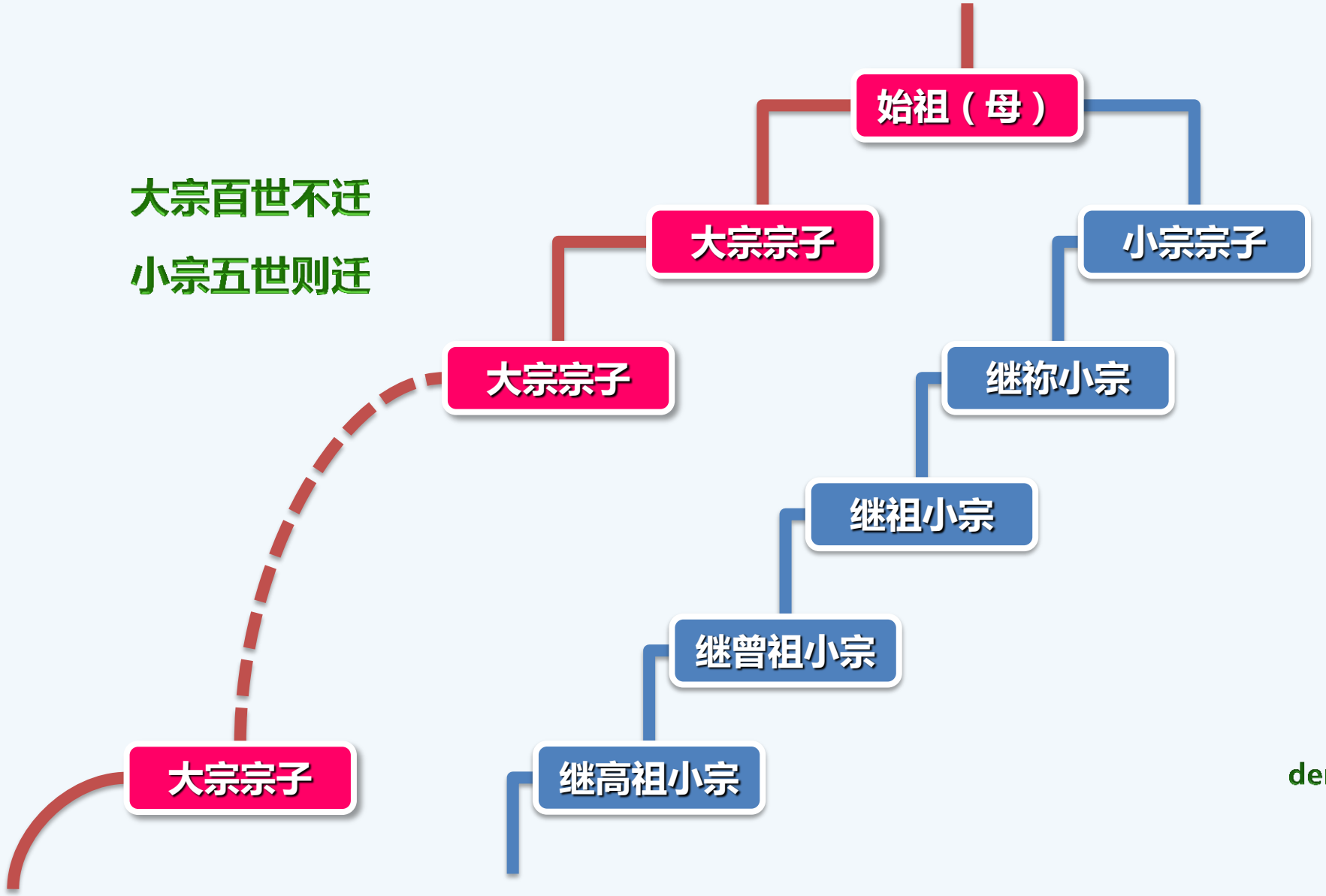

大宗百世不迁

小宗五世则迁

5.二叉树

后序遍历

迭代算法

邓俊辉

deng@tsinghua.edu.cn

gotoHLVFL()

```
❖template <typename T> static void gotoHLVFL( Stack <BinNodePosi(T)> & S ) {
    while ( BinNodePosi(T) x = S.top() ) //自顶而下反复检查栈顶节点
        if ( HasLChild( * x ) ) { //尽可能向左。在此之前
            if ( HasRChild( * x ) ) //若有右孩子，则
                S.push( x->rc ); //优先入栈
            S.push( x->lc ); //然后转向左孩子
        } else //实不得已
            S.push( x->rc ); //才转向右孩子
    S.pop(); //返回之前，弹出栈顶的空节点
}
```

travPost_I()

```
❖template <typename T, typename VST>
  void travPost_I( BinNodePosi(T) x, VST & visit ) {
     Stack < BinNodePosi(T) > S; //辅助栈
     if ( x ) S.push( x ); //根节点非空则首先入栈
     while ( ! S.empty() ) { //x为当前节点
        if ( S.top() != x->parent ) //栈顶非x之父（则必为其右兄）
           gotoHLVFL( S ); //在x的右子树中，找到HLVFL
        x = S.pop(); //弹出栈顶（即前一节点之后继）以更新x，并随即
        visit( x->data ); //访问之
     }
  }
```