Se o número de chaves N e o tamanho da tabela M são iguais (isto é, $\alpha = N/M = 1$), então temos uma **função de transformação perfeita mínima**, isto é, apenas um acesso à tabela é necessário e não há lugares vazios na tabela.

Finalmente, se $x_i \leq x_j$ e $hp(x_i) \leq hp(x_j)$, então a ordem lexicográfica é preservada. Nesse caso, temos uma **função de transformação perfeita mínima com ordem preservada**, na qual as chaves são localizadas em um acesso, não há espaço vazio na tabela e o processamento é realizado na ordem lexicográfica.

Qual a vantagem da função de transformação perfeita? Nas aplicações em que necessitamos apenas recuperar o registro com informação relacionada com a chave e a pesquisa é sempre com sucesso, não há necessidade de armazenar o conjunto de chaves, pois qualquer registro pode ser localizado a partir do resultado da função de transformação.

Uma função de transformação perfeita é específica para um conjunto de chaves conhecido, ao contrário da função de transformação universal apresentada no Programa 5.23. Em outras palavras, ela não pode ser uma função genérica e tem de ser pré-calculada. Existem duas vantagens no uso de uma função de transformação perfeita mínima: não existem colisões e não existe desperdício de espaço pois todas as entradas da tabela são ocupadas. Uma vez que colisões não ocorrem, cada chave pode ser recuperada da tabela com um único acesso. Assim, uma função de transformação perfeita mínima evita completamente o problema de desperdício de espaço e de tempo. A desvantagem no caso é o espaço ocupado para descrever a função de transformação hp.

Czech, Havas e Majewski (1992, 1997) propõem um método elegante baseado em hipergrafos randômicos para obter uma função de transformação **perfeita mínima com ordem preservada**. Como mostrado na Seção 7.10, um **hipergrafo ou r−grafo** é um grafo não direcionado no qual cada aresta conecta r vértices. A função de transformação é do tipo:

$$hp(x) = (g[h_0(x)] + g[h_1(x)] + \ldots + g[h_{r-1}(x)]) \bmod N,$$

na qual $h_0(x), h_1(x), \ldots, h_{r-1}(x)$ são r funções não perfeitas descritas pelo Programa 5.23, x é a chave de busca, e g um arranjo especial que mapeia números no intervalo $0 \ldots M-1$ para o intervalo $0 \ldots N-1$.

O algoritmo resolve o problema descrito a seguir. Dado um hipergrafo não direcionado acíclico $G_r = (V, A)$, onde $|V| = M$ e $|A| = N$, encontre uma atribuição de valores aos vértices de G_r tal que a soma dos valores associados aos vértices de cada aresta tomado módulo N é um número único no intervalo $[0, N-1]$. A questão principal é como obter uma função g adequada. A abordagem mostrada a seguir é baseada em hipergrafos acíclicos randômicos.

Vamos considerar um exemplo constituído dos 12 meses do ano, abreviados com os três primeiros caracteres. Para o exemplo vamos utilizar um hipergrafo acíclico com $r = 2$ (ou 2-grafo), onde cada aresta conecta 2 vértices. Nesse caso vamos precisar de duas funções de transformação universais $h_0(x)$ e $h_1(x)$ descritas pelo Programa 5.23. O objetivo é obter uma função de transformação perfeita hp de tal forma que o i-ésimo mês é mantido na $(i-1)$-ésima posição da tabela *hash*, como mostrado na Tabela 5.3(a). Na tabela, os valores $h_0(x)$ e $h_1(x)$ foram obtidos por duas funções de transformação representadas por dois conjuntos diferentes de pesos: p_0 e $p1$, respectivamente.

***Tabela 5.3** Tabelas para obter uma função de transformação perfeita: (a) chaves e funções* hash; *(b) arranjo g*

(a)

Chave x	$h_0(x)$	$h_1(x)$	$hp(x)$
jan	11	14	0
fev	14	2	1
mar	0	10	2
abr	8	7	3
mai	4	12	4
jun	14	6	5
jul	1	7	6
ago	12	10	7
set	11	4	8
out	8	13	9
nov	3	4	10
dez	1	5	11

(b)

v	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$g[v]$	3	3	1	2	8	8	5	3	12	-1	11	12	8	9	0

O problema de obter a função g é equivalente a encontrar um hipergrafo não direcionado acíclico contendo M vértices e N arestas. O grafo da Figura 5.16, obtido para o exemplo dos 12 meses do ano, contém $M = 15$ vértices e $N = 12$ arestas. Assim, o primeiro passo é obter um hipergrafo randômico e verificar se ele é acíclico. O Programa 7.10 para verificar se um hipergrafo é acíclico é baseado no fato de que um r-grafo é **acíclico** se e somente se a remoção repetida de arestas contendo vértices de grau 1 elimina todas as arestas do grafo.

Em um hipergrafo $G_r(V, A)$, os vértices são rotulados com valores no intervalo $0 \ldots M-1$ e as arestas definidas por $(h_0(x), h_1(x), \ldots, h_{r-1}(x))$ para cada uma das N chaves x. Assim, cada chave corresponde a uma aresta rotulada com o valor desejado para a função hp perfeita, e os valores das r funções $h_0(x), h_1(x), \ldots, h_{r-1}(x)$ definem os vértices sobre os quais a aresta é incidente. Um passo importante para obter a função hp é conseguir um arranjo g tal que, para cada aresta $(h_0(x), h_1(x), \ldots, h_{r-1}(x))$, o valor de $hp(x) = (g[h_0(x)] + g[h_1(x)] + \ldots + g[h_{r-1}(x)]) \bmod N$ seja igual ao rótulo da aresta.

A Tabela 5.3(b) mostra o arranjo g obtido para o 2-grafo da Figura 5.16. Para cada aresta $a = (v_0, v_1, \ldots, v_{r-1})$, onde $v_i = h_i(x)$ para $0 \leq i \leq r-1$, temos que atribuir valores aos vértices $v_0, v_1, \ldots, v_{r-1}$ tal que $(g[v_0] + g[v_1] +$