# 时态图中的可达性和基于时间的路径查

# 总结

**一、应用背景**

本文主要解决 时态图（Temporal Graphs）中的可达性查询，时态图与普通有向图的差别在于，时态图中顶点之间的边有两个权重，一个是开始时间，一个是持续时间。另外，一对顶点之间可以存在多条边。



给定一个查询Q（a , d , [ 1, 8]），我们可以回答：

①a→d 是否在时间区间[1,8]内可达；

②若a→d 在时间区间[1,8]内可达，我们可以找到a能最早到达d的路径；

③若a→d 在时间区间[1,8]内可达，我们可以找到a到达d中总的持续时间最小的路径；

1. **解决思路**

将时态图这一种复杂的有向有环图转换成一个有向无环图。

作者的方法

①依据每个顶点中严格的时间顺序，将顶点v分为Vin和Vout。

②每个顶点v内的信息为（v , t ）,v是顶点名称，t是该顶点的开始时间或者是到达时间，并将所有顶点按照t的大小从上到下进行升序排序。

③根据顶点名称将具有相同顶点名称的顶点划分为同一个链中；

④根据三种规则将所有的顶点按顺序连接起来，就可以将将时态图这一种复杂的有向有环图转换成一个有向无环图（DAG）。

⑤根据每个顶点所在的链中的位置，给每个顶点建立索引：Lout(V)和Lin(V)建立Lout(V)时只用扫描一遍DAG拓扑排序的逆序来建立所有顶点的Lout(V)索引，建立Lin(V)时正向扫描一遍DAG拓扑排序来建立所有顶点的Lin(V)索引。

⑥建立chain索引时，根据问题规模适当保留前K个索引信息，建立Lout(V)时不同链优先保留链值最小的，同链的时候保留所在链中位置最小的。Lin(V)相反。

⑦最后根据索引和查询规则可以快速判断a→d 是否在给定的时间区间内可达；

*L*out(*v12*) =code(*v12*)= {(4,2)}

*L*out(*v4*) = {(1,4)}

*L*out(*v10*) = code(*v10*)U*L*out(*v4*) ={(1,4),(3,3)}

*L*out(*v9*) = top2(code(*v9*)U*L*out(*v10*)U*L*out(*v12*))

=top2{(3,2) U(1,4),(3,3)U(4,2)} ={(1,4),(3,2)}

*L*out(*v8*) =top2(code(*v8*)U*L*out(*v9*))={(1,4),(3,1)}

*L*out(*v3*) =top2(code(*v3*)U*L*out(*v8*))={(1,3),(3,1)}

*Lin*(*v1*) = {(1,1)}

*Lin(v2) = code(v2) ULin(v1)=*{(1,1)U(1,2)}={(1,2)}

*Lin(v3) = code(v3) U Lin(v2)=* {(1,3)U(1,2)}={(1,3)}

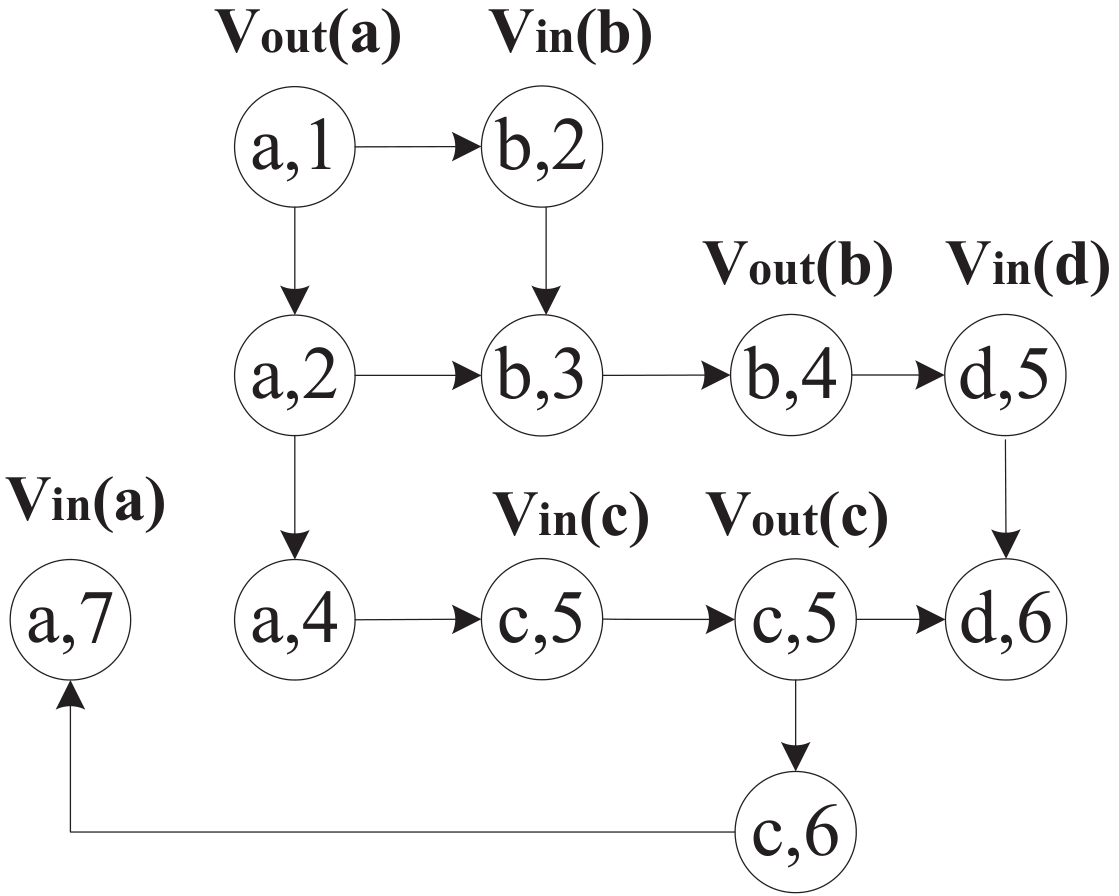
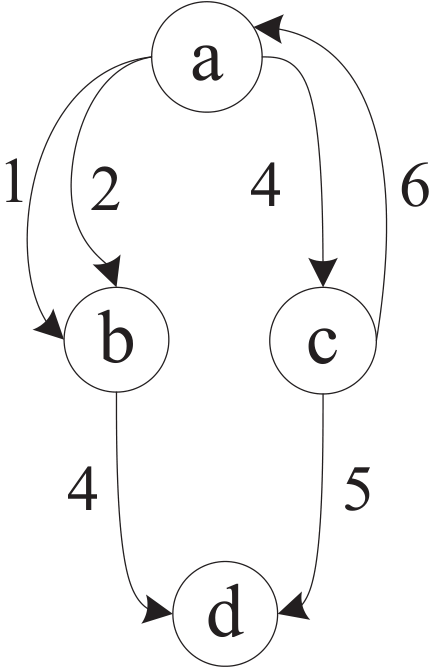
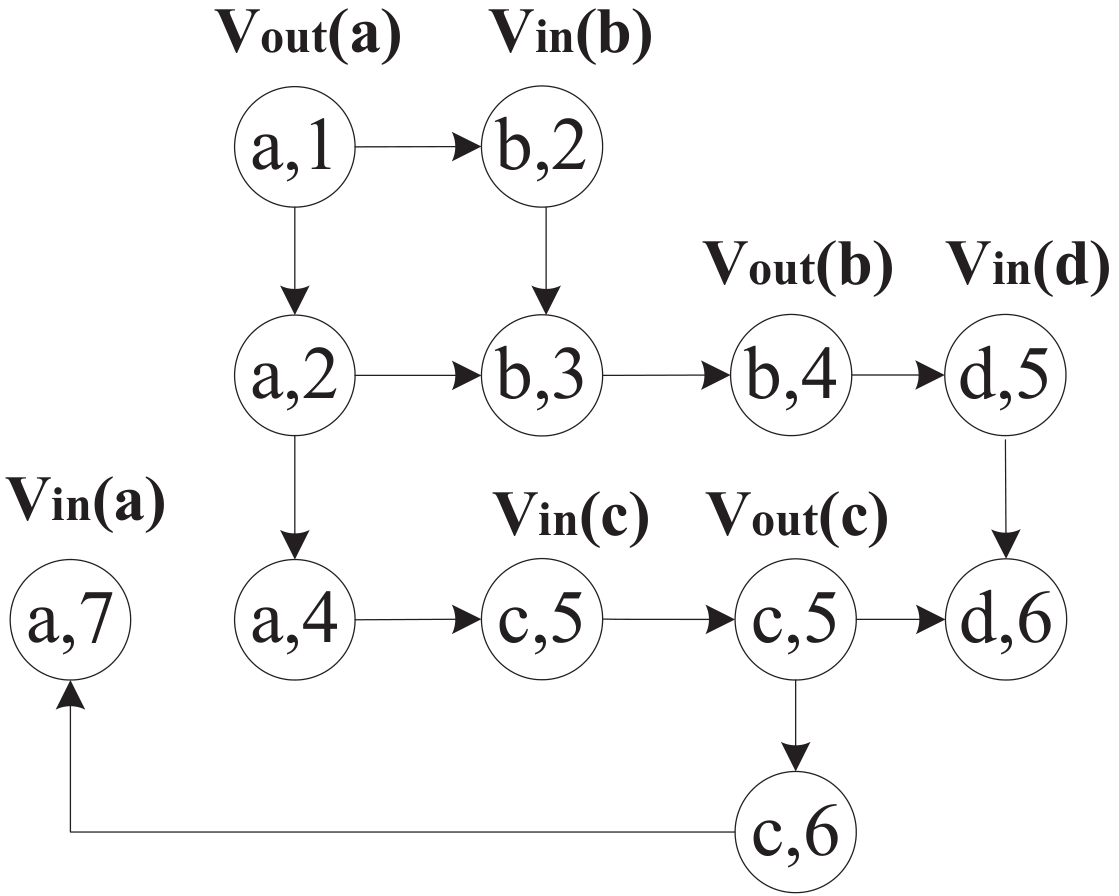
*Lin(v8) =code(v8) U Lin(v3)*={(1,3),(3,1)}

*Lin(v9) =code(v9) U Lin(v8)*

={(1,3),(3,1)*U*(3,2)}={(1,3),(3,2)}

Lin(v12) =*code(v12)* U *Lin(v9)*

=top2{(1,3),(3,2)U(4,1)}={(1,3),(3,2)}



**Chain1**

**Chain2**

**Chain3**

**Chain4**