

# Mining for Tree-Query Associations in a Graph

Bart Goethals<sup>1</sup>, Eveline Hoekx<sup>2</sup>, Jan Van den Bussche<sup>2</sup>

<sup>1</sup>University of Antwerp, Belgium

<sup>2</sup>University of Hasselt, Belgium

## Graph Data

A (directed) **graph** over a set of nodes  $N$  is a set  $G$  of edges: ordered pairs  $(i,j)$  with  $i,j$  in  $N$ .

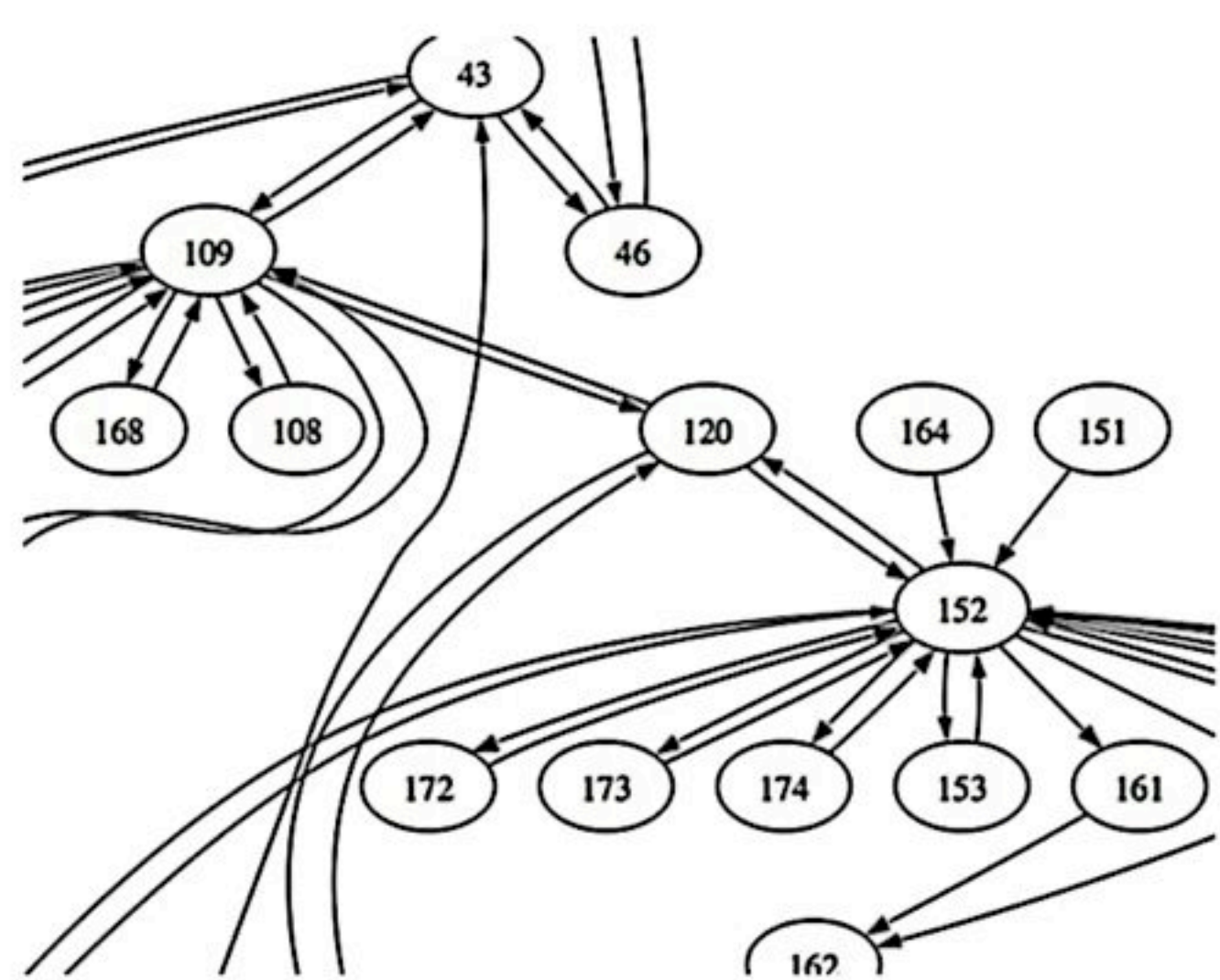


Figure 1: Snapshot of a graph representing the complete metabolic pathway of a human. A node is a compound or enzyme and an edge a reaction.

## Graph Mining

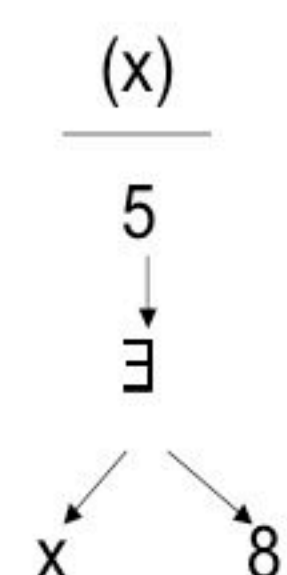
➔ Focus on pattern mining, few work on association rule mining!

	Transactional category	Single graph category
dataset	set of many small graphs (transactions)	single large graph
frequency	#graphs in which the pattern occurs	#copies of the pattern in the large graph
examples	Warmr, gSpan, AGM, FSG, FFSM	Subdue, SEus, SiGraM, Jeh-Widom

## Our work

- Single graph category
- Pattern + association rule mining
- **Tree Queries:** tree-shaped patterns inspired by conjunctive database queries

### Parameters and Existential nodes



```

select distinct G3.to as x
from G G1, G G2, G G3
where G1.from=5 and
G1.to=G2.from and
G1.to=G3.from and G2.to=8
    
```

frequency:

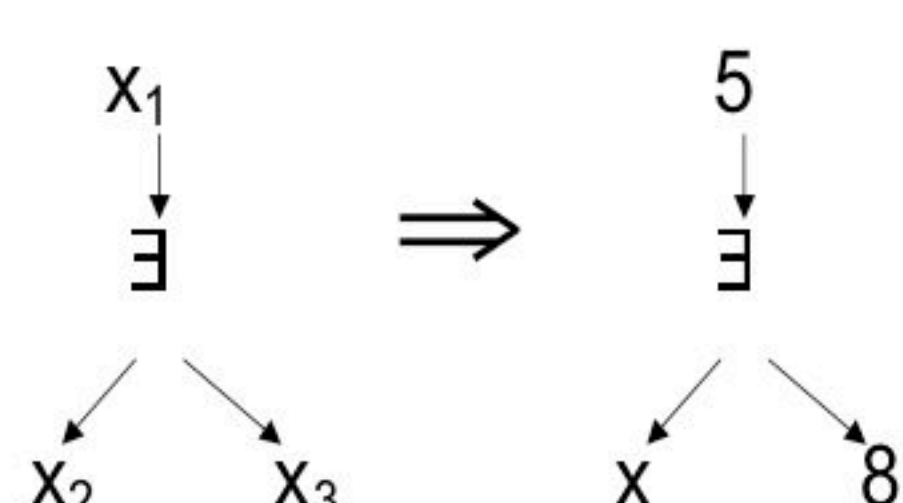
$$\#\{x \mid \exists z: (5,z) \in G \wedge (z,8) \in G \wedge (z,x) \in G\}$$

- An **occurrence** of the pattern in  $G$  is any **homomorphism** from the pattern in  $G$ .

### Tree-Query Association:

$(X_1, X_2, X_3)$

$(5, X, 8)$



$$\text{Conf} = \frac{\text{freq(lhs)}}{\text{freq(rhs)}}$$

## Features of our algorithm

1. Pattern mining phase and association rule mining phase
1. Restriction to trees  $\Rightarrow$  efficient algorithms
2. Equivalence checking
3. Apply theory of conjunctive database queries
4. Database oriented implementation

## Problem Definition

- **Input:** a graph  $G$ , threshold  $minsup$ , a tree query  $Q_{left}$  frequent in  $G$  and an threshold  $minconf$
- **Output:** all tree queries  $Q$ , such that  $Q_{left} \Rightarrow Q$  is frequent and confident in  $G$ .

## Algorithm

### 1. Pattern mining

#### Outer loop:

Generate, incrementally, all possible trees of increasing sizes. Avoid generation of isomorphic trees.

