

Attribute Expansion Graph

表示形式

$$(V, E, V_A, V_V, V_C, E_A, E_V, E_C)$$

V :原图中的点集

E :原图中的边集

V_A :Attribute点

V_V :Value点

V_C :Constant点

E_A : $E_A \in V * V_A$ ，其中 (u, v) 表示一条从 u 到 v 的有向边

E_V : $E_V \in V_A * V_V$ ，其中 (u, v) 表示一条从 u 到 v 的有向边

E_C : $E_C \in V_V * V_C$ ，其中 (u, v) 表示一条从 u 到 v 的有向边

限制条件

- 1.若 $(u, v) \in E_C$,则不存在 $u' \neq u$ 使得 $(u', v) \in E_C$
- 2.若 $(u, v) \in E_C$,则不存在 $v' \neq v$ 使得 $(u, v') \in E_C$
- 3.若 $(u, v) \in E_V$,则不存在 $v' \neq v$ 使得 $(u, v') \in E_C$
- 4.若 $(u, v) \in E_A$,则不存在 $u' \neq u$ 使得 $(u', v) \in E_C$
- 5.对于 $v \in V_A$,一定有 $indegree(v) = 1, outdegree(v) = 1$
- 6.对于 $v \in V_V$,一定有 $indegree(v) \geq 1, outdegree(v) = 1$
- 7.对于 $v \in V_C$,一定有 $indegree(v) = 1, outdegree(v) = 0$

说明

- 1.value点与constant点一一对应，有value点一定就要有对应的constant点
- 2.一个attribute点只能连向一个value点，一个value点可与多个Attribute点一一对应
- 3.一个实体点可与多个Attribute点对应，一个Attribute点只能对应一个实体点

Attribute Expansion Graph Pattern

表示形式

$$(V, E, V_A, V_V, V_C, E_A, E_V, E_C)$$

V :原图中的点集

E :原图中的边集

V_A :Attribute点

V_V :Value点

V_C :Constant点

E_A : $E_A \in V * V_A$ ，其中 (u, v) 表示一条从 u 到 v 的有向边

E_V : $E_V \in V_A * V_V$ ，其中 (u, v) 表示一条从 u 到 v 的有向边

E_C : $E_C \in V_V * V_C$ ，其中 (u, v) 表示一条从 u 到 v 的有向边

限制条件

- 1.若 $(u, v) \in E_C$,则不存在 $u' \neq u$ 使得 $(u', v) \in E_C$
- 2.若 $(u, v) \in E_C$,则不存在 $v' \neq v$ 使得 $(u, v') \in E_C$
- 3.若 $(u, v) \in E_V$,则不存在 $v' \neq v$ 使得 $(u, v') \in E_C$
- 4.若 $(u, v) \in E_A$,则不存在 $u' \neq u$ 使得 $(u', v) \in E_C$
- 5.对于 $v \in V_A$,一定有 $indegree(v) = 1, outdegree(v) = 1$
- 6.对于 $v \in V_V$,一定有 $indegree(v) \geq 1, outdegree(v) \leq 1, degree(v) \geq 2$
- 7.对于 $v \in V_C$,一定有 $indegree(v) = 1, outdegree(v) = 0$

说明

- 1.value点与constant点一一对应,若value点没有对应的constant点，需要有多多个attribute点对应该value点
- 2.一个attribute点只能连向一个value点，一个value点可与多个Attribute点一一对应
- 3.一个实体点可与多个Attribute点对应，一个Attribute点只能对应一个实体点

性质

每一个Attribute Expansion Graph Pattern均可对应一个Graph Pattern+Literal Set，且每一个Graph Pattern+Literal Set的结合体均可表示为一个Attribute Expansion Graph Pattern。其中LiteralSet支持的语义有：

- 1.x.A
- 2.x.A=c
- 3.x.A=y.B

证明

先证明：每一个Attribute Expansion Graph Pattern均可对应一个Graph Pattern+Literal Set。

若 $V_A = V_V = V_C = \emptyset$ ，则对应Graph Pattern+ \emptyset

若 $V_A \neq \emptyset, V_V = V_C = \emptyset$ ，则对应Graph Pattern+ $x.A$,其中x为Attribute点对应的实体点

若 $V_A \neq \emptyset, V_V \neq \emptyset, V_C = \emptyset$ ，此时的value点一定有多多个Attribute点对应，此时可以对应的结构

有：Graph Pattern+ $x.A = y.B$,Graph Pattern+ $x.A, x.A = y.B$

若 $V_A \neq \emptyset$, $V_V \neq \emptyset$, $V_C \neq \emptyset$, 此时的value点要么有多个Attribute点对应, 要么有一个constant点对应。此时可以对应的结构有: Graph Pattern+ $x.A = y.B$, $x.A = c$, Graph Pattern+ $x.A$, $x.A = y.B$, $x.A = c$, Graph Pattern+ $x.A$, $x.A = c$, Graph Pattern+ $x.A = c$
此时, Graph Pattern+Literal Set的全部情况均对应上

再证明: 对于任意的一种raph Pattern+Literal Set,都可以转化为一个Attribute Expansion Graph Pattern

对于x.A, 在x节点处拓展一个Attribute节点即可

对于x.A=y.B, 在x节点拓展一个Attribute节点A,在y节点处拓展一个Attribute B, 将A,B节点与同一个value节点相连

对于x.A=c,在x节点拓展一个Attribute节点A, 在A节点拓展一个value节点V,在V节点拓展一个constant节点C即可

挖掘算法

基于传统方法的挖掘算法(一条一条加边)做一定变动, 即将加边操作变为实体拓展或属性拓展, 挖掘的是在一定的操作步数内能够得到的Attribute Expansion Graph Pattern

实体拓展: 正常加边

属性拓展: 分为以下几种情况

- 1.在某个实体节点拓展一个A节点
- 2.选择 ≥ 2 个Attribute节点, 新建一个Value节点且让选择的Attribute节点与该节点连边
- 3.选择一个Value节点, 拓展一个constant节点