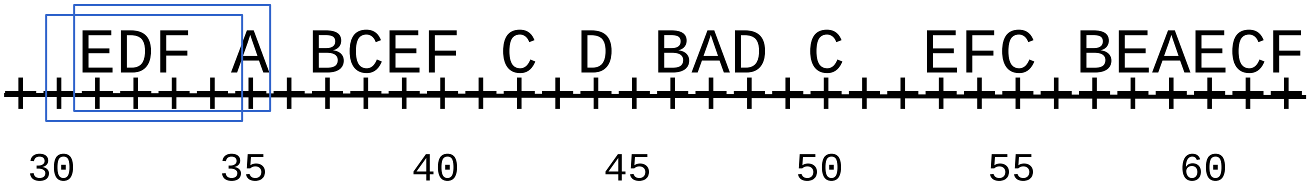
T-122.102 Seminar:
Discovery of frequent episodes in event sequences

- Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo.
  - Department of Computer Science, Series of Publications C, Report C-1997-15, University of Helsinki. 1997.
- Presented by Mathias Creutz, 18 Feb. 2003.
- **Motivation**
  - Most data mining techniques process **unordered** collections of data.
  - But there are important application areas, where the data consist of **sequences of events**, e.g.
    - alarms in a telecommunication network
    - user interface actions
    - crimes committed by a person
    - occurrences of recurrent illnesses
  - Goal: **Find relationships** between events occurring together.

Concepts

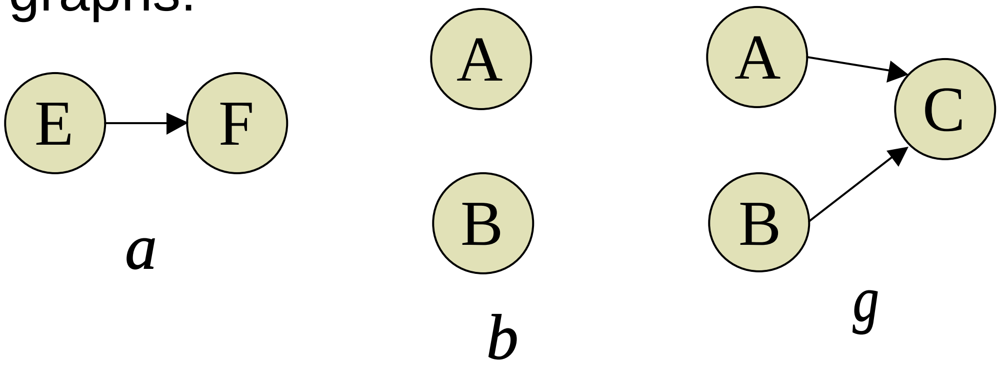


- **Event type** E: e.g., A, B, C, D, E, F.
- **Event**: Pair of an event type and its occurrence time (A, t), e.g., $(F, 40)$.
- **Event sequence**: Triple (s, T_{start}, T_{end}), where $s=<(A_1, t_1), (A_2, t_2), ..., (A_n, t_n)>$ is an ordered sequence of events, and T_{start} and T_{end} are the starting and ending times, e.g., $(<(E, 31), (D, 32), (F, 33)>, 30, 35)$.

  Note: The ending time 35 does **not** include $(A, 35)$!
- **Window**: Event sequence with a particular width, which equals $T_{end} – T_{start}$.
- **Episode**: Partially ordered set of events, e.g., whenever A and B occur (in either order), C occurs soon.

Episodes

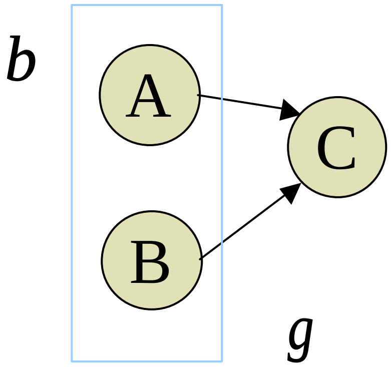
- Episodes can be described as directed acyclic graphs:



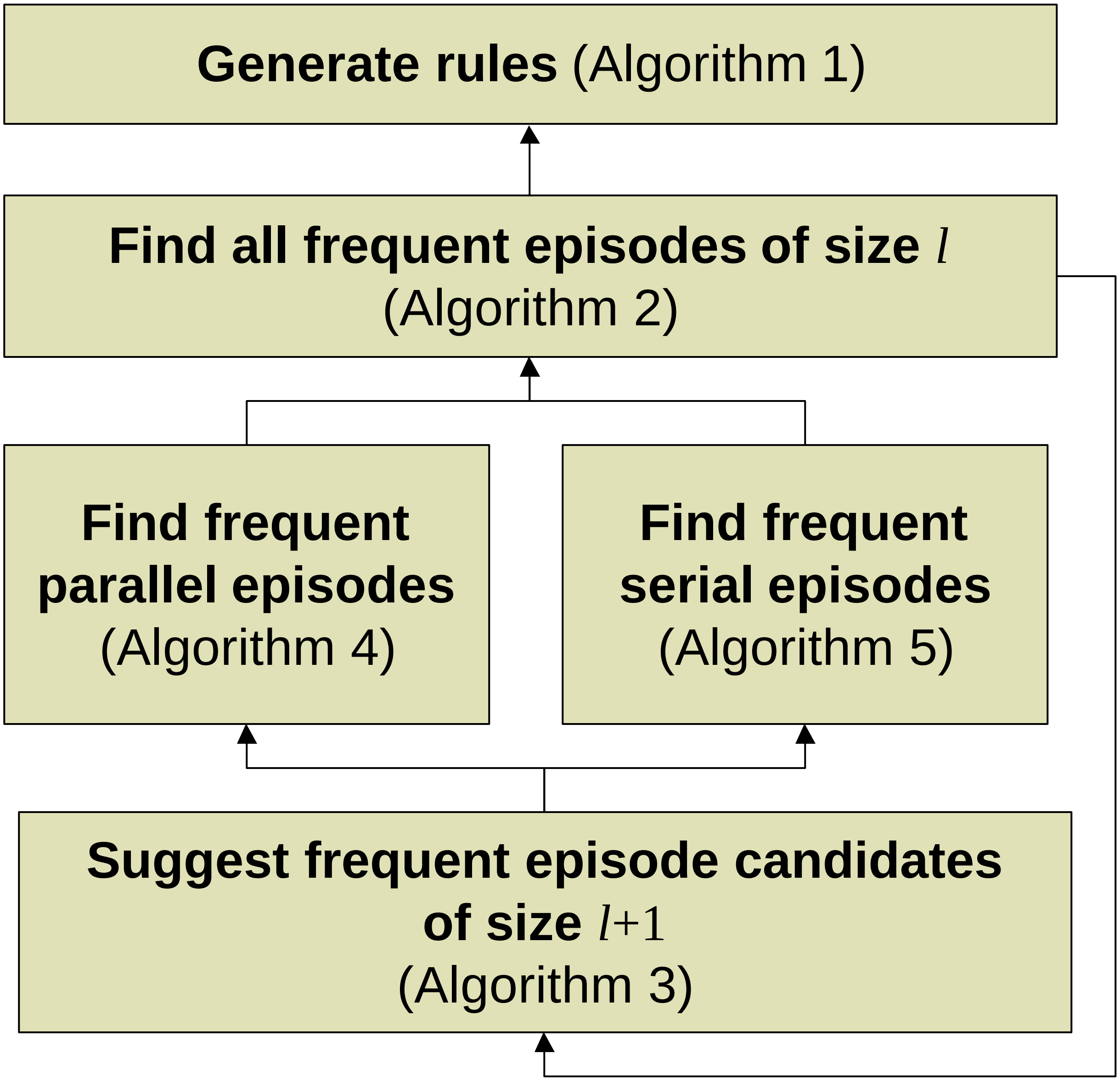
- **Serial episode**: Defined order between events, e.g., a. (But other events can intervene!)
- **Parallel episode**: No constraints on the relative order of the events, e.g., b.
- An episode is **injective** if no event type occurs twice in it, e.g., a, b, g.
- A **subepisode** contains part of the events of its **superepisode**, and the same ordering constraints apply, e.g., $b < g$.

Rules

- **Frequency of an episode**: Fraction of windows in which the episode occurs. This is a function of the window width.
- **Frequent episodes**: Set of episodes having a frequency over a particular **frequency threshold**.
- When the frequent episodes are known, **rules** can be obtained that describe connections between events, e.g.,
  - If b occurs in 4.2% and g in 4.0% of the windows, then there is a chance of 0.95 that C follows in a window, where A and B have been observed.
  - We can output the rule $b \rightarrow g$ with the confidence $fr(g) / fr(b) = 4.0 / 4.2 = 0.95$.



WINEPI algorithm


Generate rules (Algorithm 1)
Find all frequent episodes of size l (Algorithm 2)
Find frequent parallel episodes (Algorithm 4)
Find frequent serial episodes (Algorithm 5)
Suggest frequent episode candidates of size $l+1$ (Algorithm 3)

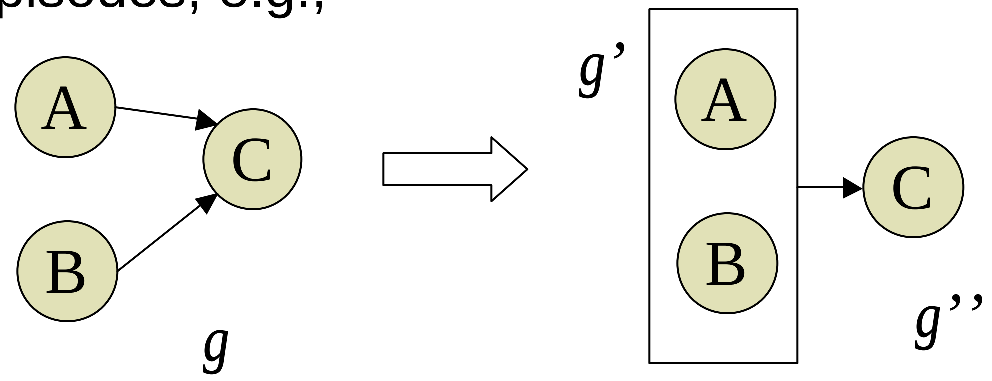
- **Lemma 1**: If an episode is frequent, then all its subepisodes are frequent.

WINEPI implementation details

- **Algorithm 2** (finding frequent episodes): **Breadth-first search:** Successively increase the size of the episodes (according to *Lemma 1*).
- **Algorithm 3** (generation of candidate episodes): **Sort** episodes **lexicographically** ⇒ All episodes that share the same first event types are consecutive in the episode list.
- **Algorithm 4** (finding parallel episodes): Slide a window over the event sequence. For every parallel episode, increase and decrease a **counter of** how many **events of the episode** are **within the window**. When an episode is entirely within a window, increase its frequency count.
- **Algorithm 5** (finding serial episodes): For every serial episode, use a **state automaton that accepts that episode** and rejects all others. Increase the frequency count of the episode when the accepting state is reached. Remove automata when they go out of the window.

General partial orders

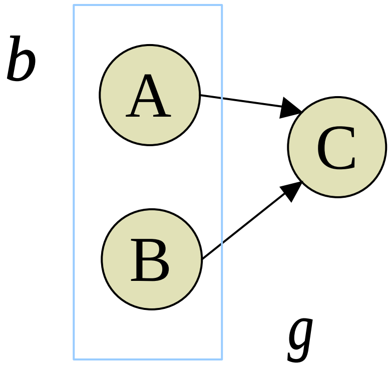
- An arbitrary episode can be reduced to a **hierarchical combination** of serial and parallel episodes, e.g.,

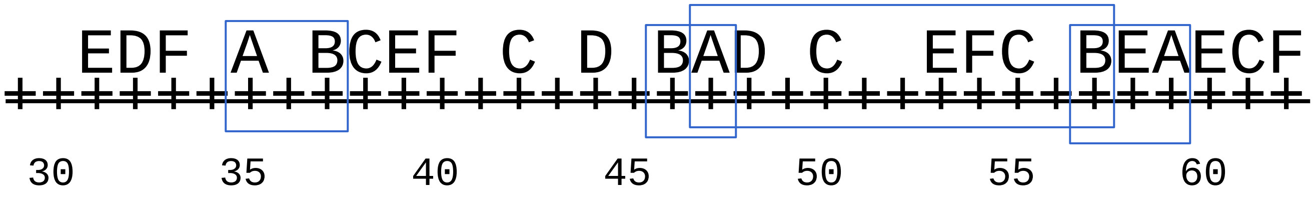


- **Complications:**
  - Sometimes necessary to duplicate event nodes (complicated for non-injective episodes)
  - Composite events have a duration unlike elementary events.
- **Practical and relatively fast alternative:**
  - Handle all episodes as parallel episodes
  - Check the correct partial ordering only when all events are within the window.

Minimal occurrences

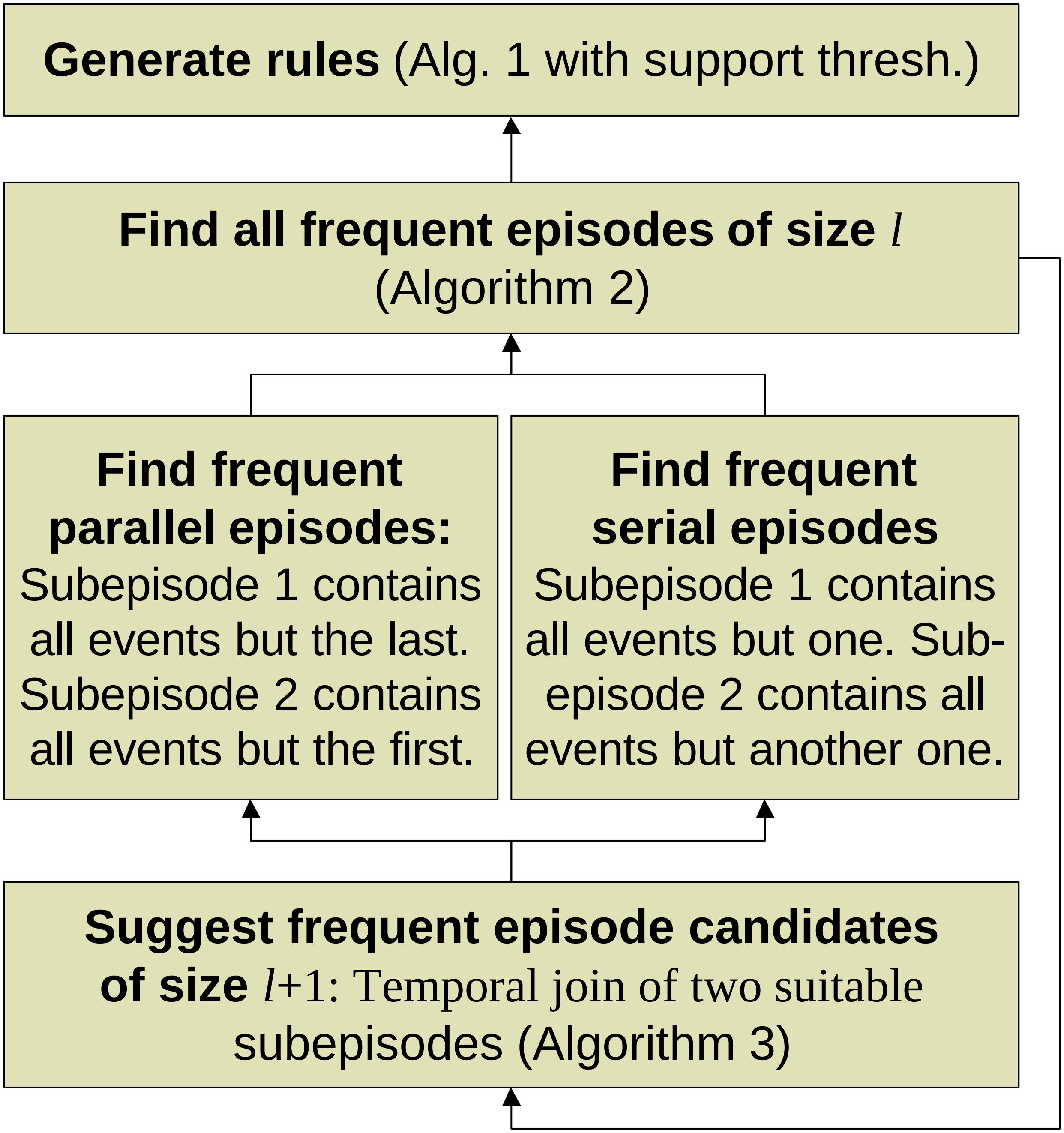
- Instead of using windows we can look at **exact occurrences of episodes**.
- This makes it more easy to find **episode rules** such as ”if A and B occur within 15 seconds, then C will follow within 30 seconds” $\Leftrightarrow b\ [15] \rightarrow g\ [30]$.
- **Minimal occurrence:** Shortest possible interval that contains a particular episode.
- E.g., the set of minimal occurrences of b : $mo(b) = \{[35,38[, [46,48[, [47,58[, [57,60[\}$





- Instead of frequency, we now use **support**, the number of minimal occurrences of an episode in an event sequence.
  - **Support threshold** vs. frequency threshold.

MINEPI algorithm



Generate rules (Alg. 1 with support thresh.)

Find all frequent episodes of size *l*
(Algorithm 2)

Find frequent parallel episodes:
Subepisode 1 contains all events but the last. Subepisode 2 contains all events but the first.

Find frequent serial episodes
Subepisode 1 contains all events but one. Subepisode 2 contains all events but another one.

Suggest frequent episode candidates of size ***l*+1**: Temporal join of two suitable subepisodes (Algorithm 3)

Experiments

- **Tests:**
  - **WINEPI**
    - **Serial episodes** (complex task) vs. **injective parallel episodes** (”easy” task)
    - **Different frequency thresholds**
    - **Different window widths**
  - **MINEPI**
    - **Serial episodes** vs. **parallel episodes**
    - **Different support thresholds**
    - **Different number of times bounds for rules**
    - **Different confidence thresholds for rules**
- **Evaluation:**
  - **Time consumption**
  - **Quality of candidate generation**
  - **Comparison of WINEPI and MINEPI**
    - Differences in frequent episodes found

Data used in experiments (1)

- **Telecommunications network fault management database**
  - 73 679 alarms covering 7 weeks
  - 287 different types of alarms
  - Average: 1 alarm / minute
  - Alarms tend to occur in bursts: It is possible to have 40 alarms in one second.
- **WWW server log**
  - Department of Computer Science at the University of Helsinki
  - 116 308 events (WWW pages fetched in February and March 1996)
  - 7634 different pages
- **English text 1**
  - GNU man pages
  - 5415 words (1102 word types)
  - Each word is indexed consecutively to give it a ”time”. Sentence boundaries cause a gap.

Data used in experiments (2)

- **English text 2**
  - The same GNU man pages, with non-informative words such as articles, prepositions and conjunctions stripped off
  - 2871 words
  - 905 word types
- **Protein sequences**
  - PROSITE database of the ExPASy WWW molecular biology server of the Geneva University Hospital and the University of Geneva
  - DNA and protein patterns
  - Target: Family of 7 sequences known to contain the string `GFRGEAL.`
  - 4941 events
  - 22 event types

Results

- **WWW server log**
  - Users navigate through long paths from the homepage of the department to the pages of individual courses (rather than using bookmarks).
- **Text database**
  - Only few rules can be found, e.g.,
    - the, value [2] → the, value, of [3]
  - Window widths from 24 to 50 produce the same amount of episodes.
- **Protein sequences**
  - Found 17 episodes of length 7 or 8
  - `GFRGEAL` is among them, and so are patterns with an 8th symbol fairly near, e.g., `GFRGEAL*S`

Conclusions

- Rules of **WINEPI** have nice **interpretations as probabilities** concerning randomly chosen windows.
- Rules of **MINEPI** usually **more informative**.
- **WINEPI more efficient** in the first phases of the discovery.
- **MINEPI outperforms WINEPI** in the later iterations.
- Methods could be modified for **cross-use**.