Dominator Tree

胥拿云 张哲恺 冼臧越洋 龙思杉
Made By 龙思杉

April 5, 2016

Contents

1 功能介绍

一些定义：

1.1 Flow Graph

定义为一个有向图 $G = (V, E, r)$, r为起点， 从r出发可以到达其他所有点。

1.2 dominator

必经点， 若在G=(V, E, r)中，从r到y的路径一定经过x，则称x为y的dominator，记为x $dom\ y$， $dom(y)$表示y的所有必经点的集合。

1.3 immediate dominator

最近必经点，简记为$idom$，$idom(y)$为y的所有dominator中离y最近的点，这个点是唯一的。

1.4 dominator tree

对于一个Flow Graph$G$=(V, E, r)它的一个子图 $D = (V, (idom(i), i|i \in V$，$i \neq r)$，$r$）.

1.5 Lenguer-Tarjan

dominance问题产生于global flow analysis，为了解决这个问题，计算机科学家提出了许多算法， 我们学习并实现了这些算法中最为先进的Lenguer-Tarjan算法。Lenguer-Tarjan算法可以在$O((n + m)\alpha(n))$的时间复杂度内，求出一个Flow Graph的dominator tree.

2 原理说明

一些定义：

2.1 Dfs Tree

depth-first search tree 深度优先搜索树， 对于一个图选定一个起点，对其进行深度优先搜索， 构成的搜索树即为深度优先搜索树。

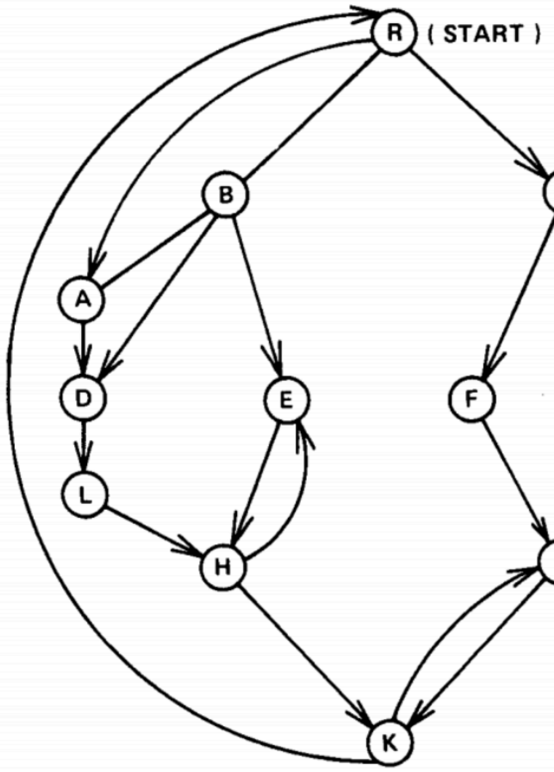


Fig. 1. A flowgraph

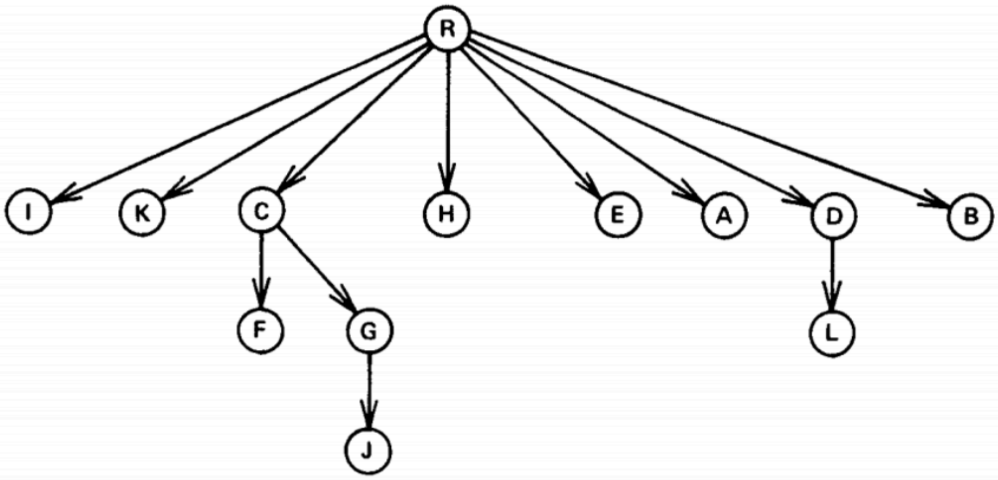


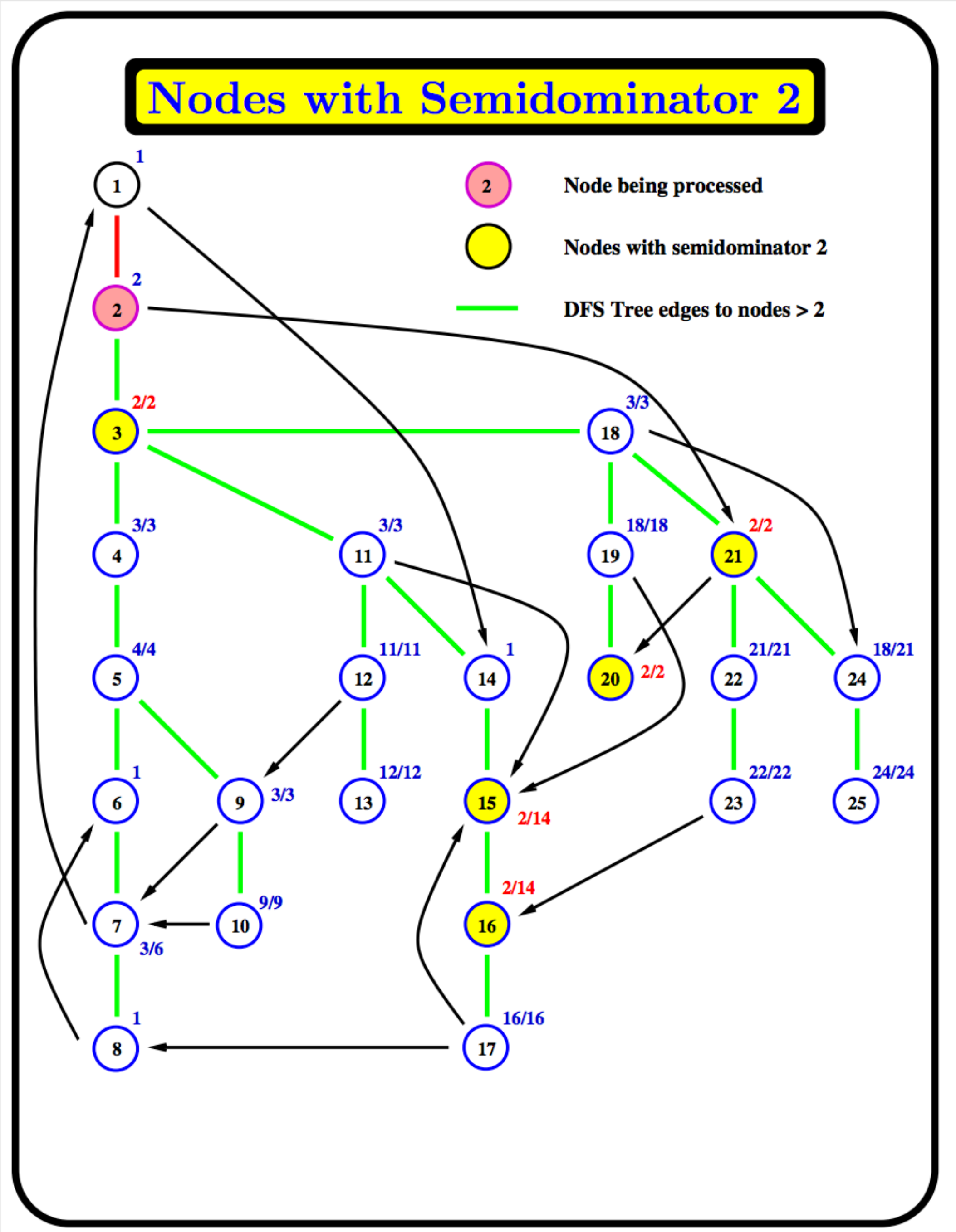
Fig. 2. Dominator tree of flowgraph in Fig. 1

2.2 dfn

depth-first number, 表示dfs tree中的时间戳。$dfn(x)$表示点x在深度优先搜索中第一次被访问的时间编号。

2.3 semidominator

半必经点,记为$semi$. $semi(w) = min\{v|$ 存在一条路径$v \to v_0, v_1, ..., v_k \to w, dfn(v_i) > dfn(v_k)(1 \leqslant i \leqslant k-1)\}$.



2.4 两个定理

2.4.1 半必经点定理

$\forall w \neq r \quad semi(w) = min(\{v|(v,w) \in E$ 且 $dfn(v) < dfn(w)\} \bigcup \{semi(u)ldfn(u) > dfn(w)$ 且 $\exists(v,w)$ 使得 u是v的祖先 $\}$

2.4.2 必经点定理

设$y = id[mindfn[semi(z)]|z \in path]$,即$path$中半必经节点的时间戳最小的节点。

$$idom(x) = \begin{cases} semi(x) & semi(x) = semi(y) \\ idom(y) & semi(x) \neq semi(y) \end{cases}$$

2.5 大致原理

首先使用半必经点定理，按dfn倒序求出每个点的半必经点.
然后使用必经点定理确定每个点的必经点，从而构建出dominator tree.

3 正确性分析

3.1 必经点定理

Case 1：$semi(x) = semi(y)$

使用反证法，假设$semi(x)\neq idom(x)$，由于DFS树上$semi(x)$之下的点一定不是x的必经点，所以$idom(x)$在$semi(x)$之上，由假设，$semi(x)$不是x的必经点，即存在一条从r到x的路径不经过$semi(x)$，考虑这条路径与$semi(x)$到x这条链上的第一个交点z，显然$semi(z) < semi(x)$，而$semi(z)\geqslant semi(y) = semi(x)$，矛盾！

Case 2：$semi(x)\neq semi(y)$

显然$semi(y) < semi(x)$（链上$semi(x)$的儿子结点的$semi$值最大为$semi(x)$，故$semi(y)\leqslant semi(x)$，由$semi(x)\neq semi(y)$，即$semi(y) < semi(x)$.

使用反证法，$idom(x)\neq idom(y)$ 若$idom(x) > idom(y)$，由于存在路径$idom(y) \rightarrow y$不经过$idom(x)$（因为$idom(x)$不是y的必经点）,即存在路径$r \rightarrow dom(y) \rightarrow y \rightarrow x$
不经过$idom(x)$，矛盾。

若$idom(x) < idom(y)$，考虑$idom(x)$到x的路径（不经过$idom(y)$，由于$idom(y)$不是x的必经点，
所以这样的路径存在）与$idom(y) \rightarrow x$这条链的第一个交点z

若$z\leqslant y$，则存在路径$r \rightarrow idom(x) \rightarrow y$不经过$idom(y)$，矛盾；

若$z > y$，考虑$idom(x) \rightarrow x$的路径上z的父亲u，由于z是$idom(y) \rightarrow x$的第一个交点，所以u在$idom(x) \rightarrow idom(y)$上，显然$u$是$z$的一个半必经结点,所以$semi(z)\leqslant u < idom(y)$ $\leqslant semi(y)$，与$semi(y)$的最小性矛盾。

所以$idom(x)$＝$idom(y).$

3.2 半必经点定理

我们考虑哪些点可能成为点x的半必经点
考虑半必经点路径$p(u \rightarrow x)$中最后一条边(y, x)
若$dfn(y) < dfn(x)$，由于p中除端点外的点z均有$dfn(z) > dfn(x)$，所以y只能为端点。
即x的半必经点
若$dfn(y) > dfn(x)$,考虑$u- > y$这条链与p的除u外的第一个交点v，由p是半必经点的
路径,有$dfn(v) > dfn(x)$,所以v一定是y及其祖先中时间戳值小于x的点，又因为v是这条
链的第一个交点,即有u是v的半必经点。反之y及其祖先中时间戳值小于x的点的半必经点

（设半必经点为q,对应的祖先为t），$q->t$的半必经路径+$t->y$（走树边）+$y->x$，显然是一个半必经路径,，即q为x的半必经结点。所以u必为y及其祖先中时间戳值小于x的点的半必经点中的一个。
综上，证毕。

4 时空复杂度证明

tarjan函数中对每个点，每条边，以及dom数组中的每个元素进行操作，dom数组总共有n个元素，同时利用并查集进行维护，时间复杂度为$O((n+m)\alpha(n))$ 代码实现中储存了n个点的信息，每条边被两个点储存一次，所有点的dom数组总共执行了n次添加操作，空间复杂度显然为$O(m+n)$

5 具体实现方式

```
procedure dfs(x is a vertex)
{
    dfn[x]←t←t+1, id[t]←x
    for each y in suc[x] do
        if dfn[y]==0 then
            dfs(y)
}
function get(x is a vertex):a vertex
{
    if x=anc[x] then return x
        y←get(anc[x])
        if dfn[semi[best[x]]]>dfn[semi[best[anc[x]]]] then
        best[x]←best[anc[x]]
        anc[x]←y, return y
}
procedure tarjan
{
        for i←t step -1 until 1 do
        {
            y←id[i]
            x←fa[y]
            for each z in pre[id[y]] do
            {
                if dfn[z]=0 then continue
                    get(z)
                    semi[y]←id[min(dfn[semi[y]],dfn[semi[best[z]]])]
            }
            push y into the back of dom[semi[y]]
            anc[y]←x
            for each z in dom[x] do
            {
                get(z)
                if dfn[semi[best[z]]]<dfn[x] then idom[z]←best[z]
                    else idom[z]←x
            }
            empty dom[x]
        }
        for i←2 step 1 until t do
        {
            x=id[i]
            if idom[x]≠semi[x] then idom[x]←idom[idom[x]]
                push x into the back of dom[idom[x]]
        }
        idom[1]←0
}
```

6 具体实现方式细节

$best$，$semi$，$father$数组的初值设定:

$best[i] = i, semi[i] = i, father[i] = i$

dom数组的意义:

在$tarjan()$的前半部分，即倒序扫描部分中，$dom[x]$存储的是所有$semi[y] = x$的y。在$tarjan()$后半部分，即正序扫描部分$dom[x]$存储的是x dom的所有点

$idom$数组的意义:

在$tarjan()$的前半部分，即倒序扫描部分中，$idom[x]$存储的是为使用必经点定理确定必经点的辅助变量，有两种情况，一种是x到$semi[x]$路径上$dfn[semi[y]]$最小的y，一种是$semi[x]$。联系必经点定理，这样的写法十分方便。