The background features a complex, abstract design. It includes a network of thin, light-colored lines forming a web-like structure. Scattered throughout are small, colorful dots in shades of green, blue, and orange. A prominent feature is a large, semi-transparent white banner with a subtle geometric pattern, which serves as the backdrop for the title text. The overall aesthetic is technical and data-oriented.

# **Sequential Pattern and Sequential Pattern Mining**

# Sequence Databases & Sequential Patterns

---

- ❑ Sequential pattern mining has broad applications
  - ❑ Customer shopping sequences
    - ❑ Purchase a laptop first, then a digital camera, and then a smartphone, within 6 months
  - ❑ Medical treatments, natural disasters (e.g., earthquakes), science & engineering processes, stocks and markets, ...
  - ❑ Weblog click streams, calling patterns, ...
  - ❑ Software engineering: Program execution sequences, ...
  - ❑ Biological sequences: DNA, protein, ...
- ❑ Transaction DB, sequence DB vs. time-series DB
- ❑ Gapped vs. non-gapped sequential patterns
  - ❑ Shopping sequences, clicking streams vs. biological sequences


# Sequential Pattern and Sequential Pattern Mining

- Sequential pattern mining: Given a set of sequences, find the **complete set of frequent subsequences** (i.e., satisfying the min\_sup threshold)

A sequence database

SID	Sequence
10	<a( <u>ab</u> c)(a <u>c</u> )d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)( <u>ab</u> )(df) <u>c</u> b>
40	<eg(af)cbc>

A sequence: < (ef) (ab) (df) c b >



- An element may contain a set of **items** (also called **events**)
- Items within an element are unordered and we list them alphabetically

<a(bc)dc> is a subsequence of <a(abc)(ac)d(cf)>

- Given support threshold min\_sup = 2, <(ab)c> is a sequential pattern

# Sequential Pattern Mining Algorithms

---

- ❑ Algorithm requirement: **Efficient, scalable, finding complete set, incorporating various kinds of user-specific constraints**
- ❑ The Apriori property still holds: If a subsequence  $s_1$  is infrequent, none of  $s_1$ 's super-sequences can be frequent
- ❑ Representative algorithms
  - ❑ **GSP** (Generalized Sequential Patterns): Srikant & Agrawal @ EDBT'96)
  - ❑ Vertical format-based mining: **SPADE** (Zaki@Machine Learning'00)
  - ❑ Pattern-growth methods: **PrefixSpan** (Pei, et al. @TKDE'04)
- ❑ Mining closed sequential patterns: **CloSpan** (Yan, et al. @SDM'03)
- ❑ Constraint-based sequential pattern mining