Implementação da Remoção

```
/*
*  Atualização do FB e balanceamento para a raíz esquerda
*/
AVL *balanceamento_esquerdo(AVL *no, bool *h) {

    AVL *f_dir;
    int fb_dir;

    switch (no->fb) {
        case 1:
            no->fb = 0;
            break;
        case 0:
            no->fb = -1;
            *h = false;
            break;
        case -1:
                f_dir = no->dir;
                fb_dir = f_dir->fb;
                if (fb_dir <= 0) {
                    f_dir = rotacaoRR(no);
                    if (fb_dir == 0) {
                       no->fb = -1;
                       f_dir->fb = 1;
                       *h = false;
                    }
                    else {
                        no->fb = 0;
                        f_dir->fb = 0;
                    }
                    no = f_dir;
                }
                else {
                        no = rotacaoRL(no);
                        no->fb = 0;
                    }
     }
     return(no);
}

/*
*  Atualização do FB e balanceamento para a raiz direita
*/
AVL *balanceamento_direito(AVL *no, bool *h) {

    AVL *f_esq;
    int fb_esq;

    switch (no->fb) {
        case -1:
            no->fb = 0;
            break;
        case 0:
            no->fb = 1;
            *h = false;
            break;
        case 1:
            f_esq = no->esq;
            fb_esq = f_esq->fb;
            if (fb_esq >= 0) {
                f_esq = rotacaoLL(no);
                if (fb_esq == 0) {
                    no->fb = 1;
                    f_esq->fb = -1;
                    *h = false;
                }
                else {
                    no->fb = 0;
                    f_esq->fb = 0;
                }
                no = f_esq;
            }
            else {
                no = rotacaoLR(no);
                no->fb = 0;
            }
    }
    return(no);
}

/*
*  Busca nó substituto e realizada a remoção (busca o mais à direita do nó esquerdo
*/
AVL *busca_remove(AVL *no, AVL *no_chave, bool *h) {

    AVL *no_removido;
    if (no->dir != NULL) {
        no->dir = busca_remove(no->dir, no_chave, h);
        if (*h)
            no = balanceamento_direito(no, h);
    }
    else {
        no_chave->info = no->info;
        no_removido = no;
        no = no->esq;
        if (no != NULL)
            no->pai = no_removido->pai;
        *h = true;     //Deve propagar a atualização dos FB
        free(no_removido);
```

```
    }
    return(no);
}

/*
*  Remoção da Árvore AVL
*/
AVL *remove(AVL *raiz, int info, bool *h) {

    if (raiz == NULL) {
        printf("Chave não localizada !");
        *h = false;
    }
    else {
        if (raiz->info > info) {
            raiz->esq = remove(raiz->esq, info, h);
            if (*h)
                raiz = balanceamento_esquerdo(raiz, h);
        }
        else
            if (raiz->info < info) {
                raiz->dir = remove(raiz->dir, info, h);
                if (*h)
                    raiz = balanceamento_direito(raiz,h);
            }
            else { //Encontrou o elemento a ser removido
                if (raiz->dir == NULL) {
                    if (raiz->esq != NULL) //Escolhe o nó à esquerda como substituto
                        raiz->esq->pai = raiz->pai;
                    raiz = raiz->esq;
                    *h = true;
                }
                else
                    if (raiz->esq == NULL) {
                        if (raiz->dir != NULL) //Escolhe o nó à direita como substituto
                            raiz->dir->pai = raiz->pai;
                        raiz = raiz->dir;
                        *h = true;
                    }
                    else { // Busca o elemento mais à direita do nó esquerdo
                        raiz->esq = busca_remove(raiz->esq, raiz, h);
                        //Se necessário efetua balanceamento (Esquerdo pois a função
                        //busca_remove foi para o nó esquerdo)
                        if (*h)
                            raiz = balanceamento_esquerdo(raiz, h);
                    }
            }
    }
    return(raiz);
}
```

« Remoção em uma Árvore AVL