Temporal pattern mining in symbolic time point and time interval data

Fabian Moerchen
Siemens Corporate Research
Princeton, NJ

SIEMENS

Introduction

SIEMENS

Temporal Data Mining

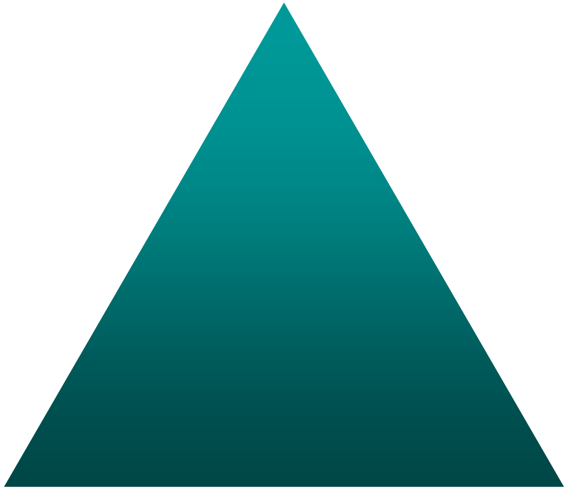
- Any data mining task involving some dimension of time.
  - Includes temporal association rules, evolutionary clustering, spatio-temporal data minig, trajectory clustering, …
- **Time Series Data Mining**
  - Mining of sequence(s) of observations over time
    - Clustering
    - Classification
    - Indexing
    - Anomaly detection
    - Prediction
    - …
    - Temporal pattern mining
      - Temporal pattern languages
      - Pattern mining algorithms

Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

Introduction

Different aspects of temporal pattern mining

Temporal Data Models
How is temporal data represented?
Time point vs. time interval

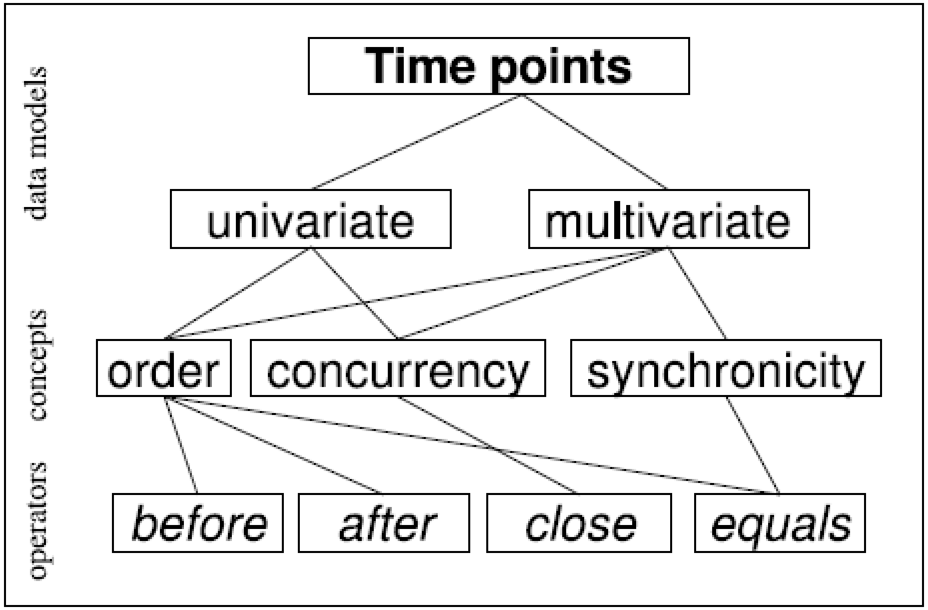


Temporal Concepts
What are the desired semantics?
Order vs. concurrency

Temporal Operators
How are data elements compared and combined by the algorithms?

Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

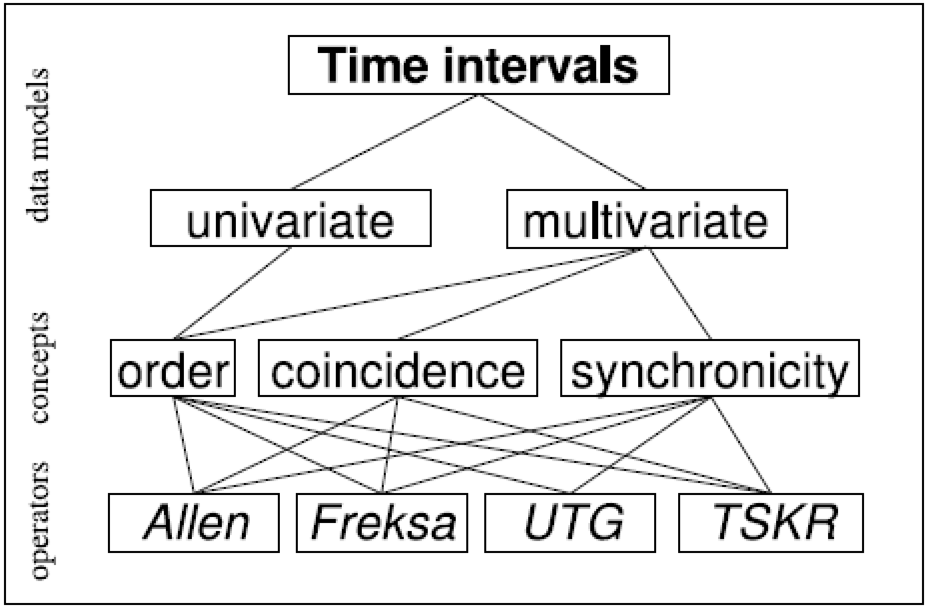
Introduction
Time point models, concepts, and operators



- In time point data models observation are associated with a **specific point in time.**
- If several 'dimensions' are observed in parallel the model is **multivariate**.
- Most commonly mined concept is **order** or the less strict **concurrency**.

Time interval models, concepts, and operators

SIEMENS



- In **time interval** data models observation are associated with the time between two time points.
- If several 'dimensions' are observed in parallel the model is **multivariate**.
- Most methods mine all three temporal concepts: **order, concurrency, synchronicity**.

Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

Temporal Data Models

SIEMENS

Definition of terminology

- Time is continuous, computers are binary.
- We assume temporal data is represented with discrete **time points**.
- A **time series** is a set of unique time points.
- A **time sequence** is a multi set of time points.
- A pair of time points defines a **time interval**, inclusively.
- Two intervals **overlap** if there is at least one time point that lies within both intervals.
- An **interval series** is a set of non overlapping time intervals.
- An **interval sequence** can include overlapping and equal time intervals.
- The series data types can be **univariate** or **multivariate**.

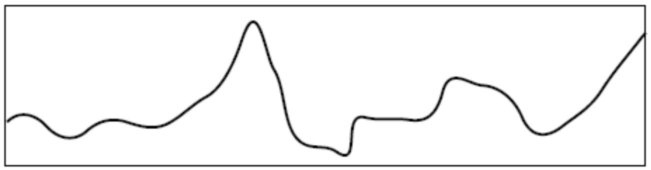
Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

Temporal Data Models

Common models and transformations

SIEMENS

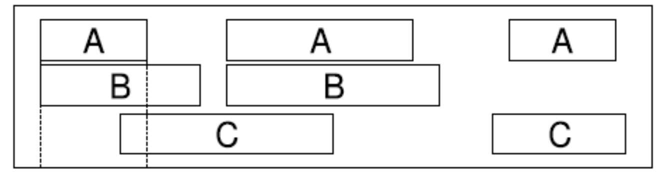
A **numeric time series** is a time series with numerical values for each time point.



Sensor data

Discretization
Segmentation
Motif discovery

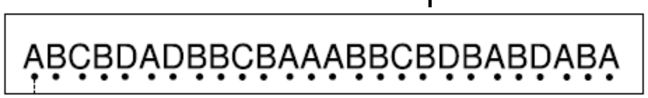
A **symbolic interval sequence** has overlapping intervals with nominal values.



Sign language

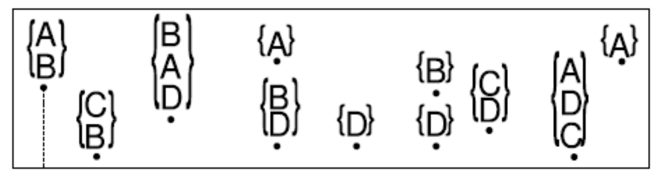
Discretization

A **symbolic time series** is a time series with nominal values for each point.



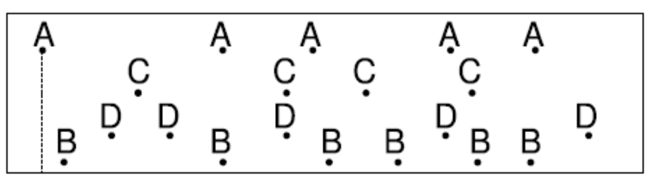
DNA sequences

An **itemset sequence** is a time sequence with sets of nominal values assigned to each time point.



Shopping baskets for same customer

A **symbolic time sequence** has nominal values with possible duplicate time points



Events from machine service logs

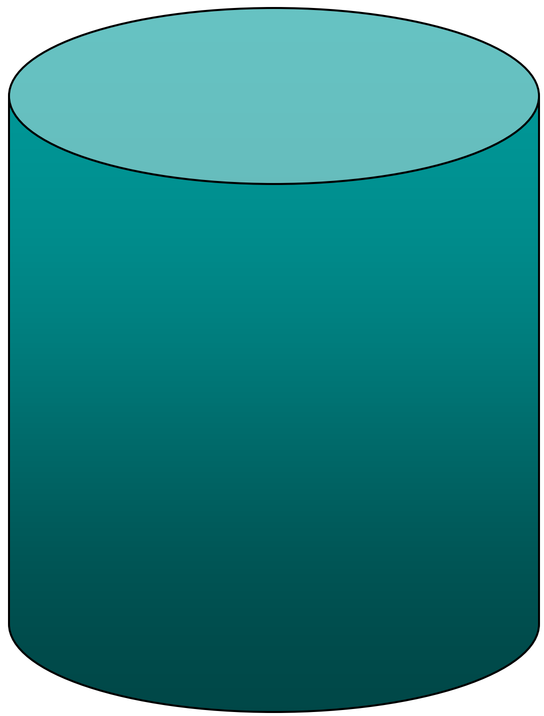
Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

Common data bases

Windowing

Single long observation

- Stock prices
- Sensor data
- DNA sequence
- Documents
- Machine logs



Many (short) observations

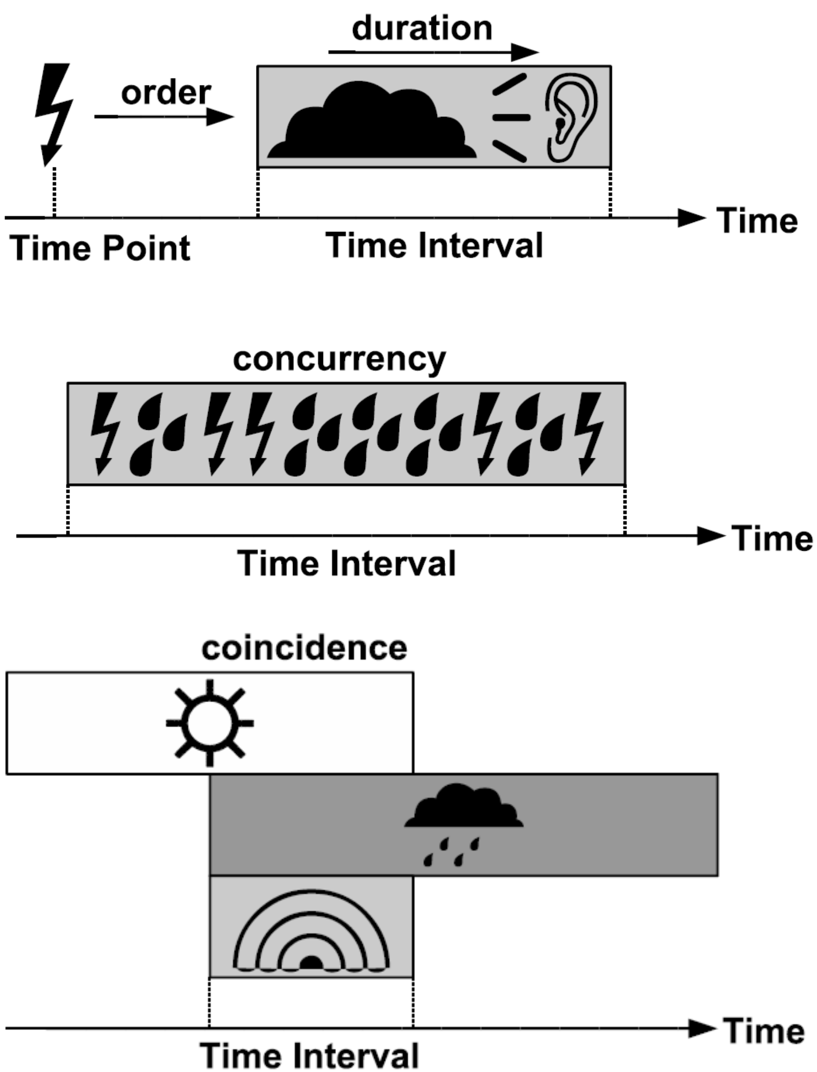
- Website click streams
- Shopping profiles
- Gene expressions
- Headlines
- Twitter
- Scenes in video
- Musical melodies
- Shapes

Temporal Concepts and Operators

Temporal Concepts

Semantic categories for temporal operators

SIEMENS



Duration is the persistence of an event over several time points.

Order is the sequential occurrence of time points or time intervals.

Concurrency is the closeness of two or more temporal events in time in no particular order.

Coincidence describes the intersection of several intervals.

Synchronicity is the synchronous occurrence of two temporal events.

Periodicity is the repetition of the same event with a constant period.

Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

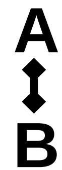
Time point operators

- Time point operators expressing strict order: **before** / **after** (→)

$$A \longrightarrow B$$

- Time point operator expressing concurrency: **close**

- Time point operator expressing synchronicity: **equals**



- Special cases
  - With constraints: **shortly before**, **closely after**
  - With specific granularity: **next business day**

Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

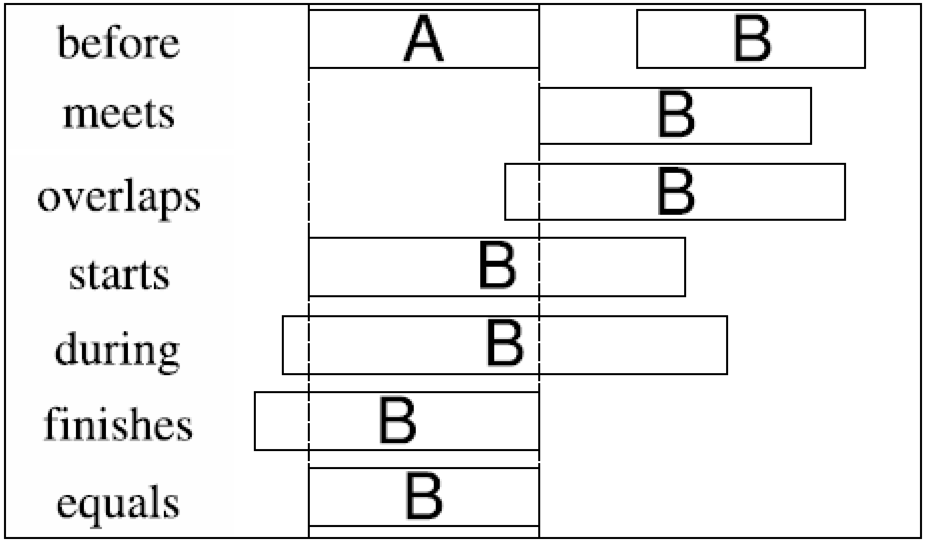
SIEMENS

Time interval operators

- **Time interval operators**
  - Allen’s interval relations [Allen 1983]
  - Freksa’s semi-interval relations [Freksa 1992]
  - Reichs’s interval and interval point relations [Reich 1994]
  - Roddick’s Midpoint interval relations [Roddick/Mooney 2005]
  - Other operators [Villafane 1999], [Ultsch 1996], [Moerchen 2006]

Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

Temporal Operators
Allen’s interval relations



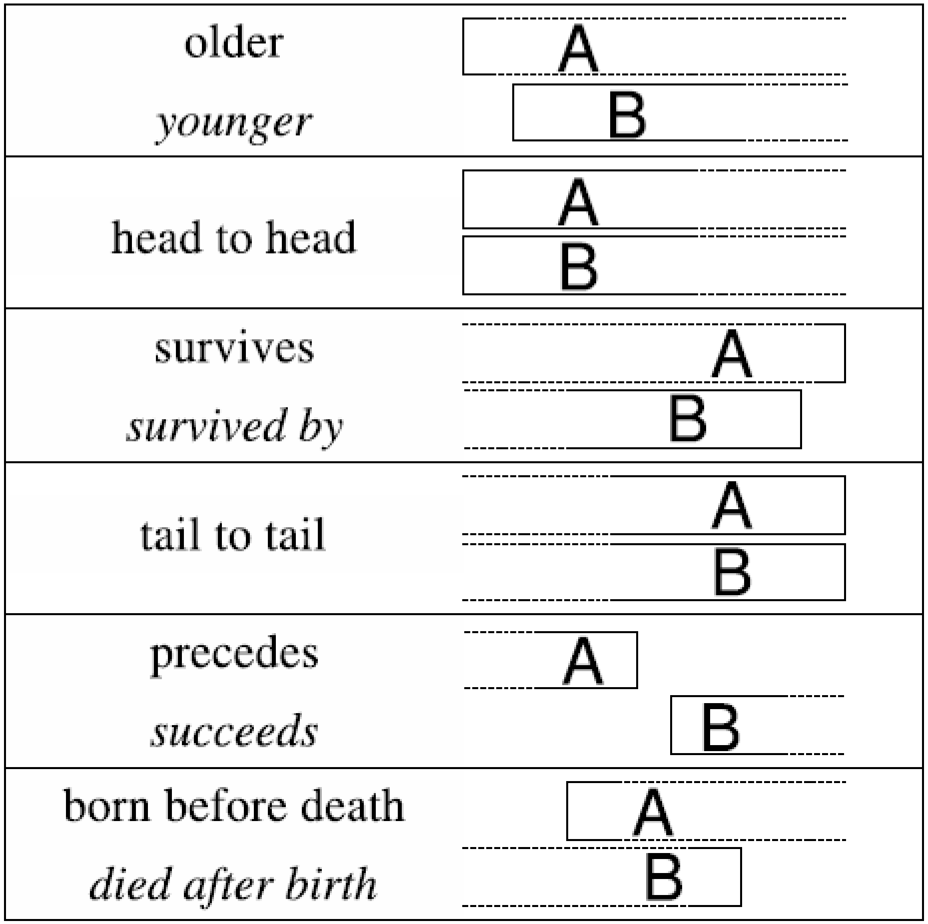
- 13 relations forming an algebra.
- **Any two intervals have exactly one of the relations**.
- Invented in AI for temporal reasoning: given facts associated with time interval derive additional facts or answer specific questions.
- Later widely used in data mining.
- **Disadvantages for knowledge discovery!**
- Thresholds and fuzzy extensions solve some of the problems.

Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

Temporal Operators

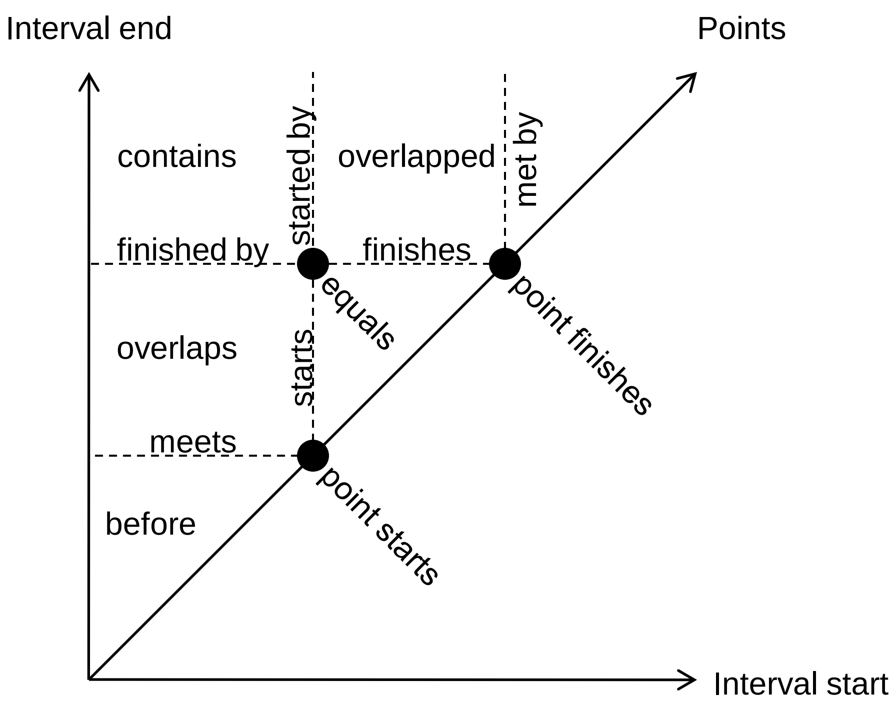
Freksa's semi-interval relations

SIEMENS



- **Semi-intervals**: one interval boundary unknown.
- Two relations between start or endpoints of the two intervals suffice to uniquely identify the relation.
- Easier to **represent incomplete or coarse knowledge**.
- Not widely used in data mining (yet).
  - [Rainsford/Roddick 1999]
  - [Moerchen/Fradkin 2010]

Reich's interval/point relations



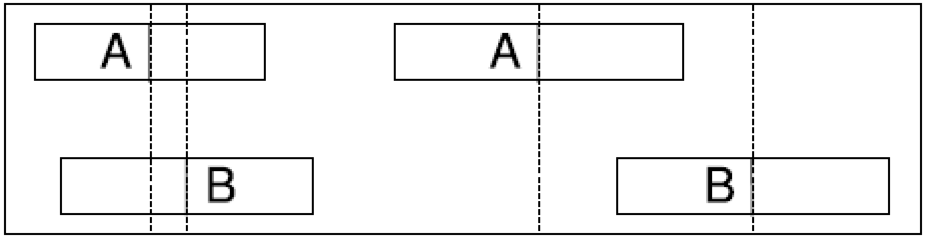
- Extension of Allen's relation to points.
- Only 5 more relations while [Vilain 1982] had 13.
  - point finishes and inverse
  - point starts and inverse
  - point equals
- Also proposed relations for branching time to reason in multiple future worlds.
- Supported in principle by [Moerchen/Fradkin 2010]

Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

Temporal Operators

Roddick's mid-point interval relations

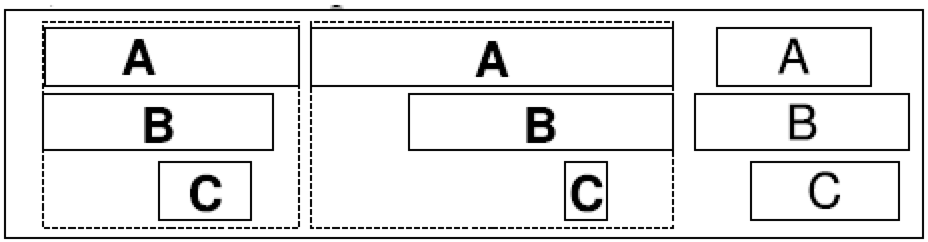
SIEMENS



- Allen's relation extended by **relation of each interval midpoints to the other interval**.
- Two versions of overlaps shown: midpoints within other intervals (largely overlap) or not (overlap to some extend).
- 9 versions of overlaps
- **Total of 49 relations!**
- Designed to coarse data with arbitrary local order.
- Shares disadvantages with Allen's relations, in some respects even worse.

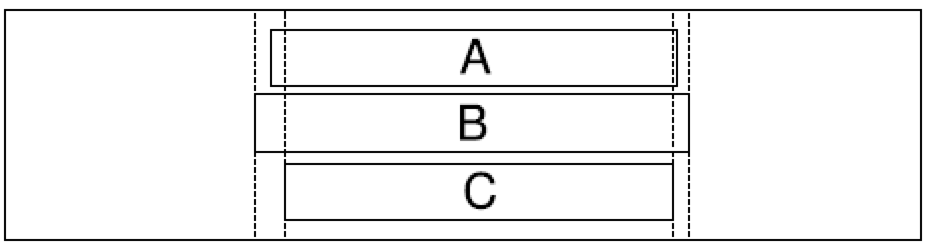
Other interval relations

SIEMENS

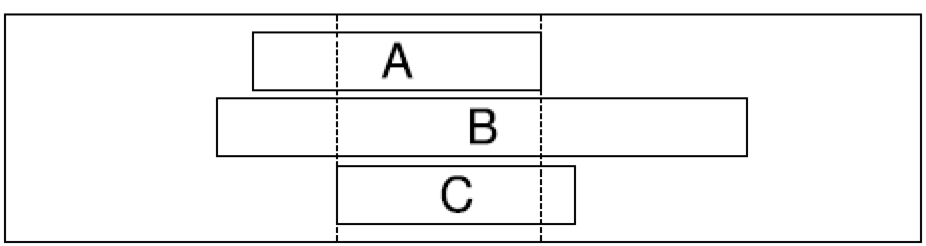


- **A contains B contains C**
  - Contains is equivalent to

  (A equals B) or (B starts A) or (B during A) or (B ends A)
  - [Villafane 1999]



- **A,B,C approximately equal**
  - Allow slight variations.
  - N-ary operator
  - [Ultsch 1996]



- **A,B,C coincide**
  - Intersection of intervals
  - N-ary operator
  - [Moerchen 2006]

Temporal patterns
Time point patterns

Frequent and closed patterns

SIEMENS

- **Frequent patterns**
  - Frequency = number of occurrences / size of database.
  - Specify minimum frequency for patterns to be significant.
  - Redundancy problem: Even if a sub-pattern has exactly same frequency as a super-pattern both are reported as frequent.
- **Closed frequent patterns**
  - Cannot make the pattern more specific without decreasing the support.
  - Sequence example (adapted from [Wang/Han 2004])

Database
C A A B C
A B C B
C A B C
A B B C A
B C

A B observed 4x
B C observed 5x
A B C also observed 4x
C A B C observed 2x
other extensions of A B C observed 1x

B C is closed

A B C is closed

A B is not

Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

Substring patterns

- **Substring patterns**
  - Sequence of symbols without gaps
  - Expresses concept of **order**
  - Example: **B** → **C** → **B**

A**BCB**DADB**BCB**AAAB**BCB**DBABDABA

- **Regular expression patterns**
  - Extension to allow gaps (via wildcards), negations, repetitions, etc.
  - Example: **B** → **¬C** → **A | B**

ABC**BDA**DBBC**BAA**ABBC**BDB**A**BDA**BA

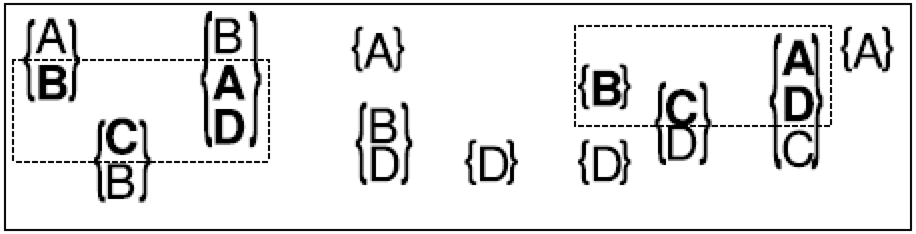
Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

Substring pattern algorithms

- **Algorithms for substring patterns**
  - Special data structures are used to represent the data and derive frequent patterns efficiently.
  - **Suffix trees**
    - Allowing wildcards [Vilo 2002]
    - Reporting over/under represented patterns [Apostolico et al. 2000]
  - **Suffix arrays**
    - Optimal time [Fischer et al. 2006]
    - Space efficient [Fischer et al. 2007]
  - Many applications in **bioinformatics**
    - Finding discriminative patterns
    - Sequence alignment

Sequential patterns

- **Sequential patterns** [Agrawal/Srikant 1995]
  - Sequence of (sets of) symbols
  - Expresses concept of **order**
  - Example: **{B}** → **{C}** → **{A,D}**



  - Observed sets of symbols at each time point can contain more symbols.
  - Gaps are allowed.
  - Motivated by example of repeated purchases by customers, i.e., database of many short sequences.

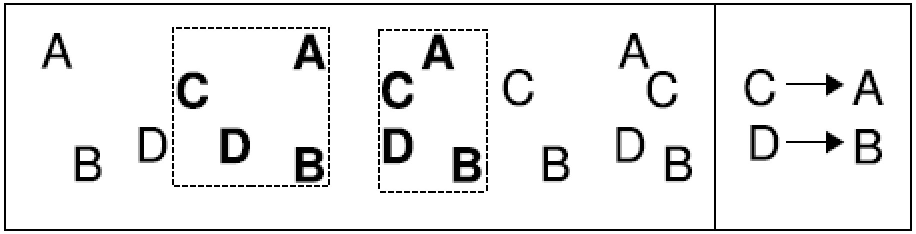
Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

Sequential pattern algorithms

- Algorithms for **sequential patterns**
  - AprioriAll [Agrawal/Srikant 1995],
  - SPADE [Zaki 1998]
  - PrefixSpan [Pei et al. 2001]
- Algorithms for **closed sequential patterns**
  - CloSpan [Yan et al. 2003]
  - BIDE (BIDirectional Extension checking) [Wang/Han 2004]
  - BIDEMargin [Fradkin/Moerchen 2010]
    - enforces margin of support difference among reported patterns
- Variations
  - Regular expressions [Garofalakis et al. 1999]
  - Multivariate data [Pinto et al. 2001]
  - Allow mismatches [Kum et al. 2003][Zhu et al. 2007]
  - Generators [Lo et al. 2008]

Episode patterns

- **Episode patterns** [Mannila et al. 1995, 1996, 1997]
  - Constrained partial order of (sets of) symbols.
  - Expresses concepts of **order** and **concurrency**.
  - Example: **C → A and D → B within a time window**



  - **Serial episodes**: Order relation between symbols (or episodes).
  - **Parallel episodes**: Symbols or episodes observed within time window.
  - **Episodes**: in principle arbitrary partial order of symbols, but often combination of serial and parallel episodes.
  - Motivated by patterns in long message stream from telecommunication equipment.

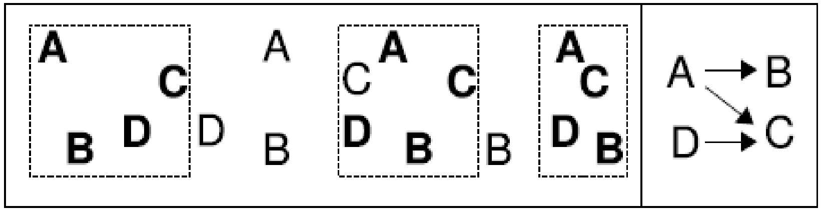
Episode pattern algorithms

SIEMENS

- **Apriori-style** algorithms for episodes.
  - Given maximum window length
    - WINEPI: percentage of windows with the pattern [Mannila et al. 1995]
    - MINEPI: windows that do not contain sub-windows with the pattern [Mannila/Toivonen 1996]
  - Given maximum gap [Meger/Rigotti 2004]
- Algorithms for **closed** (groups of) **episodes** [Harms et al. 2001]
- Determining **significance** beyond the concept of frequency.
  - Significance of episodes against Bernoulli and Markov background models [Gwadera et al. 2005]
  - Formal relation of Episodes with HMM [Laxman et al. 2005]

Partial order patterns

- **Partial order patterns** [Casas-Garriga 2005]
  - Partial order of (sets of) symbols.
  - Expresses concepts of **order** and **concurrency**.
  - Example: **A → B and A → C and D → C**



  - Order between some symbols (A and D) is unspecified (hence partial).
  - **Not equivalent to serial-parallel episodes!** [Pei et al. 2006]
    - Example above cannot be expressed as combination of parallel and serial episodes.
    - Definition of [Mannila et al. 1995] captures this but algorithms focus on serial/parallel episodes and many other authors have done the same (or even only deal with serial episodes).

Time point patterns
Closed partial order pattern algorithms

- Three step algorithm [Casas-Garriga 2005] [Moerchen 2006]
  - **Mine closed sequential patterns.**
  - Mine maximal **conjunctive groups** of non-redundant sequential patterns that are observed in the same windows. [Casas-Garriga 2005]
    - [Moerchen 2006]
      - Interpret sequential patterns as items and windows as itemsets.
      - **Mine closed itemsets** = maximal groups**.**
      - Remove redundant sequential patterns in each group
  - **Convert each group to partial order** [Casas-Garriga 2005]
    - Every sequential patterns is a path in the directed graph.

Time point patterns
Closed partial order example

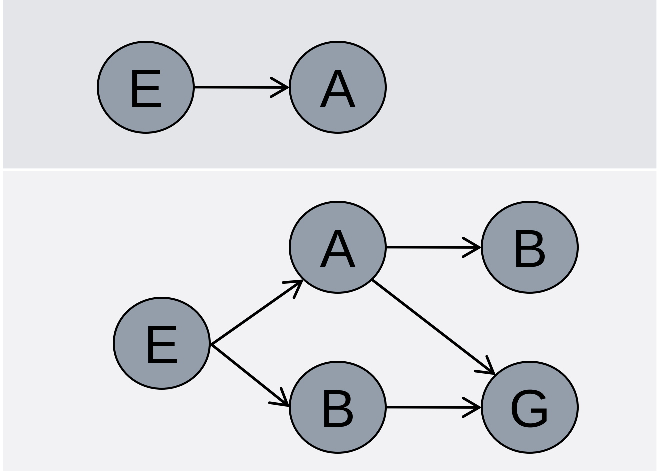
SIEMENS

1. Sequences	
S1	{D}{E}{A}
S2	{E}{ABF}{G}{BDE}
S3	{E}{A}{B}{G}

2. Closed seq. patterns		S1	S2	S3
{E}{A}	3x	x	x	x
{E}{A}{B}	2x		x	x
{E}{A}{G}	2x		x	x
{E}{B}{G}	2x		x	x

3. Closed item (=seq. pattern) sets	
{E}{A}	3x
{E}{A}{B}, {E}{A}{G}, {E}{B}{G}	2x

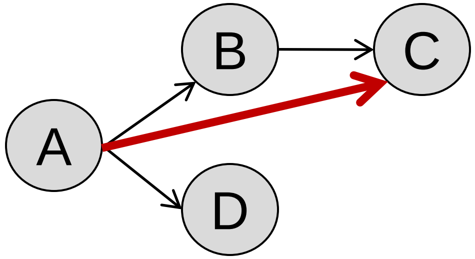
4. Closed partial orders

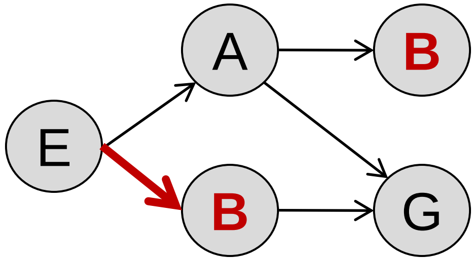


Closed partial order pattern algorithms

SIEMENS

- Efficient algorithm for closed partial orders [Pei et al. 2006]
  - Interpret edges in partial order as items.
  - **TranClose**: mine closed item (=edge) sets of transitive closure
    - **ABC** has edges **A → B, A → C, B → C**
    - Reduce resulting patterns
  - **Frecpo**: PrefixSpan-like algorithms with pruning
    - Directly generates transitive reduction.
    - Forbidden edges: not **A → C** if already **A → B** and **B → C**.
  - Both do not allow for repeating symbols in patterns.
    - **E → B** is forbidden but it's not the 'same' B





Time point patterns
Conjunctive sequential patterns

- Mining conjunctive groups of sequential patterns [Raïssi et al. 2008] .
- Closed sequential pattern mining algorithms use equivalence classes based on prefix.
- Non-derivable does not work for single sequential patterns and these classes.
  - Lower bound is always 0
- In contrast conjunctive groups of sequential patterns form equivalence classes.
  - see also [Casas-Garriga 2005], [Harms et al. 2001]
- Conjunctive groups can be used just similar to itemsets.
  - Mine closed groups of sequential patterns [Moerchen 2006]
  - Non-derivable groups of sequential patterns [Raïssi et al. 2008]
  - Generate sequential association rules [Raïssi et al. 2008]
  - Mine generators for groups of sequential patterns.

SIEMENS

Sequential patterns vs. Episodes vs. Partial Orders

Supports	Sequential Patterns	Episodes parallel	serial	serial/ parallel	Partial Orders
Itemsets	✓	✓ Unordered set of symbols	✘ Typically not discussed	✘ Typically not discussed	✓ Often straightforward extension
Full partial order	✘	✘	✘	✘ Typically not discussed	✓
Closedness	✓ Easy - CloSpan	✓ Easy - Closed itemsets	✓ Easy - Equivalent to seq. patterns	✓ Requires looking at groups of episodes	✓ Requires looking at groups of seq. pat.
Condensed representation	✓ several	✓	✓	✘	✓ Margin-closed

Summary

SIEMENS

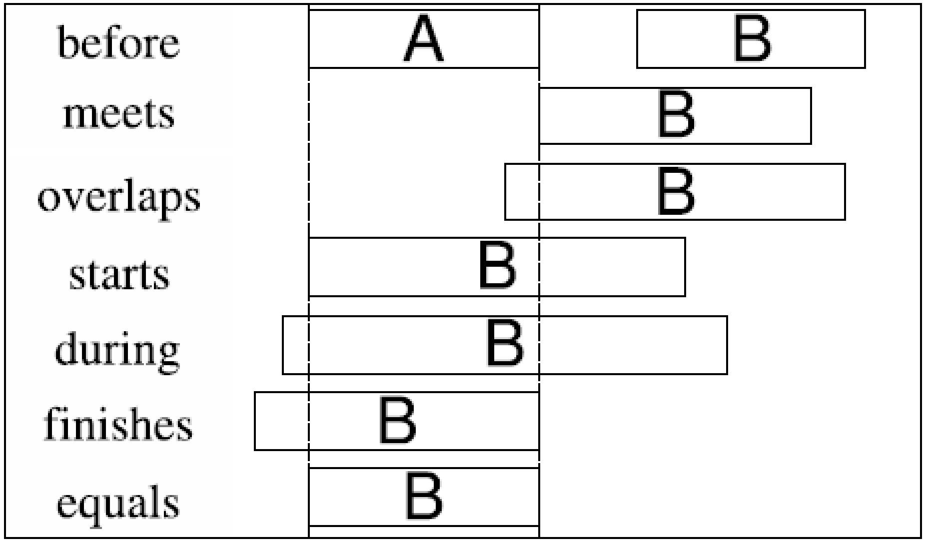
- Use substrings if no gaps required.
- Use closed sequential patterns if gaps are and strict ordering are required.
- Partial orders are the most flexible representation but also more complex to mine.

	Univariate symbolic time series	Multivariate symbolic time series	Symbolic time sequence	duration	order	concurrency	synchronicity	partial order
Substrings	✓				✓			
Sequential Patterns	✓	✓	✓		✓		✓	
Episodes	✓	✓	✓		✓	✓		✓
Partial orders	✓	✓	✓		✓	✓	✓	✓

Temporal patterns

Time interval patterns

Temporal Operators
Allen's interval relations

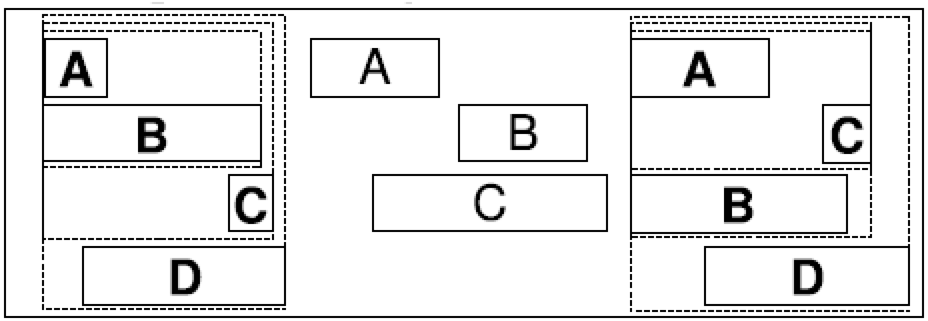


- 13 relations forming an algebra.
- **Any two intervals have exactly one of the relations**.
- Invented in AI for temporal reasoning: given facts associated with time interval derive additional facts or answer specific questions.
- Later widely used in data mining.
- **Disadvantages for knowledge discovery!**
- Thresholds and fuzzy extensions solve some of the problems.

Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

Patterns using Allen's relations

SIEMENS



A1 patterns [Kam/Fu 2000]

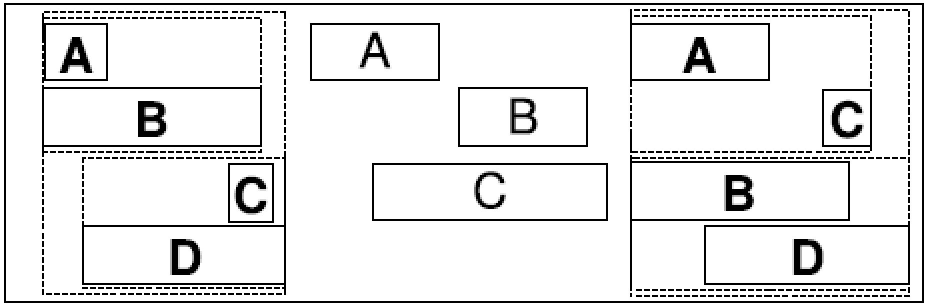
- Combine two intervals with relation from Allen.
- Combine resulting pattern with another single interval.
- Ambiguous representation [Moerchen 2006]:

(((A starts B) overlaps C) overlaps D)

- or

(((A before C) started by B) overlaps D)

Time interval patterns
Patterns using Allen's relations



Fluents [Cohen 2001]

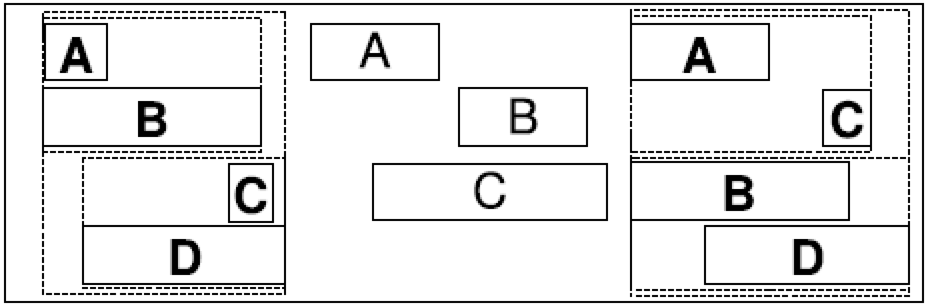
- Combine two intervals with relation from Allen.
- Combine two patterns.
- Ambiguous representation [Moerchen 2006]:

(A starts B) overlaps (C during D)

- or

(A before C) starts (B overlaps D).

Time interval patterns
Patterns using Allen's relations



	A	B	C	D
A	Equals	Starts	before	overlaps
B	Started by	Equals	overlaps	overlaps
C	after	Overlapped by	Equals	during
D	Overlapped by	Overlapped by	contains	Equals

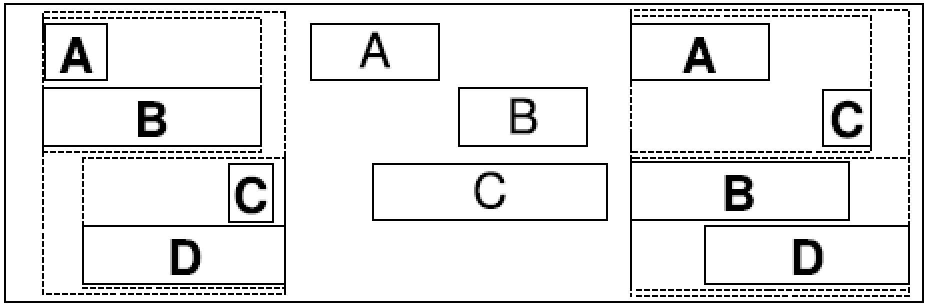
[Hoeppner 2001]

- Set of intervals with pairwise relations of Allen.
- K(K-1)/2 relations for K intervals.
- **Not ambiguous** but difficult to write out as rule [Moerchen 2006].
- Transitivity of Allen's relations can be used to derive some relations from others.

Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

Patterns using Allen's relations

SIEMENS



= both =

$A^+=B^+>D^+>A^->C^+>B^->C^->D^-$

Sequence of interval boundaries
[Wu/Chen 2007]

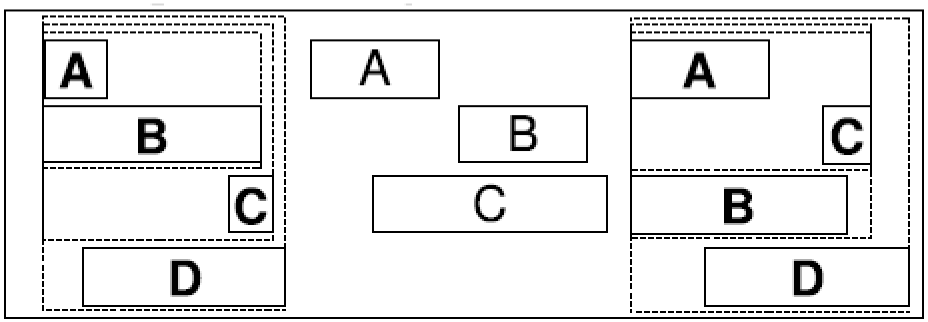
- Represent each interval with start and end points (A^+ vs. A^-).
- Pattern with K intervals represented by sequence of 2K boundaries.
- Not ambiguous and equivalent to Hoeppner, but more compact:

2K+(2K-1) vs. K(K-1)/2

Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

Time interval patterns

Patterns using Allen’s relations



= both =

((A starts [0, 0, 0, 0, 1] B)

overlaps [0, 0, 0, 1, 0] C)

overlaps [1, 0, 0, 2, 0] D)

Contains

Finished by

Meets

OverlapsS

Starts

Nested representation with counters [Patel et al 2008]

- Fix ambiguity of nested representation by adding counters for 5 relations
- K+5(K-1)
- Not as compact as Wu/Chen and not very readable.

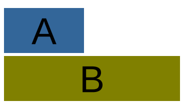
Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

Time interval patterns

Allen's relations pattern mining algorithms

SIEMENS

- Apriori-style [Hoeppner 2001]
  - Combine two length k patterns with common k-1 prefix.
  - Use transitivity of Allen's relations to prune some candidates for the relations of the two kth intervals.

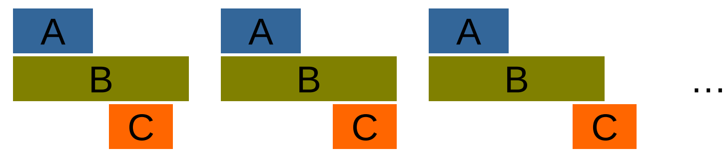


	A	B
A	Equals	Starts
B	Started by	Equals

\+

	A	C
A	Equals	before
C	after	Equals

=



	A	B	C
A	Equals	Starts	before
B	Started by	Equals	?
C	after	?	Equals

- **B {contains, ended by, overlaps, meets, before} C**
- Pruned relations: {after, met by, overlapped by, started by}

Allen's relations pattern mining algorithms

SIEMENS

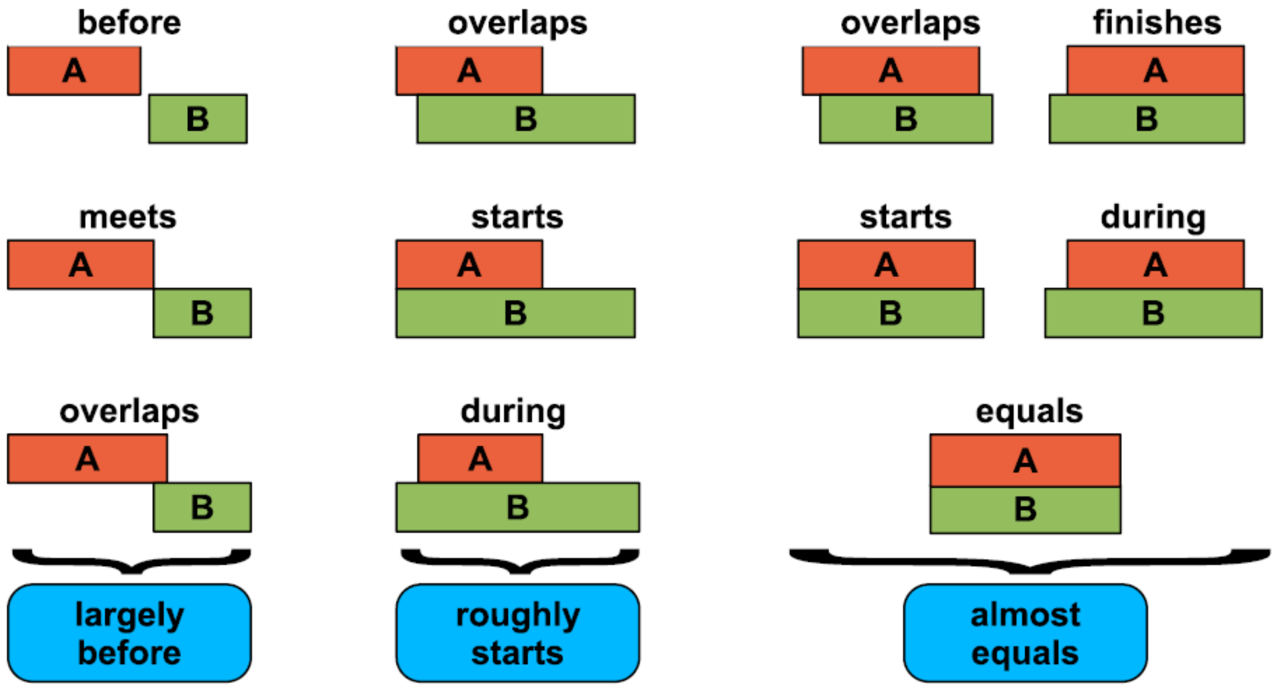
- Early work using only Apriori-style algorithms. Recently more efficient algorithms have been proposed:
  - H-DFS (Hybrid Depth First Search) [Papaterou et al. 2005]
    - Enumeration tree
  - ARMADA [Winarko/Roddick 2006]
    - Adaptation of sequential pattern mining algorithm to intervals.
  - TPrefixSpan [Wu/Chen 2007]
    - PrefixSpan using interval boundaries pruning patterns that are not valid interval patterns.
  - IEMiner [Patel et al 2008] (compares to TPrefixSpan and H-DFS)
    - Apriori using nested representation with counters and pruning.
  - KarmaLego [Moskovitch/Shahar 2009] (compares to Armada and H-DFS)
    - Enumeration tree exploiting transitivity.

SIEMENS

Excursus: What’s wrong with Allen?

Disadvantages of Allen's relation

Allen is not robust



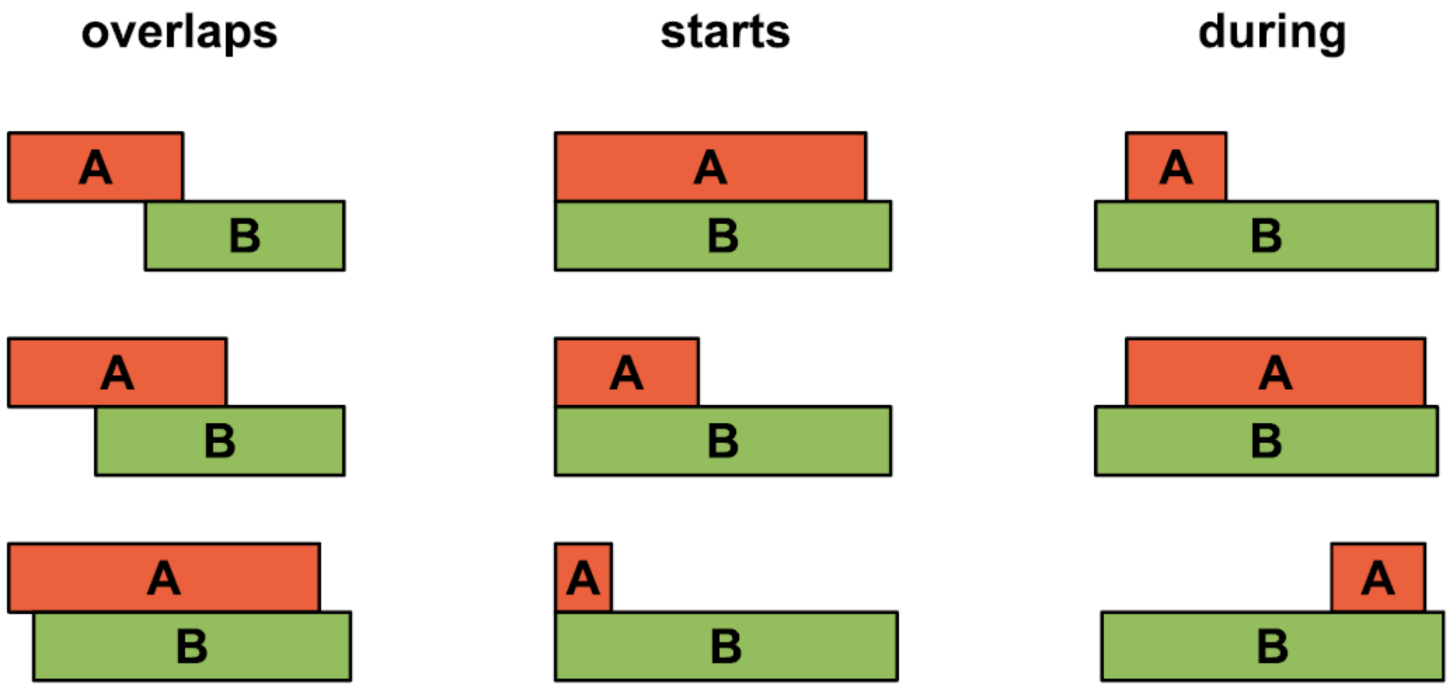
Small changes can result in different relations that intuitively describe the same pattern.

Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

Disadvantages of Allen's relation

SIEMENS

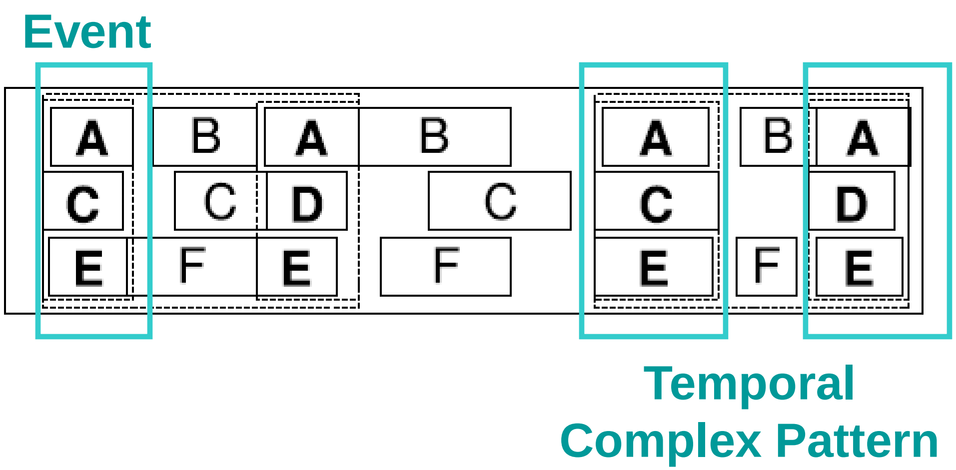
Allen is ambiguous



The same relation describes intuitively different patterns.

Unification-based Temporal Grammar

SIEMENS

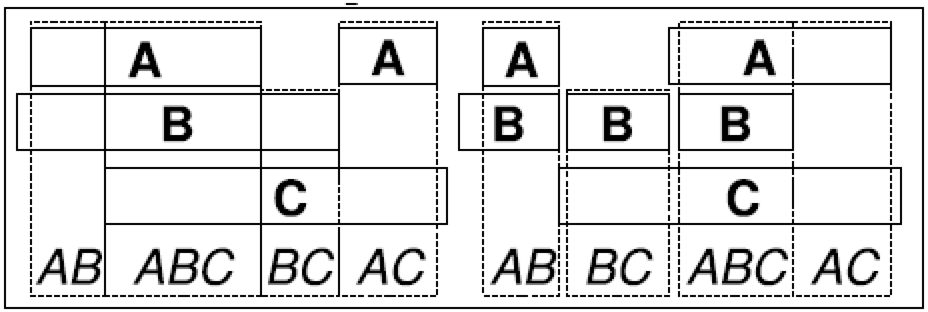


Unification-based Temporal Grammar (UTG) [Ultsch 1996]

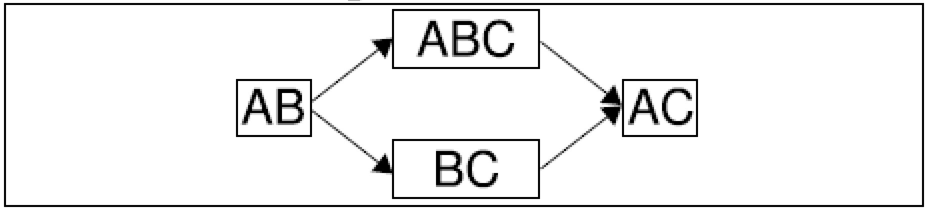
- Events: several intervals occur **more or less simultaneous**.
- Temporal Complex Patterns: sequence of Events.
- Annotations for duration of intervals and gaps.
- Detection of patterns can be formulated Prolog (hence unification-based)

Time Series Knowledge Representation

SIEMENS



Chords



Phrase

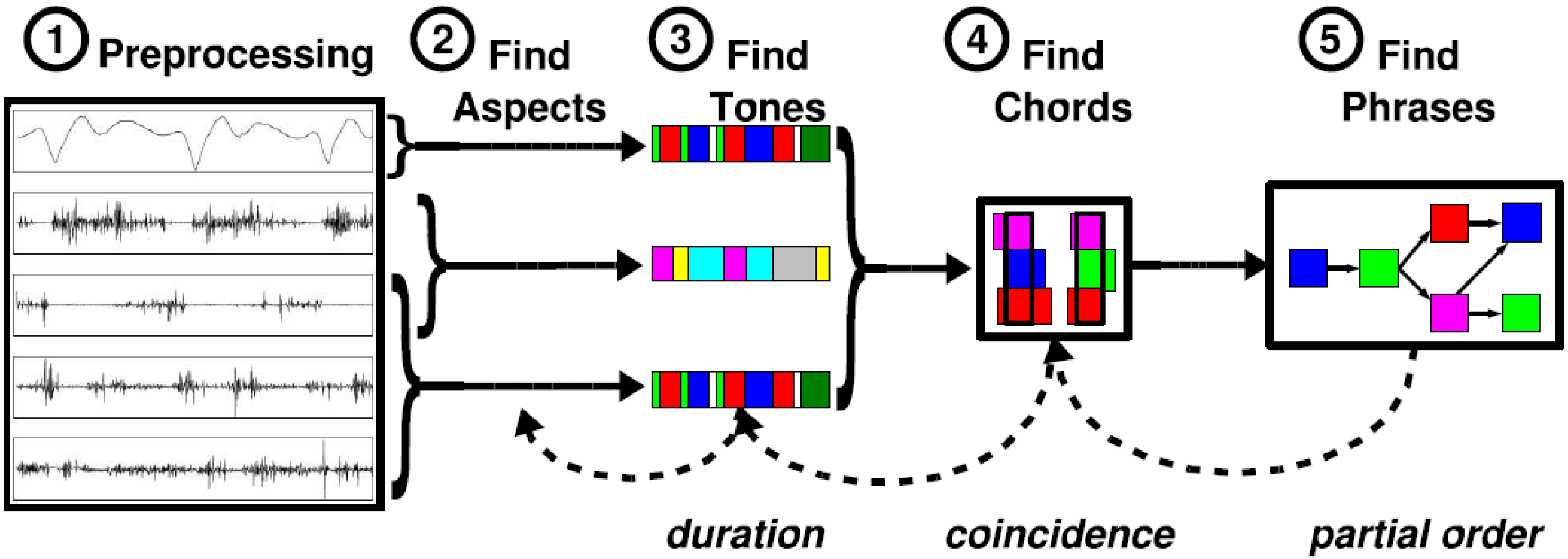
Time Series Knowledge Representation (TSKR) [Moerchen 2006]

- Extension of UTG
- Tones represent **duration** with intervals.
- Chords represent **coincidence** of Tones.
- Phrases represent **partial order** of Chords
- Compact, unambiguous representation with details on demand.
- Robust against noise in the interval boundaries.

TSKR mining algorithms

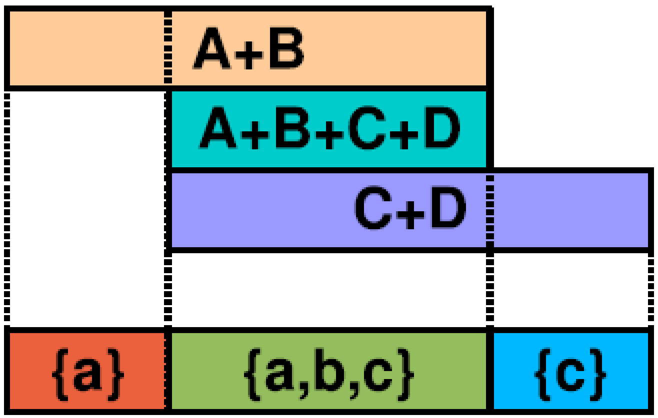
SIEMENS

- **Time Series Knowledge Mining** [Moerchen 2006]
  - Methodology for mining from multivariate time series.
  - Tone mining: discretization, segmentation, clustering.
  - **Chord mining**: variation of itemset mining.
  - **Phrase mining**: variation of partial order mining.



Time interval patterns
Phrase mining algorithm

SIEMENS



where

a = A+B

b = A+B+C+D

a = C+D

Phrase mining

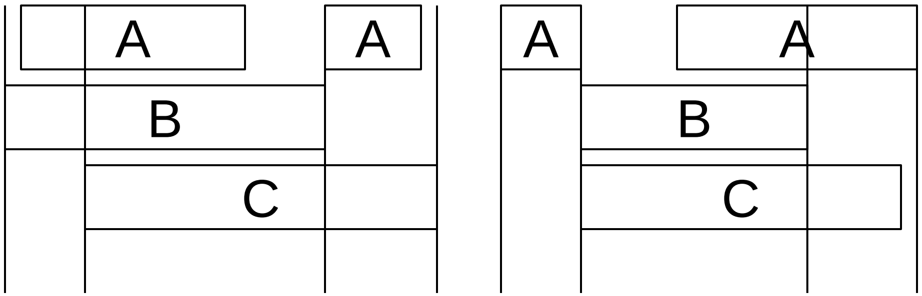
- Transform Chords to itemset sequence: set of Chords observed over time interval.
- Mine sequential patterns with additional constraints: Only one Chord can be picked per time interval.
- Mine closed groups of sequential patterns.
- Convert to partial order.

Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

Time interval patterns

Templates for (partial) presence/absence

SIEMENS



ed	ed	ed
pe	pe	ad
ad	pe	pe

$\cap$

pe	ed	pe
ad	pe	ad
ad	pe	ed

=

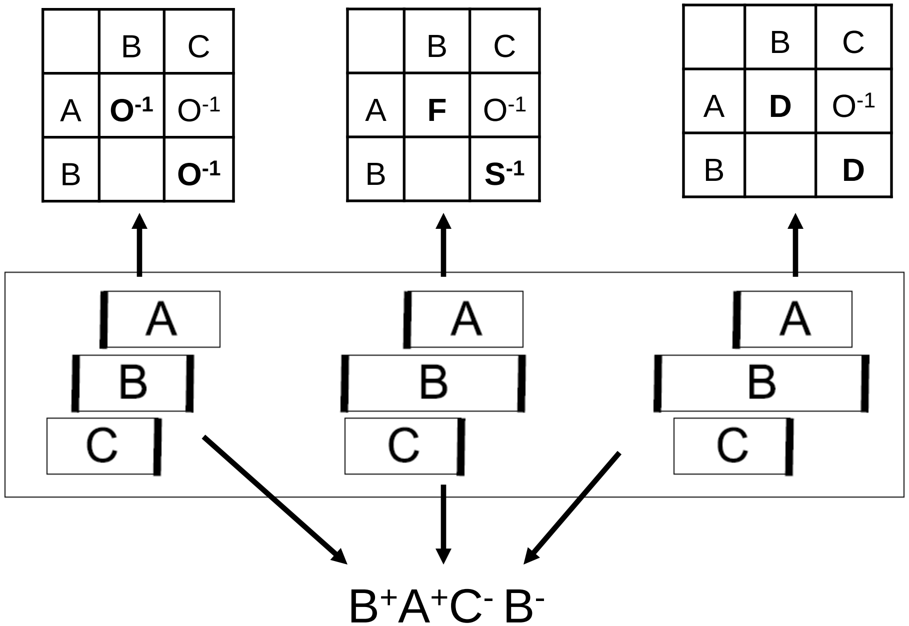
e	e	e
*	p	a
a	p	e

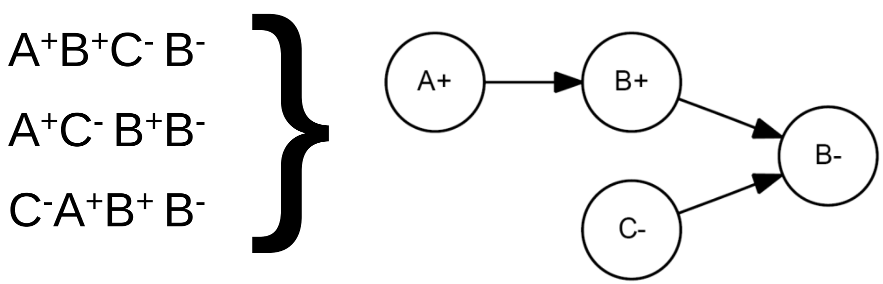
Templates [Peter/Hoeppner 2010]

- Segment time axis
- 5 Interval predicates
  - present ($\forall$)
  - absent ($\forall\neg$)
  - unconstrained (*)
  - exists ($\exists$)
  - disappears ($\exists\neg$)
- Algorithm to search classification rules (not frequent patterns)
- Picks time segments and captures constraints on duration

SIEMENS

Semi-interval Partial Order (SIPO) patterns





Semi-Interval Partial Order (SIPO)
[Moerchen/Fradkin 2010]

- Sequential pattern of interval boundaries **without constraints**.
  - Superset of full interval (Allen) patterns.
  - Matches more similar situations in the data.
- **Partial order patterns** of interval boundaries.
  - Even less constraints.
  - Even more matches.
- Can be applied to **mixed time point / time interval** data!

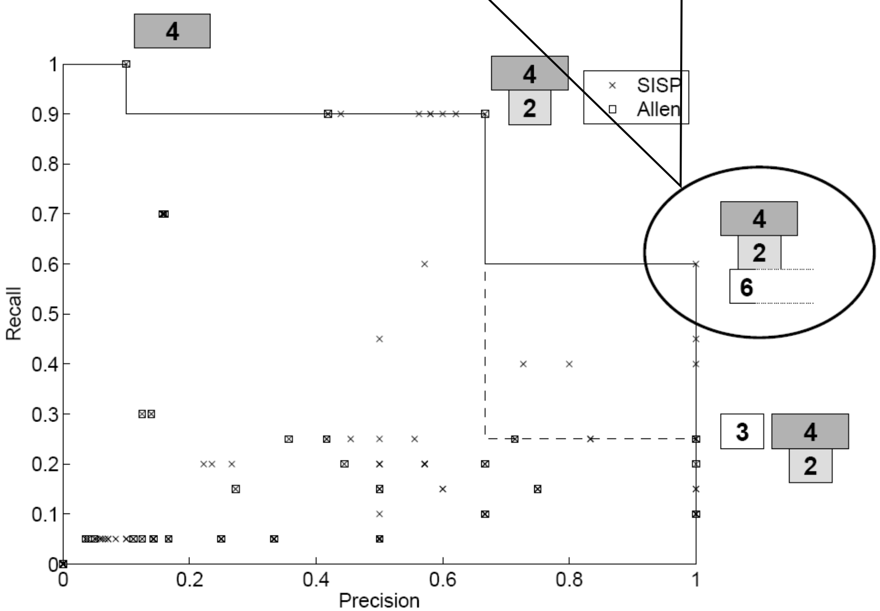
Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

SIEMENS

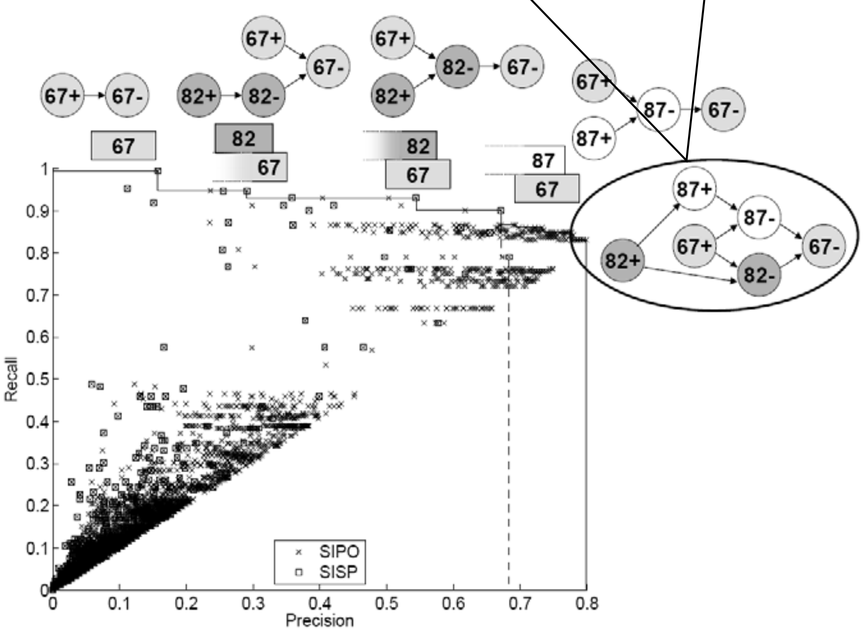
Semi-interval patterns for classification

Semi interval sequential pattern (SISP) with **better precision/recall than any full interval pattern**.

Semi interval partial order (SIPO) pattern with **better precision/recall than any SISP or full interval pattern.**



'Name' class in Australian Sign language dataset.



'I' class in American Sign language dataset.

Fabian Moerchen, Knowledge & Decision Systems, Siemens Corporate Research, Princeton, NJ

SIEMENS

Summary

- Use **Allen** only if exact interval boundaries matter and Allen semantics are required.
- **SIPO**: superset of Allen patterns allowing for missing boundaries (Freksa) and partial order.
- **TSKR** and **Templates**: alternative approaches that match partial intervals.

	Univariate symbolic time series	Multivariate symbolic time series	Symbolic time sequence	duration	order	coincidence	synchronicity	partial order
Allen		✓	✓	✓	✓	✓	✓	
UTG		✓	✓	✓	✓		✓	
TSKR		✓	✓	✓	✓	✓	✓	✓
Templates		✓	✓	✓	✓	✓		
SIPO		✓	✓	✓	✓	✓	✓	✓

SIEMENS

Summary

- Introduced **data models, patterns, and algorithms** for point and interval data.
- Online version: additional material on **preprocessing** to convert other data into symbolic temporal data and **applications** using temporal patterns for data mining.

Research opportunities

- **Algorithms**: efficiency, error tolerant patterns, data streams,…
- **Interestingness**: pattern significance, expert knowledge, …
- **Classification**: directly mine patterns predict events (early).
- **Anomaly detection**: learn to distinguish normal from abnormal behavior for symbolic temporal data based on patterns.
- **Applications** in medicine, finance, maintenance, meteorology, …

SIEMENS

Thank you!

Please send any feedback to **fabian.moerchen@siemens.com**

www.siemens.com

www.usa.siemens.com/en/about_us/research/home.htm

More material:

www.timeseriesknowledgemining.org (bibliography, slides, etc.)

Fabian Moerchen: **Unsupervised pattern mining from symbolic temporal data**, SIGKDD Explorations 9(1), ACM, pp. 41-55, 2007