Topological Equivalence of Disjoint Sets and Union-Find

- Goal: Prove that Disjoint Sets = Union-Find = π_0 (path-connected components)
- Perspective: Computer Science + Topology
- Idea: Union-Find dynamically maintains π_0 of a graph (1-D simplicial complex).

Constraint

- Throughout this section, the simplicial complex K_t is constructed from the **underlying undirected graph** of the data (i.e., we ignore edge orientations when forming $E(K_t)$ Hence, $\pi_0(K_t)$ denotes the usual set of (undirected) path-connected components. Direction is only used later for causal summaries (SCC contraction) and does **not** affect π_0 .

Dataset:SAML-D

```python
import pandas as pd


data_path = r"C:\Users\Leon\Desktop\程式語言資料\python\TD-UF\Anti Money Laundering Transaction Data (SAML-D)\SAML-D.csv"
df = pd.read_csv(data_path)

# 簡單瀏覽前幾列
df.head()
```

[4] Python

	Time	Date	Sender_account	Receiver_account	Amount	Payment_currency	Received_currency	Sender_bank_location	Receiver_bank_location	Payment_type	Is_laundering	Laundering_type
0	10:35:19	2022-10-07	8724731955	2769355426	1459.15	UK pounds	UK pounds	UK	UK	Cash Deposit	0	Normal_Cash_Deposits
1	10:35:20	2022-10-07	1491989064	8401255335	6019.64	UK pounds	Dirham	UK	UAE	Cross-border	0	Normal_Fan_Out
2	10:35:20	2022-10-07	287305149	4404767002	14328.44	UK pounds	UK pounds	UK	UK	Cheque	0	Normal_Small_Fan_Out
3	10:35:21	2022-10-07	5376652437	9600420220	11895.00	UK pounds	UK pounds	UK	UK	ACH	0	Normal_Fan_In
4	10:35:21	2022-10-07	9614186178	3803336972	115.25	UK pounds	UK pounds	UK	UK	Cash Deposit	0	Normal_Cash_Deposits

Dataset:SAML-D

- As shown in Figure 1, each record in the SAML-D dataset contains both a **Sender_account** and a **Receiver_account** field.
  Although the transaction has an inherent direction (sender → receiver),
  the Union–Find model focuses only on whether two accounts **belong to the same connected group**.

- Hence, any two accounts that have ever transacted are treated as connected through an **undirected edge** in the topological complex.
  When constructing the 1-dimensional simplicial complex K_t,
  we can safely **ignore edge orientations** and consider the network as undirected for computing $\pi_0(K_\mathrm{t})$.
  Directional information will later be incorporated when analyzing **causal or temporal structures**,
  such as SCC contraction in the higher-level DAG.

Parent Function and Forest

- Let $X = \{x_1, x_2, \dots, x_{\mathrm{n}}\}$
- Define parent function:
- $p(x)$ = x if x is root; else $p(x) = parent(x)$

- Edges: $E = \{(x, p(x)) \mid x \in X, p(x) \neq x\}$
- Forest: $G = (X, E) \rightarrow each\, tree$
  $= one\ disjoint\ set$.

Topological Representation

- Build 1-D simplicial complex K from forest G.
- Define:
- $x \sim y \iff \exists\, path\, (v_0, \ldots, v_{\mathrm{k}})\, where\, (v_{\mathrm{i}}, v_{\mathrm{i+1}}) \in E$.
- $\pi_0(K) = X / \sim$ (set of connected components).

Find and Union Operations

- Find(x): if p(x)=x return x; else Find(p(x))
- $Union(x, y)$: $r_x = Find(x), r_y = Find(y)$
- If $r_x \neq r_y$,then link smaller rank root to higher rank root.
- Path Compression: $p(x) \leftarrow Find(p(x))$

Lemma 1 — Find ↔ Path Connectivity

- $Find(x) = Find(y) \;\Leftrightarrow\; x{\sim}y$
- $(\Rightarrow)\; Same\ root \;\Rightarrow\; exists\ path \;\Rightarrow\; connected.$
- $(\Leftarrow)\; Connected\ via\ Union\ edges \;\Rightarrow\; same\ root.$

Lemma 2 — Path Compression Invariance

- After compression: $\pi_0(K') = \pi_0(K)$
- Reason: Edge contraction inside a tree does not alter connectivity.

Lemma 3 — Union Operation

- If two roots r_1, r_2 connected:
- $\pi_0(K_{after}) = \pi_0(K_{before}) - 1$
- Reason: Adding edge (r_1, r_2) merges two components.

Theorem — Topological Equivalence

- $\forall t$: $\boldsymbol{\mathcal{P}}_{UF(t)} \cong \pi_0(K_t)$
- Mapping $\Phi_t(S_i) = C_i, where\ C_i$= component of K_t induced by S_i.
- Lemma 1: UF partition = path-connected partition.
- Lemma 2: Path compression preserves π_0.
- Lemma 3: Union merges two components.

Corollary — Dynamic Connectivity

- Find → returns component label
- Union → merges components
- Path Compression → edge contraction inside component

- ⇒ UF = dynamic maintenance of $\pi_0(K)$

Summary Diagram

- Forest → Graph → Topological Space → Components
- Union–Find Partition = $\pi_0(K)$
- Union-Find maintains dynamic connectivity in $O(\alpha(n))$ time.

Topological Interpretation of Union–Find

Dynamic Connectivity and Three-layer DAG Representation

Isomorphism Between Union–Find and $\pi_0(K_t)$

- At any time $t, P_{UF}\ (t)\ \cong\ \pi_0(K_{\mathrm{t}})$

- → Union–Find is not merely a data structure,
- but a discrete implementation of the path-connected component functor π_0.

- This isomorphism gives rise to three corresponding DAGs across operational layers:

1. Original Causal DAG

- • Constructed from temporal sequences and directional edges (sender → receiver)
- • Represents the base topological space K_0
- • Encodes real transaction or causal flow

- → K_0 serves as the foundation of the dynamic connectivity structure.

2. Rank-Filtration DAG (G_UF)

- • Generated by the sequence of Union operations
- • Each Union reduces π_0 by 1, forming a filtration chain:
- $K_0 \subset K_1 \subset \ldots \subset K_{\mathrm{t}}$
- • Edge condition: $(u, v) \in E_{rank} \Leftrightarrow Union(u, v)\, with\, r(u) \geq r(v)$
- • Since rank is non-decreasing, G_{UF} is a Directed Acyclic Graph (DAG).

3. Homotopic Quotient DAG (G′)

- • Strongly Connected Components (SCCs) are contracted into single points
- • Produces the quotient space $K' = K / \simeq$
- • Edge set: $E' = \{(C_{\mathrm{i}}, C_{\mathrm{j}}) \mid \exists (u, v) \in E, u \in C_{\mathrm{i}}, v \in C_{\mathrm{j}}\}$
- • If $H_1(G') = 0$, then G' is a Directed Acyclic Quotient up to Homotopy (DAQH)
- → Represents the partial order topology among stable groups.

Three Representations of π_0

Layer	Corresponding DAG	Topological Interpretation	Description
Static	Original DAG (K_0)	Base Topological Space	Represents real transaction or causal flow
Evolutionary	Rank-Filtration DAG (G_{UF})	Filtration $K_0 \subset K_1 \subset \cdots \subset K_t$	Structure generation process
Homotopic	DAQH (G')	Quotient $K' = K / \simeq$	Stable inter-group topology