(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 111241348 B
(45)授权公告日 2024.03.01

(21)申请号 201911410941.1

(22)申请日 2019.12.31

(65)同一申请的已公布的文献号
申请公布号 CN 111241348 A

(43)申请公布日 2020.06.05

(73)专利权人 浙江华云信息科技有限公司
地址 310012 浙江省杭州市西湖区西园一路16号2幢

(72)发明人 陈士云 张晓宁 叶兴会

(74)专利代理机构 浙江翔隆专利事务所(普通合伙) 33206
专利代理师 王晓燕

(51)Int.Cl.
G06F 16/901(2019.01)

(56)对比文件
CN 102413509 A,2012.04.11
CN 104915952 A,2015.09.16
CN 106059861 A,2016.10.26
US 2017068893 A1,2017.03.09
US 2017091241 A1,2017.03.30
US 6978271 B1,2005.12.20
孔涛;熊毅.故障树自动生成系统的树图布局算法研究.中国安全科学学报.2017,(第05期),全文.
陈大岳;章复熹.关于随机树上λ-随机游动速度的一个注记.中国科学:数学.2015,(第05期),全文.

审查员 蔡智勇

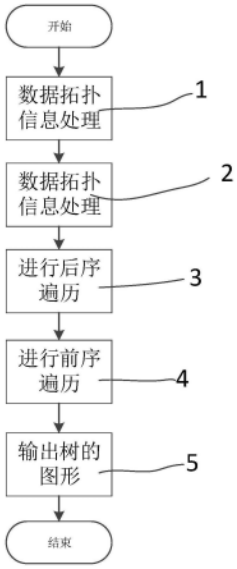
权利要求书1页 说明书3页 附图2页

(54)发明名称

一种基于Walker's Tree动态调整树间距的方法

(57)摘要

本发明公开了一种基于Walker's Tree动态调整树间距的方法,涉及结构树的自动成图领域。Walker's Tree算法的提出有效的解决了绘制树的宽度很宽和树中的父节点并不是其所有子节点的中心位置这两个问题,唯一缺憾的是没有考虑到动态调整树中的间距。本方法过程为:1)进行数据拓扑信息处理;2)建立树结构模型;3)对树模型中所有节点进行后序遍历,计算树中每个节点坐标的初始值和调整值;4)进行前序遍历,计算树中每个节点的最终坐标;5)输出树的图形。本方法不仅保证了父节点关于子节点的对称性,也保证了可以根据节点的大小动态调整树结构的间距,可以满足树中节点大小不一,树的结构不拥挤且整体整齐,实现完善的树图输出。



CN 111241348 B

1.一种基于Walker's Tree 动态调整树间距的绘制树方法,绘制树包括计算树的每个节点的x和y坐标,然后使用计算的坐标来绘制树;其特征在于,绘制树包括以下步骤:

1)首先进行数据拓扑信息处理,建立树节点对象,初始化树节点属性;

2)建立树结构模型,处理树中每个节点的父子关系;

3)对树模型中所有节点进行后序遍历,计算树中每个节点坐标的初始值和调整值;

4)进行前序遍历,计算树中每个节点的最终坐标,前序遍历之前,逐层遍历每层的树节点,判断该层所有节点的大小,根据最大的节点来设置上一层的层间距,所有层间距设置好之后再进行前序遍历设置每个节点的坐标,实现动态调整树结构的层间距,使输出的树图能够达到的树中节点大小不一,树的结构不拥挤且整体整齐美观的效果;

5)输出树的图形,得到树图;避免因Walker's Tree没有考虑到动态调整树图形绘制中的间距情况发生,使树图形绘制结构整体美观的同时树的层次结构不会很高很突兀的效果。

2.根据权利要求1所述的一种基于Walker's Tree 动态调整树间距的绘制树方法,其特征在于:步骤3)中,调整值针对以下情况而设:当前节点有左兄弟节点时,为了两个节点隔开一定的距离,需要考虑两者之间节点的大小以及节点间距;当前节点是一个子树的父节点时,要逐层的考虑其孩子节点是否与其他子树的节点有交叉,如果有交叉,则要拉开子树的距离。

3.根据权利要求1所述的一种基于Walker's Tree 动态调整树间距的绘制树方法,其特征在于:步骤3)中,后序遍历时,第一次遍历首先从最小的子树叶子结点开始,递归地从左向右去构造更大的子树,其中,确保兄弟节点之间有着一定的距离Sibling Separation,相邻的子树之间也有一定的距离Sibling Separation,当树的遍历由叶子结点到根节点时,它将较小的子树和其根节点结合在一起形成了一个更大的子树,对一个固定的节点,其子树是从左向右一个一个的移动,定位,当其中一个新的子树已经设计好,将其新子树放在邻居的左边,一直移动到新子树与其他节点没有交叉,移动的过程先开始的是它们子树的根节点,由Silbling Separation定义的距离隔离开来,再下一层,它们由Subtree Separation定义的距离分开,这样的依次往下一层继续,直到走到子树的根节点为止,当针对该节点的左右子孙节点完成后,该节点就处于其左边所有的子节点和右边所有的子节点的中间位置。

4.根据权利要求1所述的一种基于Walker's Tree 动态调整树间距的绘制树方法,其特征在于:步骤4)中,由树的根节点开始,将其横坐标的值与其所有祖先节点的调整值加一起,即为节点的坐标值。

一种基于Walker's Tree动态调整树间距的方法

技术领域

[0001] 本发明涉及结构树的自动成图领域,尤其涉及一种基于Walker's Tree 动态调整树间距的方法。

背景技术

[0002] 绘制树的关键任务是决定树中每个节点的位置。绘制树算法主要是计算树的每个节点的x和y坐标。然后可以使用这些坐标来绘制树。其中节点定位算法必须解决两个关键问题。首先,绘制树的整体美观。其次,节点定位算法应节省空间。这两个问题中的每一个都可以直接处理,但将它们结合在一起会带来一些挑战。

[0003] 许多研究者提出的绘制树的算法也存在绘制出树的结果不尽如人意的情况,主要有三种问题需要解决,分别是1)绘制树的宽度很宽;2)树中的父节点并不是其所有子节点的中心位置;3)树中每层间距都是固定的,是一个固定常数,这种设置并没有考虑上下两层节点大小与间距之间产生的影响。

[0004] Walker's Tree算法的提出有效的解决了前两个问题,唯一缺憾的是没有考虑到动态调整树中的间距,使树结构整体美观的同时树的层次结构不会很高很突兀的效果。

发明内容

[0005] 本发明要解决的技术问题和提出的技术任务是对现有技术方案进行完善与改进,提供一种基于Walker's Tree 动态调整树间距的方法,以动态调整树中的间距,实现完善的树图为目的。为此,本发明采取以下技术方案。

[0006] 一种基于Walker's Tree 动态调整树间距的方法,包括以下步骤:

[0007] 1)首先进行数据拓扑信息处理,建立树节点对象,初始化树节点属性;

[0008] 2)建立树结构模型,处理树中每个节点的父子关系;

[0009] 3)对树模型中所有节点进行后序遍历,计算树中每个节点坐标的初始值和调整值;

[0010] 4)进行前序遍历,计算树中每个节点的最终坐标;

[0011] 5)输出树的图形。

[0012] 本方法不仅保证了父节点关于子节点的对称性,也保证了可以根据节点的大小动态调整树结构的间距,可以满足树中节点大小不一,树的结构不拥挤且整体整齐,实现完善的树图输出。

[0013] 作为优选技术手段:步骤3)中,调整值针对以下情况而设:当前节点有左兄弟节点时,为了两个节点隔开一定的距离,需要考虑两者之间节点的大小以及节点间距;当前节点是一个子树的父节点时,要逐层的考虑其孩子节点是否与其他子树的节点有交叉,如果有交叉,则要拉开子树的距离。

[0014] 作为优选技术手段:步骤3)中,后序遍历时,第一次遍历首先从最小的子树叶子结点开始,递归地从左向右去构造更大的子树,其中,确保兄弟节点之间有着一定的距离

Sibling Separation,相邻的子树之间也有一定的距离Sibling Separation,当树的遍历由叶子结点到根节点时,它将较小的子树和其根节点结合在一起形成了一个更大的子树,对一个固定的节点,其子树是从左向右一个一个的移动,定位,当其中一个新的子树已经设计好,将其新子树放在邻居的左边,一直移动到新子树与其他节点没有交叉,移动的过程先开始的是它们子树的根节点,由Silbling Separation定义的距离隔离开来,再下一层,它们由Subtree Separation定义的距离分开,这样的依次往下一层继续,直到走到子树的根节点为止,当针对该节点的左右子孙节点完成后,该节点就处于其左边所有的子节点和右边所有的子节点的中间位置。有效实现后序遍历,考虑子树与已布局是否有交叉处理和根据相邻节点大小计算相邻节点之间间距。

[0015] 作为优选技术手段:步骤4)中,前序遍历之前,需要逐层遍历每层的树节点,判断该层所有节点的大小,根据最大的节点来设置上一层的层间距,所有层间距设置好之后再进行前序遍历设置每个节点的坐标。该过程由树的根节点开始,将其横坐标的值与其所有祖先节点的调整值加一起,即为节点的坐标值。有效实现前序遍历,实现根据层节点的大小调整每层的层间距。

[0016] 有益效果:本方法不仅保证了父节点关于子节点的对称性,也保证了可以根据节点的大小动态调整树结构的间距,可以满足树中节点大小不一,树的结构不拥挤且整体整齐,实现完善的树图输出。

附图说明

[0017] 图1是本发明流程示意图。

[0018] 图2是本发明后序遍历和前序遍历的树示例图。

具体实施方式

[0019] 以下结合说明书附图对本发明的技术方案做进一步的详细说明。

[0020] 如图1所示,一种基于Walker's Tree 动态调整树间距的方法,其过程包括以下步骤:

[0021] 1)首先进行数据拓扑信息处理,建立树节点对象,初始化树节点属性;

[0022] 2)建立树结构模型,处理树中每个节点的父子关系,本实例中的树均为只有一个根节点的树;

[0023] 3)进行后序遍历,计算树中每个节点坐标的初始值和调整值,调整值针对以下情况而设:当前节点有左兄弟节点时,为了两个节点隔开一定的距离,需要考虑两者之间节点的大小以及节点间距;当前节点是一个子树的父节点时,要逐层的考虑其孩子节点是否与其他子树的节点有交叉,如果有交叉,则要拉开子树的距离;

[0024] 4)进行前序遍历,计算树中每个节点的最终坐标,树的方向是向下的,所以树节点的纵坐标主要是依靠其所在的层数和上下两层节点大小而决定的;

[0025] 5)输出树的图形。

[0026] 后序遍历和前序遍历体现了动态调整树间距的过程。后序遍历的过程不仅决定了每个节点坐标的初始值和调整值,也会根据相邻节点调整两个节点之间的距离。前序遍历主要是根据每层节点的大小动态设置每层的层间距,再决定每个节点坐标值。

[0027] 后序遍历时,第一次遍历首先从最小的子树叶子结点开始,递归地从左向右去构造更大的子树,其中,确保兄弟节点之间有着一定的距离Sibling Separation,相邻的子树之间也有一定的距离Sibling Separation,当树的遍历由叶子结点到根节点时,它将较小的子树和其根节点结合在一起形成了一个更大的子树,对一个固定的节点,其子树是从左向右一个一个的移动,定位,当其中一个新的子树已经设计好,将其新子树放在邻居的左边,一直移动到新子树与其他节点没有交叉,移动的过程先开始的是它们子树的根节点,由Silbling Separation定义的距离隔离开来,再下一层,它们由Subtree Separation定义的距离分开,这样的依次往下一层继续,直到走到子树的根节点为止,当针对该节点的左右子孙节点完成后,该节点就处于其左边所有的子节点和右边所有的子节点的中间位置。

[0028] 以下为后序遍历和前序遍历示例。

[0029] 如图2所示,其后序遍历序列为:EIPHBCKMNQRSGDA。由该序列可知,需要先访问的节点是左边的叶子节点E。由于节点E左边没有兄弟节点,所以直接设置E节点的prelimnary Value为0,modeifierValue的初始值也设置为0。下一个处理的节点为I节点,I节点也是叶子节点,没有左兄弟节点,所以节点I的preliminaryValue设置为0,modifierValue设置为0。再者处理是叶子节点P。P节点是叶子节点,有左兄弟节点I,所以P.prelimnaryValue = I.prelimnaryValue + SiblingSeparation + TreeNodeSize = 0 + 4 + 2 = 6。当该处理节点B时,其leftMostNode = E (最左边的子节点),rightMostNode = E(最右边的子节点),H节点没有左兄弟节点,所以B.prelimnaryValue = (EprelimnaryValue + E.prelimnaryValue) / 2;接下来连续去处理节点B(同节点E的处理方法)。节点F也是叶子节点所以同I节点的处理方法。D节点和B节点已经是一个子树了。所以需要逐层判断下层的节点是否会重叠。如果有交叉和重叠,则对D子树移动相应的距离避免交叉和重叠。

[0030] 前序遍历之前,需要逐层遍历每层的树节点,判断该层所有节点的大小,根据最大的节点来设置上一层的层间距,所有层间距设置好之后再进行前序遍历设置每个节点的坐标。该过程由树的根节点开始,将其横坐标的值与其所有祖先节点的调整值加一起,即为节点的坐标值。

[0031] 如图2所示,其前序遍历序列为:ABEHIPCDKGMNQRS。图中每个节点的横坐标是将其prelimnaryValue + All ancestor's modifierValue。所以可依据第一个后序遍历的过程计算图中每个节点的横坐标。例如:根节点A : 13.5,B:B.prelimnaryValue + A.modifierValue = 3 + 0 = 3;.....最后可以计算出树中每个节点的坐标值。

[0032] 本发明主要是基于Walker's Tree算法的基础之上提出的改进算法——动态调整树结构的层间距。这样得出的树可以满足树中节点大小不一,树的结构不拥挤且整体整齐美观的效果。

[0033] 以上图1-2所示的一种基于Walker's Tree 动态调整树间距的方法是本发明的具体实施例,已经体现出本发明突出的实质性特点和显著进步,可根据实际的使用需要,在本发明的启示下,对其进行形状、结构等方面的等同修改,均在本方案的保护范围之列。

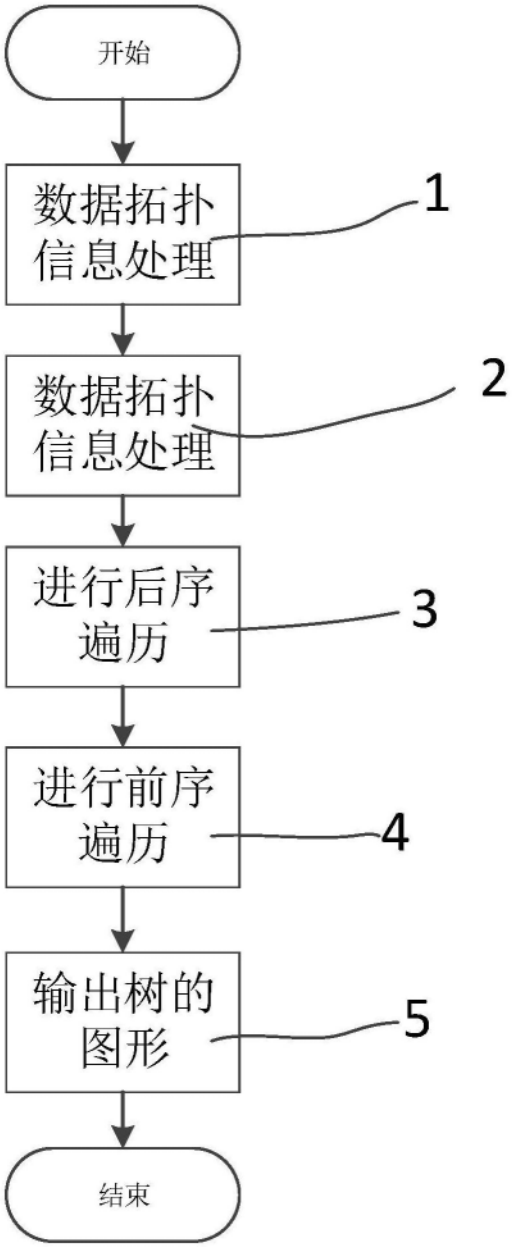


图1

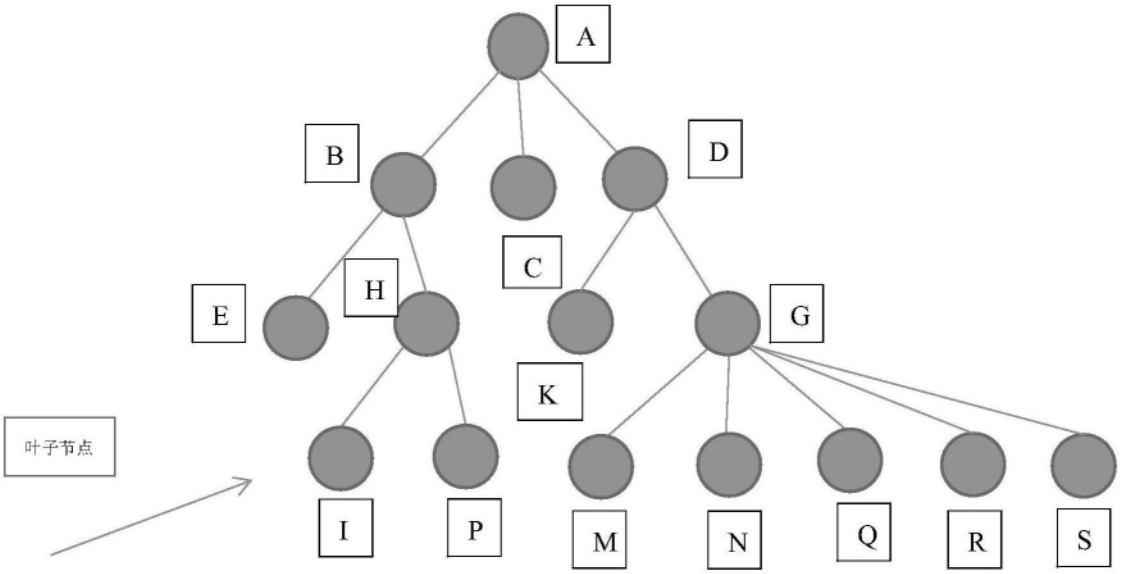


图2