**Graph Pattern Mining**

- **Unlabeled graphs** vs. **Labeled graphs**
  - $G = (V, E)$
    - $L(u)$ = label of vertex $u$
    - $L(u, v)$ = label of an edge $(u, v)$

**Task:** find commonly occurring subgraphs

- Frequent subgraphs
- Pattern

- $f_{sup}(p) = \sup(p) = \left\{ \begin{array}{ll}
    \# of distinct graphs $G_i$ that contain $p$ \\
    \hline
    \# of occurrences of $p$ over all $G_i \in D$
\end{array} \right.$

**Graph database**
\[
\begin{align*}
\text{The \textbf{p}: } & \quad \sup(p) = 2 \quad (\text{distinct graphs}) \\
& \quad \sup(p) = 6 \quad (\text{total \# q \ occurrences})
\end{align*}
\]

\textbf{Mining Task:} \textit{given a minimum support threshold \( \Theta \) find all frequent subgraphs, i.e. \( \sup(p) \geq \Theta \)}

\[=\]

\textbf{Potentially exponential space}

---

Graph \( G_1 \) is isomorphic to \( G_2 \) if

\[\exists \phi : 1-1 \text{ correspondence between } G_1 \text{ and } G_2\]

\textbf{s}\textit{uch that}

\[\phi \text{ is } \textbf{onto}\]

1. \textbf{Structure has to be preserved}

\[ (x, y) \in G_1 \iff (\phi(x), \phi(y)) \in G_2 \]

2. \textbf{Labels have to be preserved}

\[ L(x) = L(\phi(x)) \]

\[ L(x, y) = L(\phi(x), \phi(y)) \]

---

\[1 \quad 2 \quad 3 \quad 4 \leftrightarrow \phi(x) \in G_1\]

\[5 \quad 6 \quad 7 \quad 8 \quad \phi(\psi) \in G_2\]
graph isomorphism problem
is it in P?

\[ \begin{array}{ccc}
10 & a & 30 \\
20 & b & 70 \\
40 & c & 70 \\
\end{array} \]

\[ \begin{array}{ccc}
10 & 20 & 30 \\
10 & 30 & 20 \\
40 & 20 & 30 \\
40 & 30 & 20 \\
50 & 60 & 70 \\
50 & 70 & 20 \\
\end{array} \]

\( \phi \): Subgraph isomorphism

\[ \text{embedding} : (\phi(x_1), \phi(x_2), \ldots, \phi(x_n)) \rightarrow G_i \]

1. Systematic search
   - extend a pattern by one extra edge
   - pattern extension step
Step 1: Candidate generation / Pattern extension

For each "duplicate", list/copy only the
- (graph isomorphism)

Collect the frequency for each pattern P

Subgraph isomorphism step
Every pattern will be represented by its canonical code.

DFS tree

Extend:
1. Go depth-first to a forward extension from the current node.
2. Add backward edge closer to root first from the current vertex.
3. Store backtracking bottom-to-top only to rightmost path.

\[ S = \{ G_1, G_2, \ldots, G_t \} \]

Isomorphic graphs
Isomorphic graphs

\[ \text{min } \text{DFScode}(G) \rightarrow \text{a unique canonical representative in terms of the DFS-tree} \]

\[ \text{DFScode}(G_1) \]
\[ (0, 1, 6, 9, -) \]
\[ (1, 2, 9, 6, -) \]
\[ (1, 3, 6, 9, -) \]
\[ (3, 0, 6, 9, -) \]

\[ \text{DFScode}(G_2) \]
\[ (0, 1, 4, 9, -) \]
\[ (1, 2, 4, 9, -) \]
\[ (2, 0, 4, 9, -) \]
\[ (3, 0, 4, 9, -) \]

\[ \text{P:} \]

\[ \text{Is this canonical?} \]
Is this canonical?
→ is this a duplicate?

\[
\begin{array}{c|c|c}
\emptyset & 0 & 1 \\
\hline
a & a & a \\
\hline
b & b & b \\
\end{array}
\]

\[
\begin{array}{c|c|c}
\phi & 0 & 1 \\
\hline
a & 10 & 30 \\
\hline
b & 30 & 10 \\
\end{array}
\]

\[
\begin{array}{c|c|c}
\emptyset & 0 & 1 \\
\hline
a & 10 & 30 \\
\hline
b & 30 & 10 \\
\end{array}
\]

\[
\begin{array}{c|c|c}
\emptyset & 0 & 1 \\
\hline
a & 10 & 30 \\
\hline
b & 30 & 10 \\
\end{array}
\]