fórmulas obtidas. Observamos que F é no máximo igual ao número de componentes fortemente conexas do digrafo. Estamos considerando aqui apenas as posições necessárias para o armazenamento dos rótulos de vértices e numerais indicativos dos graus, uma vez que os delimitadores não necessitam de posições extra (por exemplo, basta termos um "tag" indicando se o elemento em questão é um rótulo ou um numeral).

XVII-4. Referências

Esta representação foi desenvolvida por Ercoli^{4,5}.

A representação por k -fórmula (capítulo XVI) constitui uma abordagem similar para a representação de digrafos, onde uma string de m asteriscos representa um operador m -ário (em vez de um numeral), e o primeiro rótulo após o asterisco representa o vértice ao qual o operador está aplicado, sendo usadas somente fórmulas de saída.

CAPÍTULO XVIII

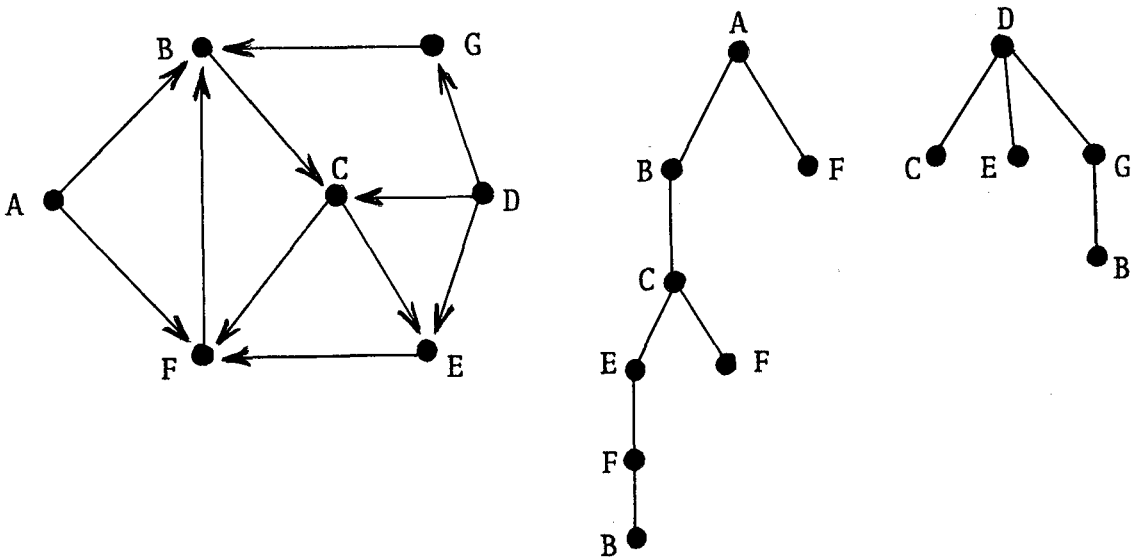
REPRESENTAÇÕES SEQUENCIAIS DE PSEUDO-ÁRVORES

No capítulo X, vimos que, dado um digrafo D , existe uma estrutura pseudo-árvore que o representa univocamente a menos de isomorfismos. Para a implementação dessa estrutura em computador, foi usada a transformação da floresta em árvore binária. Existem, no entanto, outras maneiras de se representar florestas. Apresentamos aqui algumas representações sequenciais de florestas, que naturalmente são também representações sequenciais de digrafos.

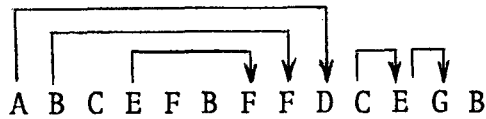
XVIII-1. Representações Sequenciais Ligadas

Dizemos que uma representação é sequencial ligada se faz uso da sequencialidade das informações e também de elos entre elas.

Apresentamos aqui dois tipos de representações sequenciais ligadas para florestas: representação sequencial em pré-ordem e representação sequencial em ordem-família. As definições e maiores detalhes para essas representações podem ser encontrados em Knuth⁹. A figura XVIII-1 mostra um exemplo dessas duas representações para um digrafo. Nesse exemplo, a pseudo-floresta foi construída supondo os rótulos dos vértices em ordem lexicográfica. Na representação sequencial em pré-ordem, além do campo de informação disposto sequencialmente, temos o campo de elo, que contém o apontador direito da árvore binária correspondente. Na representação sequencial em ordem-família, o campo de elo contém o apontador esquerdo da árvore binária correspondente.



Representação seqüencial em pré-ordem :



Representação seqüencial em ordem-família :

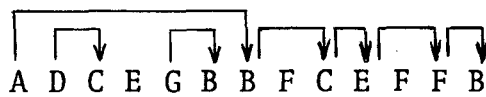


Figura XVIII-1. Um digrafo, uma representação por pseudo-árvore e as correspondentes representações seqüenciais em pré-ordem e ordem-família.

XVIII-2. Representação linear k-fórmula reversa em ordem-família

Seja D um digrafo cujos N vértices estão rotulados v_1, \dots, v_N .

Uma k-fórmula reversa pode ser definida recursivamente da seguinte maneira:

- (i) Um rótulo de um vértice é uma k-fórmula reversa.
- (ii) Se α e β são k-fórmulas reversas, então $\alpha\beta^*$ é uma k-fórmula reversa.

Dizemos que uma fórmula linear é uma representação k-fórmula reversa em ordem-família (KFROF) se é uma k-fórmula reversa e se os seus rótulos aparecem em ordem-família com res-

peito à pseudo-árvore do digrafo considerado.

Uma floresta é percorrida em k-fórmula reversa em ordem-família de acordo com a seguinte definição :

- (i) Se a primeira árvore da floresta for vazia, apenas retorne um "*". Caso contrário, visite a raiz da primeira árvore da floresta.
- (ii) Percorra as demais árvores em k-fórmula reversa em ordem-família.
- (iii) Percorra as sub-árvores da raiz da primeira árvore em k-fórmula reversa em ordem-família.

Como exemplo, a representação linear KFROF do digrafo da figura XVIII-1 é :

AD*CEG*B****BF**C*EF**F*B** .

Como foi visto no capítulo X, o número de nós da pseudo-árvore é $M + B$, onde M é o número de arestas do digrafo e B é o número de árvores na floresta. Como uma árvore binária qualquer com P nós possui $P+1$ sub-árvores vazias, o número de "*"s na representação linear KFROF é $M + B + 1$ - e portanto o número total de elementos na seqüência é $2(M + B) + 1$.

Para a obtenção da representação linear KFROF, a partir de uma dada pseudo-floresta, pode-se adaptar o algoritmo que percorre uma floresta em k-fórmula reversa em ordem-família. Por outro lado, dada uma representação linear KFROF, o algoritmo que obtém a pseudo-floresta que a gerou não é complicado. Uma idéia seria usar uma pilha contendo os rótulos de vértices na ordem em que aparecem na representação; o "*" comandaria a operação de retirada da pilha.

Observe que o número de "*"s é uma unidade maior que o número de rótulos de vértices. Aliás, a seqüência sempre termina com um "*", o qual poderia ser suprimido, mas será mantido por ser útil, por exemplo, para indicar o final da seqüência através da condição de pilha vazia.

Observe também que, se um vértice v é adjacente aos vértices v_1, \dots, v_p , então a seqüência de rótulos v_1, \dots, v_p aparece junta na representação linear KFROF, sem nenhum "*" entre esses rótulos.

Esta representação nos fornece uma maneira sistemática de percorrer um digrafo, que é na verdade uma estrutura do tipo labirinto.

XVIII-3. Representação linear k-fórmula reversa em pré-ordem

Dizemos que uma fórmula linear é uma representação k-fórmula reversa em pré-ordem (KFRPO) se é uma k-fórmula reversa e se os seus rótulos aparecem em pré-ordem com respeito à pseudo-árvore do digrafo considerado.

Uma floresta é percorrida em k-fórmula reversa em pré-ordem de acordo com a seguinte definição :

- (i) Se a primeira árvore da floresta for vazia, apenas retorne um "*". Caso contrário, visite a raiz da primeira árvore da floresta.
- (ii) Percorra as sub-árvores da primeira árvore em k-fórmula reversa em pré-ordem.
- (iii) Percorra as demais árvores em k-fórmula reversa em pré-ordem.

Como exemplo, a representação linear KFRPO do digrafo da figura XVIII-1 é :

ABCEFB***F***F**DC*E*GB**** .

Sendo esta representação similar à representação linear KFROF, as propriedades válidas para essa última permanecem válidas para a KFRPO, com exceção da propriedade referente ao conjunto de vértices adjacentes a um dado vértice, que aparecem juntos na KFROF.

XVIII-4. Representação linear k-fórmula reversa em ordem de nível

Dizemos que uma fórmula linear é uma representação k-fórmula reversa em ordem de nível (KFRON) se é uma k-fórmula reversa e se os seus rótulos aparecem em ordem de nível com relação à pseudo-árvore do digrafo considerado. Estamos considerando aqui a floresta sendo percorrida da raiz para as folhas ; o sentido oposto (das folhas para a raiz) é similar.

Para a construção da seqüência KFRON, inicialmente acrescentamos um nó auxiliar (extra) contendo um asterisco ("*") como informação, como filho de cada nó terminal das árvores e como irmão mais à direita de cada nó. Agora percorre-se a floresta em ordem de nível, da raiz para as folhas, concatenando as seqüências assim obtidas em ordem de aparição.

Como exemplo, a figura XVIII-2 apresenta uma pseudo floresta correspondente ao digrafo da figura XVIII-1 já contendo os nós extra.

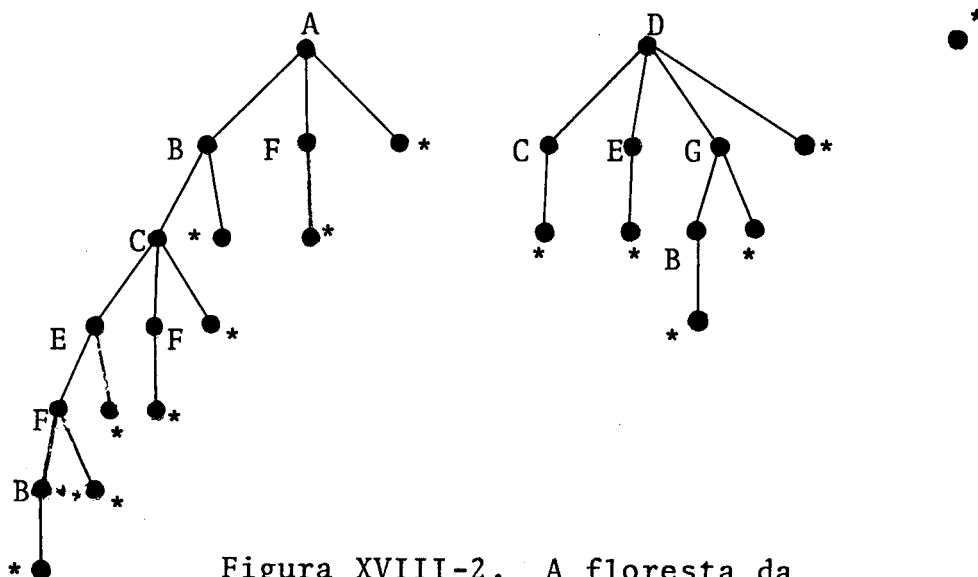


Figura XVIII-2. A floresta da figura XVIII-1 acrescida de nós auxiliares contendo "*"s.

A representação linear KFRON correspondente à figura XVIII-2 é :

AD*BF*CEG*C****B*EF**F**B** .

De acordo com a definição, a seqüência KFRON tem $2(M + B) + 1$ elementos, onde M é o número de arestas do digrafo e B é o número de árvores na floresta da representação pseudo-árvore correspondente.

Um algoritmo para obter a representação KFRON a partir da pseudo-árvore, e outro para obter a pseudo-árvore a partir da representação KFRON, usaria uma fila em vez de uma pilha; a idéia desta representação é mais iterativa do que recursiva.

XVIII-5. Referências

Outras representações diferentes na mesma linha poderiam ser definidas, variando o modo de percorrer a pseudo-árvore. As representações seqüenciais para florestas em pré-ordem, ordem-família e ordem de nível podem ser encontradas em Knuth⁹. As representações definidas neste capítulo podem ser encontradas em Szwarcfiter¹⁷.

CAPÍTULO XIX

REPRESENTAÇÕES SEQUÊNCIAIS TERNÁRIASXIX-1. Representações de florestas por seqüências binárias

Definimos uma árvore-b ou árvore estritamente binária como sendo uma árvore binária na qual cada nó possui zero ou dois filhos.

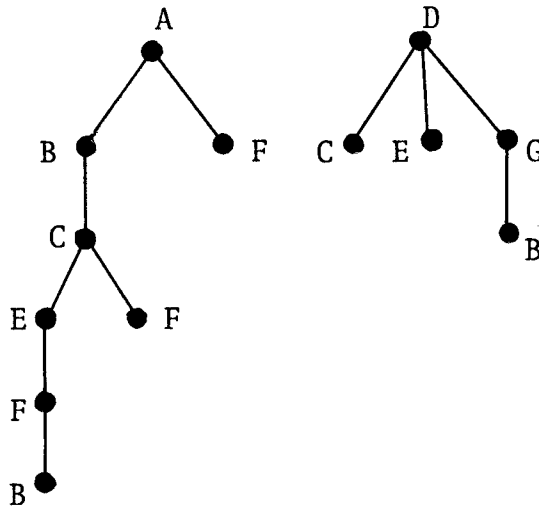
A cada árvore-b corresponde univocamente uma seqüência binária que a representa. Por outro lado, a cada floresta ordenada enraizada com n nós, corresponde univocamente uma árvore binária com n nós, e a cada árvore binária com n nós corresponde univocamente uma árvore-b com $2n+1$ nós. Portanto, por meio destas correspondências é possível obter uma representação por seqüência binária de qualquer floresta (ordenada enraizada).

Apresentamos aqui métodos para a obtenção de diferentes representações por seqüência binária de uma floresta ordenada enraizada, baseados nas representações lineares do capítulo XVIII (KFROF, KFRPO e KFRON).

As representações KFROF, KFRPO e KFRON são basicamente representações lineares de digrafos onde aparecem dois tipos de elementos: rótulos de vértices e asteriscos ("*"). Se todos os nós de uma floresta forem representados pelo mesmo rótulo, a estrutura da floresta se manterá, embora a informação contida em cada nó seja perdida. A seqüência resultante dessa substituição não mais representa um digrafo, mas ainda representa o aspecto topológico da floresta.

Assim, se substituirmos todos os "*"s por 0's e todos os rótulos de vértices por 1's, obteremos três tipos diferentes de representações de florestas por seqüências binárias, correspondente às representações lineares KFROF, KFRPO e KFRON, que chamaremos respectivamente de KFROF binária, KFRPO binária

e KFRON binária. A figura XIX-1 abaixo contém exemplos.



KFROF: AD*CEG*B****BF**C*EF**F*B**

KFRPO: ABCEFB***F***F**DC*E*GB****

KFRON: AD*BF*CEG*C****B*EF**F**B**

KFROF binária: 110111010000110010110010100

KFRPO binária: 111111000100010011010110000

KFRON binária: 110110111010000101100100100

Figura XIX-1. Uma floresta e representações binárias dessa floresta.

Estas representações binárias têm n dígitos '1' e $n+1$ dígitos '0', onde n é o número de nós da floresta. Na sequência KFROF binária, o número de 1's do começo da sequência é igual ao número de árvores da floresta.

Os algoritmos para a obtenção dessas representações a partir da floresta, e para a construção da floresta a partir dessas representações, são similares àqueles que executam essas tarefas para as representações lineares KFROF, KFRPO e KFRON.

XIX-2. Representações de digrafos por seqüências ternárias

Observando as representações de florestas KFROF, KFRPO e KFRON binárias, podemos ver que elas não representam digrafos, pois não há maneira de se concluir, pela representação, se dois elementos com valor 1 representam ou não o mesmo vértice do digrafo.

Para que seja possível identificar na seqüência os elementos que correspondem a um mesmo vértice, podemos substituir todos os 1's que correspondem a elementos repetidos na seqüência por uma subseqüência composta de 2's, 1's e 0's que indicaria a posição da primeira ocorrência do elemento na seqüência. Desta forma é possível transformar as representações lineares de digrafos KFROF, KFRPO e KFRON em representações seqüenciais ternárias de digrafos, chamadas respectivamente de KFROF ternária, KFRPO ternária e KFRON ternária.

XIX-2-1. Representação KFROF ternária

Seja D um digrafo cujos vértices estão rotulados v_1, \dots, v_N .

Os passos necessários para a obtenção da representação KFROF ternária são os seguintes :

- (i) Obtenha uma representação pseudo-árvore de D .
- (ii) Obtenha a representação KFROF.
- (iii) Obtenha a representação KFROF binária.
- (iv) Obtenha agora a representação KFROF ternária, através do seguinte procedimento:
 Suponha que um vértice v aparece mais de uma vez na representação KFROF.
 Seja $B_1(v)$ a subseqüência binária que vai do início da representação KFROF binária até o

elemento '1' correspondente à primeira ocorrência de v na KFROF (inclusive).

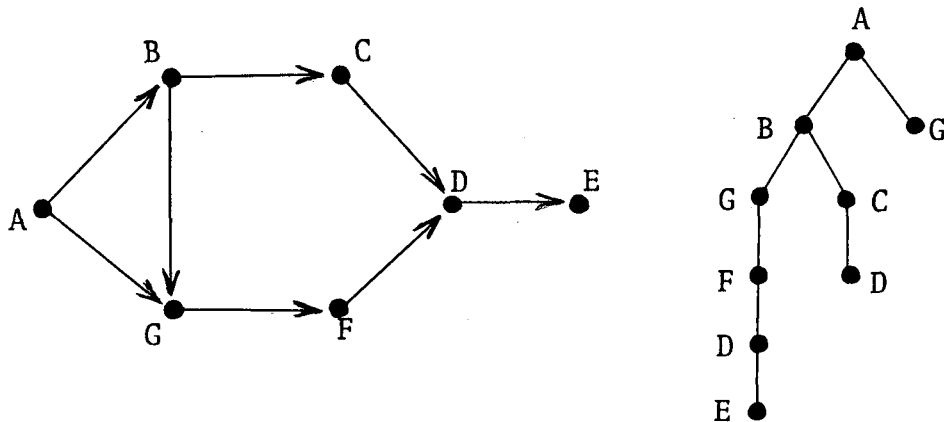
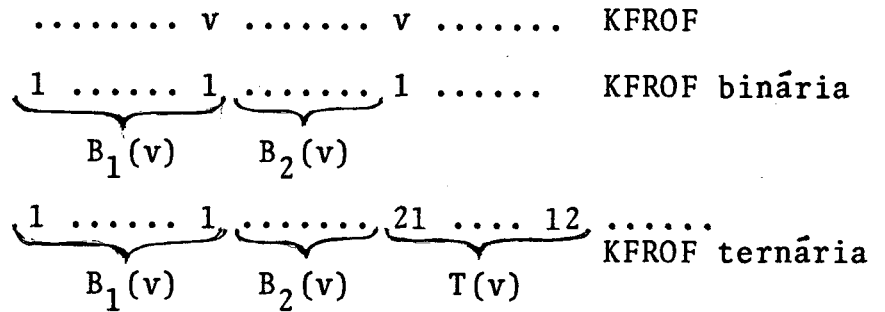
Seja $B_2(v)$ a subsequência binária que começa imediatamente após $B_1(v)$ e termina imediatamente antes da segunda ocorrência de v na KFROF.

Seja agora $T(v)$ a seqüência obtida a partir de $B_1(v)$ pelo acréscimo de um '2' no início e no fim de $B_1(v)$.

Substitua todas as ocorrências repetidas de v por $T(v)$, retirando o '1' correspondente.

Repita este passo para todas as ocorrências de vértices repetidos na representação KFROF.

O procedimento descrito aqui pode ser esquematizado pela seguinte figura :



KFROF: A*BG**GC*D**F*D*E**
 KFROF binária: 1011001101001010100
 KFROF ternária: 10110021011210100102101100110120100

Figura XIX-2. Um digrafo, uma representação por pseuo-árvore e a correspondente representação KFROF ternária.

O algoritmo para a construção da representação KFROF ternária a partir de um digrafo D é o mesmo algoritmo para a obtenção da KFROF binária acrescido de um procedimento relativo ao passo (iv) da definição, que introduz as subsequências ternárias.

O algoritmo para a obtenção do digrafo, a partir da representação KFROF ternária, é também uma extensão do algoritmo que obtém a floresta dada a KFROF binária, adicionado dos passos necessários para a avaliação de cada subsequência ternária.

Como o número de nós da pseudo-floresta é $M+B$ (onde M é o número de arestas do digrafo e B é o número de árvores na floresta), e como cada elemento repetido na representação linear KFROF produz exatamente dois 2's na KFROF ternária, o número total de 2's nesta representação é $2(M+B-N)$, onde N é o número de vértices do digrafo. Se o digrafo em questão for uma árvore enraizada, então $M = N-1$ e $B=1$, e portanto a KFROF binária e a KFROF ternária são iguais.

Poderíamos definir uma sequência ternária um pouco mais compacta, redefinindo a subsequência $T(v)$ de tal forma que $T(v)$ passa a ser $20\dots02$, com tantos zeros quantos forem os vértices que antecederem v na KFROF. Para a figura XIX-2, esta redefinição produziria a seguinte KFROF ternária :

1011002002101001020000020100 .

Contudo, esta redefinição tem a desvantagem de obrigar o algoritmo que constrói o digrafo a partir da KFROF ternária a retornar ao início da representação para avaliar cada subsequência ternária $T(v)$. Isto não acontecia com a definição original pois a própria subsequência ternária fornecia meios de se obter informações sobre a estrutura da pseudo-floresta.

Uma vez que esta representação pode gerar sequências ternárias bastante longas, seu interesse é mais teórico do que para aplicação direta em manipulação de digrafos em computadores.

XIX-2-2. Representação KFRPO ternária

O estudo da representação KFRPO ternária é similar ao da representação KFROF ternária, e não será apresentado aqui.

Para o digrafo e a pseudo-árvore mostrados na figura XIX-2, a representação KFRPO ternária é :

KFRPO: ABGFDE****CD***G***
 KFRPO binária: 1111110000110001000
 KFRPO ternária: 11111100001211111200021112000

XIX-2-3. Representação KFRON ternária

A representação KFRON ternária é também similar às duas anteriores, e para o digrafo e a pseudo-árvore da figura XIX-2, a representação KFRON ternária é :

KFRON: A*BG*GC**F*D*D**E**
 KFRON binária: 1011011001010100100
 KFRON ternária: 1011021011210010102101101100101200100

XIX-3. Referências

Representações binárias de árvores podem ser encontradas em Page & Wilson¹¹, Scoins¹⁵ e Berge². As representações definidas aqui podem ser encontradas em Szwarcfiter¹⁷.

CAPÍTULO XX

CONCLUSÃO

Vários tipos de representações para grafos foram apresentadas, algumas com características bem diversas das outras, o que torna bastante difícil uma comparação geral entre todas elas. Muitas dessas representações apresentam características práticas e adequadas a manipulações, outras já revelam um interesse mais teórico; outras ainda se apresentam úteis para o armazenamento temporário de grafos (em memória auxiliar, por exemplo) devido ao pouco espaço que ocupam.

Este capítulo se propõe a apresentar uma comparação de todas as representações contidas neste trabalho em termos de quantidade de espaço ocupado.

A tabela XX-1 contém, para cada representação do trabalho, uma fórmula do espaço requisitado pela representação e o correspondente valor obtido para a representação do grafo da figura XX-1. Para a representação desse grafo foi considerada a rotulação dos vértices indicada na figura, com os rótulos ordenados lexicograficamente.

Nas fórmulas apresentadas na tabela XX-1, temos :

N = número de vértices,

M = número de arestas,

S = número de sumidouros,

B = número de árvores na representação por pseudo-
árvore (capítulo X),

X = número de vértices auxiliares que precisam ser
acrescentados para a representação por árvore bi-
nária (capítulo XI),

L = número de listas na representação por estrutura
de lista (capítulo XIII),

K = número de fórmulas na representação por k -fórmula (capítulo XVI),

F = número de fórmulas na representação por strings (capítulo XVII).

No caso do digrafo da figura XX-1, temos $N=8$, $M=5$, $S=0$, $B=1$, $X=2$, $L=1$, $K=1$, $F=1$.

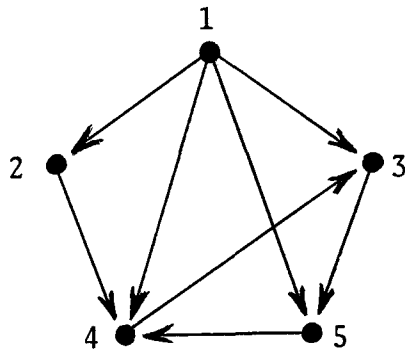


Figura XX-1. Um digrafo e uma rotulação de seus vértices.

Tabela XX-1. Espaço requisitado para o armazenamento de cada representação.

Representação	Fórmula do espaço requisitado	Espaço para o digrafo da fig.XX-1	Observações
Matriz de adjacência (cap.II)	N^2	25	informações binárias
Matriz de incidência (cap.III)	$M.N$	40	informações ternárias
Matriz de posicionamento (cap.IV)	M^2	64	informações ternárias
Representação orientada graficamente (cap.V)	M^2	64	informações quaternárias
Estrutura de adjacência (cap.VI)	$2M$	16	sem cabeças de linha ou coluna
Estrutura de incidência (cap.VII)	$4M$	32	sem cabeças de linha ou coluna
Estrutura de posicionamento (cap.VIII)	$2 \left[-M + \frac{1}{2} \sum_{i=1}^N \text{grau}^2(v_i) \right]$	38	sem cabeças de linha ou coluna
Representação vetorial de arestas (cap.IX)	$2M$	16	
Pseudo-árvore (cap.X)	$3(M+B)$	27	
Representação por árvore binária (cap.XI)	$2(N+X)$	14	
Árvore-coárvore (cap.XII)	$4(2M-N+2)$	52	
Estrutura de lista (cap.XIII)	$2(M+N+L)$	28	

(continua)

(Tabela XX-1 - continuação)

Representação	Fórmula do espaço requisitado	Espaço para o digrafo da fig.XX-1	Observações
Lista de ramos (cap.XIV)	$M+N$	13	
Seqüência de adjacência (cap.XV)	$M+S$	8	
Representação por k-fórmula (cap.XVI)	$2M+K$	17	
Representação por strings (cap.XVII)	$N+M+F$	14	
Representações seqüenciais ligadas (cap.XVIII)	$2(M+B)$	18	
Representações por k-fórmula reversa (cap.XVIII)	$2(M+B)+1$	19	
KFROF ternária (cap.XIX)		43	informações ternárias
KFRPO ternária (cap.XIX)		38	informações ternárias
KFRON ternária (cap.XIX)		43	informações ternárias

REFERÊNCIAS BIBLIOGRÁFICAS

1. BAYS, C. - A non-recursive technique for recreating a di-graph from its k-formula representation. The Computer Journal, 19, pp.326-328, 1976.
2. BERGE, C. - The Theory of Graphs and its Applications, J. Wiley, London, 1962.
3. BERZTISS, A.T. - Data Structures - Theory and Practice , Academic Press, New York, 1971.
4. ERCOLI, P. - A String Representation of Graphs and Some of its Applications. Rapporti dell'Instituto di Automatica dell'Universitá di Roma e del Centro di Studio dei Sistemi di Controllo e Calcolo Automatici del C.N.R., Roma, 1972.
5. ERCOLI, P. - Implementation of Formulae for Graph Representation and Some Applications. Rapporti dell'Instituto di Automatica dell'Universitá di Roma e del Centro di Studio dei Sistemi di Controllo e Calcolo Automatici del C.N.R., Roma, 1974.
6. HARARY, F. - Graph Theory, Addison-Wesley, Massachusetts , 1969.
7. HOPCROFT, J. & TARJAN, R. - Efficient Algorithms for Graph Manipulation, Stanford University, California, STAN-CS-71-207, 1971.
8. HOPCROFT, J. & TARJAN, R. - Efficient Planarity Testing . J.ACM, 21, pp.549-568, 1974.

9. KNUTH, D.E. - The Art of Computer Programming, Volume 1 , Fundamental Algorithms, Addison-Wesley, Massachusetts, 1968.
10. OBRUKA, K.O. - The Manipulation of Trees and Linear Graphs Within a Computer, and Some Applications. Newcastle Upon Tyne, University of Newcastle Upon Tyne, Ph. D. Thesis, 1966.
11. PAGE, E. & WILSON, L. - Information Representation and Manipulation in a Computer, Cambridge University Press , London, 1974.
12. PFALTZ, J.L. - Representing Graphs by Knuth Trees. J.ACM, 22, pp.361-366, 1975.
13. PFALTZ, J.L. - Computer Data Structures, McGraw-Hill, New York, 1977.
14. SCOINS, H.I. - Linear Graphs and Trees. *In* Machine Intelligence 1, N.L.Collins and D.Michie ed., Oliver & Boyd, Edinburgh and London, 1967.
15. SCOINS, H.I. - Placing Trees in Lexicographical Order. *In* Machine Intelligence 3, N.L.Collins and D.Michie ed. , Oliver _ Boyd, Edinburgh and London, 1968.
16. SNOW, C.R. & SCOINS, H.I. - Towards the Unique Decomposition of Graphs. *In* Machine Intelligence 4, N.L.Collins and D.Michie ed., Oliver & Boyd, Edinburgh and London, 1969.
17. SZWARCFITER, J.L. - On Representation of Graphs and Digraphs. Manuscrito não publicado, 1973.
18. WELLS, M.B. - Elements of Combinatorial Computing, Pergamon Press, Oxford, 1971.