

2

Propriedades estruturais

Provaremos neste capítulo algumas propriedades de uma atribuição de hotlinks P-ótima. Estas propriedades serão importantes nas análises dos algoritmos propostos para o PBH no Capítulo 3. Na seção seguinte provamos limites superiores para $h^*(T)$ e $H^*(T)$. Apresentamos limites inferiores para $h^*(T)$ na Seção 2.2.

2.1

Limites superiores

Para provar limites superiores para $h^*(T)$ e $H(T^{A^*})$, introduzimos o algoritmo ASSIGN-HOTLINK.

Algoritmo ASSIGN-HOTLINK

O algoritmo ASSIGN-HOTLINK lê a árvore de entrada T e inicializa o conjunto de hotlink A . Em seguida, ele chama o procedimento recursivo LOG, que escreve em A uma atribuição de hotlinks para a árvore T .

O procedimento LOG atribui um hotlink do nó raiz para um nó u tal que as sub-árvores geradas sejam “balanceadas”. No nosso caso, uma sub-árvore é balanceada se não tem mais que $\max\{t, (1 - t)\}|L|$ hotleaves, onde t é uma constante cujo valor será determinado na análise. Finalmente, este procedimento processa recursivamente cada uma das sub-árvores balanceadas.

A idéia chave de LOG é gastar no máximo dois níveis da árvore melhorada para reduzir por um fator constante o número de hotleaves nos novos sub-problemas. A Figura 2.2.(a) mostra uma árvore T com raiz em r e a Figura 2.2.(b) mostra a árvore $T^{\{(r,u)\}}$. Em $T^{\{(r,u)\}}$, as sub-árvores com raiz nos filhos de r , exceto u , têm no máximo $(1 - t)|L|$ hotleaves. As sub-árvores com raiz nos nós que são filhos de u têm no máximo $t|L|$ hotleaves.

Note que a raiz de cada uma destas sub-árvores está no máximo dois níveis a mais que r em $T^{\{(r,u)\}}$.

De modo a decidir para onde apontar o hotlink de r , LOG mantêm um vetor \mathbf{w} . Para um nó v em uma sub-árvore T' de T , $w(v, T')$ fornece o número de descendentes de v que pertencem a L . Deve ficar claro que $w(v, T')$ não é necessariamente igual ao número de folhas em T' que são descendentes de v . Um pseudo-código para ASSIGN-HOTLINK é apresentado na Figura 2.1.

Algoritmo 1: LOG

Entrada: T : árvore direcionada

Saída: A : atribuição de hotlinks

- (1) $r \leftarrow$ raiz de T
- (2) **if** T tem exatamente uma hotleaf l
- (3) $A \leftarrow A \cup \{(r, l)\}$
- (4) **else if** T tem exatamente duas hotleaves l e l'
- (5) $A \leftarrow A \cup \{(r, l), (s, l')\}$, onde s é o ancestral de l' em $S(r, T)$
- (6) **else if** T tem mais de duas hotleaves
- (7) Execute uma busca em profundidade em T para calcular o valor de $w(v, T)$, para todo nó $v \in T$.
- (8) Encontre um nó u em T tal que $w(u, T) \geq tw(r, T)$, e $w(v, T) < tw(r, T)$ para todo $v \in S(u, T)$.
- (9) $A \leftarrow A \cup \{(r, u)\}$.
- (10) **if** $u \neq r$ **then** seja s_1 o ancestral de u em $S(r, T)$
- (11) **if** s_1 é definido **and** $s_1 \neq u$ **then** LOG($T_{s_1} - T_u$)
- (12) **foreach** $v \in S(u, T) \cup S(r, T) - \{s_1\}$ **do** LOG(T_v)

Algoritmo 2: ASSIGN-HOTLINK

- (1) Leia T ; $L \leftarrow$ conjunto de folhas de T ; $A \leftarrow \emptyset$
- (2) LOG(T)

Figura 2.1: Pseudo-código do algoritmo ASSIGN-HOTLINK.

Corretude

Para mostrar que o algoritmo está correto, precisamos garantir que sempre é possível encontrar o nó u na linha 8 do pseudo-código. Para isso, usamos o Lema 2.1 abaixo. De agora em diante, assumimos que $t \in (1/3, 1)$.

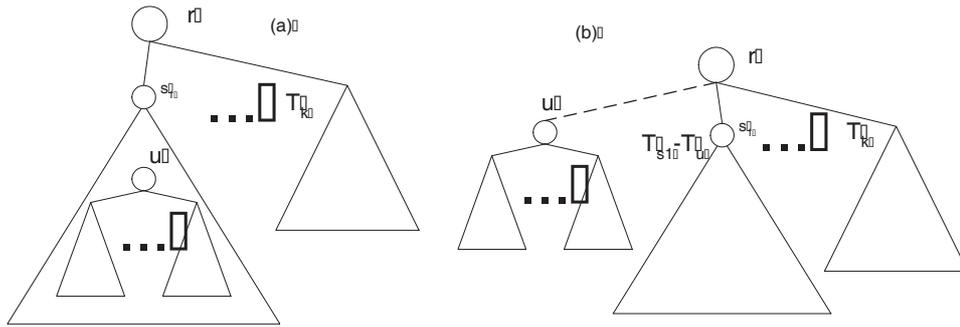


Figura 2.2: (a) Árvore T . (b) Árvore $T^{(r,u)}$ obtida pelo procedimento LOG após adicionar o hotlink (r, u) em T

Lema 2.1 *Seja T' uma sub-árvore de T com raiz em r' , tendo $w(r', T') \geq 3$. Então, existe um nó $u \in T'$ tal que $w(u, T') \geq tw(r, T')$, e $w(v, T') < tw(r, T')$ para todo $v \in S(u, T')$.*

Prova. Como $t \leq 1$, temos que $w(r', T') \geq tw(r', T')$. Seja u o nó em T' com nível máximo, tal que $w(u, T') \geq tw(r', T')$. Como $t > 1/3$ e $w(r', T') \geq 3$, concluímos que u não é uma folha. Pela escolha de u , temos que $w(v, T') < tw(r', T')$ para todo $v \in S(u, T')$. \square

Análise

Precisamos agora mostrar como o hotlink da raiz, apontado para o nó u escolhido na linha 8, pode gerar sub-árvores balanceadas. Isto é demonstrado no Lema 2.2 abaixo.

Lema 2.2 *Seja u o nó selecionado na linha 8 quando LOG é executado na sub-árvore T' de T com raiz em r' . Então,*

- (i) $w(s_1, T'_{s_1} - T'_u) \leq (1 - t)w(r', T')$;
- (ii) $w(v, T'_v) \leq (1 - t)w(r', T')$, $\forall v \in S(r', T') - S(u, T') - \{s_1\}$;
- (iii) $w(v, T'_v) < tw(r', T')$, $\forall v \in S(u, T')$.

Prova. As condições (i) e (ii) podem ser verificadas examinando a Figura 2.2.(b), que apresenta a árvore T depois da atribuição do hotlink (r, u) . Nesta árvore, todas as sub-árvores com raiz em filhos de r (exceto u) têm no máximo $(1 - t)w(r, T)$ hotleaves, pois o número de hotleaves em T_u é no mínimo $tw(r, T)$. A condição (iii) decorre diretamente da escolha de u . \square

Com o resultado do Lema 2.2 podemos provar o Teorema 2.3, que estabelece um limite superior para o nível máximo das hotleaves na árvore melhorada pela execução do procedimento LOG.

Teorema 2.3 *Seja A' é o conjunto de hotlinks adicionados em A durante a execução de $LOG(T')$, e $h_{LOG}(T') = h(T'^{A'})$. Temos que $h_{LOG}(T') \leq c \log w(r', T') + 1$, onde*

$$c = \max \left\{ \frac{2}{\log \left(\frac{1}{t} \right)}, \frac{1}{\log \left(\frac{1}{1-t} \right)} \right\}.$$

Prova. Provamos por indução no valor de $w(r', T')$. Se $w(r', T') = 1$ então $h_{LOG}(T') = 1 = c \log w(r', T') + 1$. Se $w(r', T') = 2$ então $h_{LOG}(T') \leq 2 < c + 1$ (o intervalo de valores de t deve ser considerado aqui).

Agora, vamos assumir que o resultado vale para o caso onde $w(r', T') < k$. Com base nisso, provaremos que ele vale também para o caso onde $w(r', T') = k$, com $k \geq 3$. Como $w(r', T') \geq 3$ e $1/3 < t < 1$, o Lema 2.1 garante que o nó u é encontrado na linha 8 do pseudo-código de LOG. A atribuição do hotlink (r', u) e as linhas 11 e 12 do pseudo-código implicam que

$$\begin{aligned} h_{LOG}(T') \leq \max \{ & \max \{ 1 + h_{LOG}(T'_v) \mid v \in S(r', T') - S(u, T') - \{s_1\} \}, \\ & \max \{ 2 + h_{LOG}(T'_v) \mid v \in S(u, T') \}, \\ & 1 + h_{LOG}(T'_{s_1} - T'_u) \} \end{aligned} \quad (2-1)$$

A relação dada pelo Lema 2.2 e a hipótese indutiva nos permitem dar um limite superior para o lado direito da equação anterior. Assim,

$$\begin{aligned} h_{LOG}(T') & \leq \max \{ 1 + c \log((1-t)w(r', T')) + 1, \\ & \quad 2 + c \log(tw(r', T')) + 1 \} \\ & = c \log w(r', T') + 1 + \\ & \quad \max \left\{ 1 - c \log \left(\frac{1}{1-t} \right), 2 - c \log \left(\frac{1}{t} \right) \right\}. \end{aligned} \quad (2-2)$$

Provamos este teorema substituindo a segunda e a terceira ocorrências de c no lado direito da equação anterior por $\frac{1}{\log(1/(1-t))}$ e $\frac{2}{\log(1/t)}$, respectivamente. \square

Limites superiores para $h^*(T)$ e $H^*(T)$

Devemos escolher um valor de t que maximiza o limite superior dado pelo Teorema 2.3. Utilizando cálculo, encontramos $t = \frac{3-\sqrt{5}}{2} > 0.38$. Para

este valor de t temos que $c = \frac{2}{1 - \log(3 - \sqrt{5})} < 1.441$. Com isso, temos no corolário a seguir um limite superior para $h^*(T)$.

Corolário 2.4 $h^*(T) \leq h_{LOG}(T) \leq 1.441 \log w(r, T) + 1 = 1.441 \log m + 1$.

Este limite superior para $h^*(T)$ não serve como limite superior para $H^*(T)$. Isto porque alguns nós internos podem ter nível maior que $h^*(T)$ na árvore melhorada, como ilustra a Figura 2.3.(a). Para obter um limite superior para $H^*(T)$, precisamos limitar a altura destas sub-árvores cujos nós têm nível maior ou igual a $h^*(T)$. Para isso, modificamos o algoritmo LOG fazendo o vetor \mathbf{w} armazenar o número de nós das sub-árvores. Chamamos este novo algoritmo de LOG Modificado. Neste caso, afirmamos que o algoritmo ASSIGN-HOTLINK constrói uma atribuição de hotlinks A tal que a altura de T^A não é maior que $1.441 \log n + 1$. A prova desta afirmação é análoga a do Teorema 2.3. Com isso temos um algoritmo que fornece um limite superior para todos os nós, não apenas os nós folha. O próximo teorema usa este resultado para dar um limite superior para a altura mínima da árvore melhorada obtida por uma atribuição de hotlinks P-ótima.

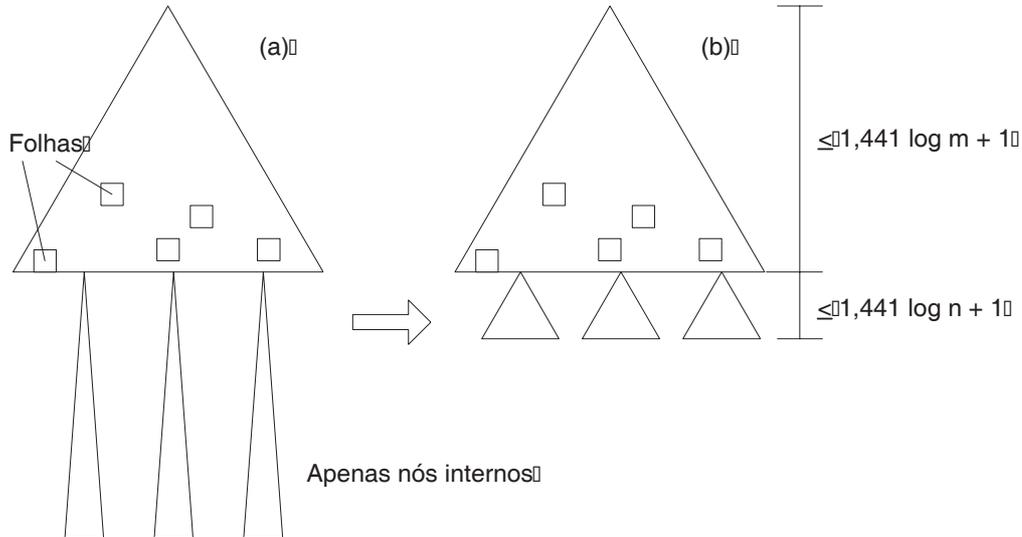


Figura 2.3: (a) Árvore melhorada, depois de executar ASSIGN-HOTLINK. (b) Árvore obtida da árvore anterior depois de executar o algoritmo LOG Modificado nas sub-árvores com apenas nós internos.

Teorema 2.5 *Existe uma atribuição de hotlinks P-ótima A^* tal que $H(T^{A^*}) \leq 1.441(\log m + \log n) + 2$.*

Prova. Seja A' uma atribuição de hotlinks P-ótima para T . Provamos este teorema construindo outra atribuição de hotlinks P-ótima A^* utilizando A' . Inserimos em A^* todos os hotlinks $(u, v) \in A'$ tal que u está em um nível menor que $h^*(T)$ em $T^{A'}$. Neste ponto, já temos $h(T^{A^*}) = h^*(T)$. Pelo Corolário 2.4, temos que qualquer sub-árvore máxima não trivial de T^{A^*} , com raiz em um nível não menor que $\lfloor 1.441 \log m + 1 \rfloor$, não tem hotleaves nem hotlinks partindo de nenhum de seus nós. Como nenhuma modificação nestas sub-árvores afeta a otimalidade de A^* , aplicamos o procedimento LOG Modificado em todas as sub-árvores máximas de T^{A^*} com raiz no nós no nível $\lfloor 1.441 \log m + 1 \rfloor$. Lembre que este algoritmo sempre constrói árvores melhoradas com altura não maior que $1.441 \log n + 1$. Assim, inserindo os hotlinks produzidos pelo LOG Modificado em A^* , temos que $H(T^{A^*}) \leq 1.441(\log m + \log n) + 2$, como ilustra a Figura 2.3.(b). \square

Observamos que o limite anterior foi melhorado por C. Bornstein para $1.441 \log m + \log n + 2.882 \log \log m + O(1)$ [?]¹.

2.2

Limites inferiores

Provaremos agora um limite inferior para $h^*(T)$. Observe que o grau de saída máximo em T^{A^*} é no máximo $d + 1$, onde d é o grau de saída máximo em T . Então, pelo menos uma das m hotleaves devem estar em um nível não menor que $\log_{d+1} m$ em T^{A^*} . Logo, temos o seguinte limite inferior para $h^*(T)$.

Lema 2.6 *Seja T uma árvore com m folhas e grau de saída máximo d . Então, $h^*(T) \geq \log_{d+1} m$.*

Este limite inferior pode ser muito folgado se d é próximo de m . Para obter um limite inferior, combinamos o limite inferior anterior com o fato de que podemos utilizar sub-árvores de T para fornecer limites inferiores para $h^*(T)$. Demonstramos esta idéia no Teorema a seguir.

Teorema 2.7 *Seja $T' = (V', E')$ uma sub-árvore binária de T com raiz em r . Então,*

$$h^*(T) \geq \log_3 |V' \cap L|.$$

Prova. Seja $T_2 = (V_2, E_2)$ uma árvore induzida pelos ancestrais de $V' \cap L$ em T . Aplicando o Lema 2.6 em T_2 , obtemos que $h^*(T_2) \geq \log_3 |V' \cap L|$. Logo,

¹que referência usar??

basta mostrar que $h^*(T) \geq h^*(T_2)$. Seja A^* uma atribuição de hotlinks P-ótima para T , e seja $A = \{(u, v) \in A^* \mid u, v \in V_2\}$. Para todo nó $i \in V' \cap L$, temos que o nível de i em T_2^A e T^{A^*} são os mesmos. Logo, temos que $h^*(T) \geq h(T_2^A) \geq h^*(T_2)$, o que completa a prova. \square

Por exemplo, se T é a árvore da Figura 3.1.(a), então o Lema 2.6 garante que $h^*(T) \geq \log_5 14 > 1.639$. Porém, se consideramos T' como a árvore induzida pelos nós sombreados na mesma figura, temos que $h^*(T) \geq \log_3 8 > 1.892$, pelo Lema 2.7.