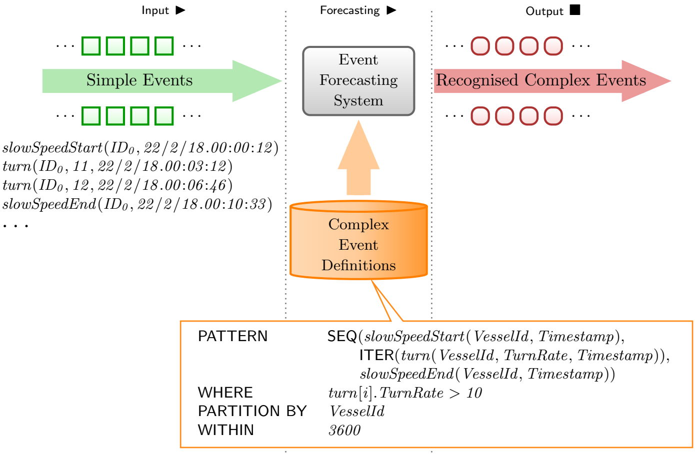
# Can Computers Understand what is Happening? Complex Event Forecasting

Alexander Artikis[1,2] and Periklis Mantenoglou[1,3]
*Slides: Elias Alevizos*[1]

[1]NCSR Demokritos, Athens, Greece
[2]University of Piraeus, Greece
[3]NKUA, Greece

https://cer.iit.demokritos.gr

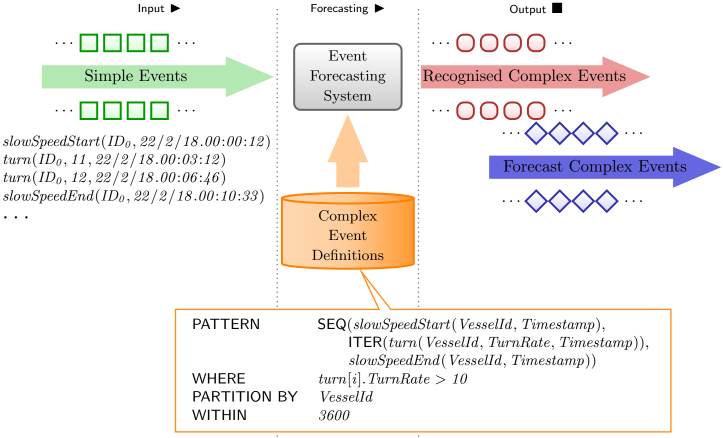
# Complex Event Forecasting (CEF)*,†




* Giatrakos et al, Complex Event Recognition in the Big Data Era: A Survey. VLDB Journal, 2020.

† Alevizos et al, Probabilistic Complex Event Recognition: A Survey. ACM Computing Surveys, 2017.

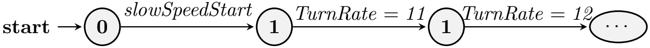
# Complex Event Forecasting (CEF)[*,†]




[*] Giatrakos et al, Complex Event Recognition in the Big Data Era: A Survey. VLDB Journal, 2020.

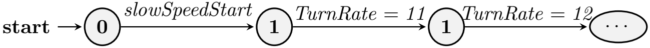
[†] Alevizos et al, Probabilistic Complex Event Recognition: A Survey. ACM Computing Surveys, 2017.

# Complex Event Forecasting (CEF)



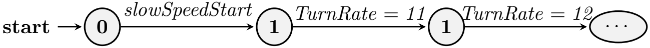

- Is this a fishing manoeuvre?
- How long will it last?
- With what probability?

# Complex Event Forecasting (CEF)




- Is this a fishing manoeuvre?
- How long will it last?
- With what probability?
- Given a stream of events $S$ and a pattern $R$,
  - Recognition: Find the 'full matches' of $R$ in $S$.

# Complex Event Forecasting (CEF)




- Is this a fishing manoeuvre?
- How long will it last?
- With what probability?
- Given a stream of events $S$ and a pattern $R$,
  - Recognition: Find the 'full matches' of $R$ in $S$.
  - Forecast the full matches.

# Wayeb: A Complex Event Forecasting system*


https://cer.iit.demokritos.gr (forecasting)

*https://github.com/ElAlev/Wayeb

# Complex Event Forecasting vs Machine Learning

- In Machine Learning, the goal is to predict the output of a function on previously unseen input data.
  - The input data need not necessarily have a temporal dimension.

# Complex Event Forecasting vs Machine Learning

- In Machine Learning, the goal is to predict the output of a function on previously unseen input data.
  - The input data need not necessarily have a temporal dimension.
- In Complex Event Forecasting, the task is to predict the temporally future output of a function, ie the future occurrence of an event.
  - Time is thus a crucial component.
  - From the (current) timepoint where a forecast is produced until the (future) timepoint for which we try to make a forecast, no data is available.

# Complex Event Forecasting vs Machine Learning

We have Deep Learning and it seems to work. Can we go home?

Complex event forecasting:

- Formal semantics* for trustworthy models.
- Explanation — why did we forecast a complex event?
- Machine Learning is necessary. But:
  - Complex events are rare.
  - Supervision is scarce.
- More often than not, background knowledge is available — let's use it!

* Grez et al, A Formal Framework for Complex Event Recognition. ACM Transactions on Database Systems, 2021.

# Automata-based Complex Event Forecasting*

- Patterns defined as regular expressions.

*Alevizos et al, Event Forecasting with Pattern Markov Chains. Distributed and Event-based Systems (DEBS), 2017.

# Automata-based Complex Event Forecasting*

- Patterns defined as regular expressions.
- Consume streams of events and forecast when a pattern is expected to be fully matched.

*Alevizos et al, Event Forecasting with Pattern Markov Chains. Distributed and Event-based Systems (DEBS), 2017.

# Automata-based Complex Event Forecasting*

- Patterns defined as regular expressions.
- Consume streams of events and forecast when a pattern is expected to be fully matched.
- Revise forecasts to reflect changes in the 'state' of the pattern.

* Alevizos et al, Event Forecasting with Pattern Markov Chains. Distributed and Event-based Systems (DEBS), 2017.

# Approach overview

- Given a stream of events $S$ and a pattern $R$:
  - Convert $R$ into an automaton $A$.

# Approach overview

- Given a stream of events $S$ and a pattern $R$:
  - Convert $R$ into an automaton $A$.
  - Feed $S$ to $A$.

# Approach overview

- Given a stream of events $S$ and a pattern $R$:
  - Convert $R$ into an automaton $A$.
  - Feed $S$ to $A$.
  - For every (non-final) state $A$ finds itself in, estimate the probability of reaching its final state within some window $w$.

# Approach overview

- Given a stream of events $S$ and a pattern $R$:
  - Convert $R$ into an automaton $A$.
  - Feed $S$ to $A$.
  - For every (non-final) state $A$ finds itself in, estimate the probability of reaching its final state within some window $w$.
  - If the model is confident enough, emit a forecast.

# Approach overview

- Given a stream of events $S$ and a pattern $R$:
  - Convert $R$ into an automaton $A$.
  - Feed $S$ to $A$.
  - For every (non-final) state $A$ finds itself in, estimate the probability of reaching its final state within some window $w$.
  - If the model is confident enough, emit a forecast.
  - We may also emit a future interval within which a complex event is expected.

# Probability Estimation

- Convert pattern automata into Markov chains to produce a probabilistic model for the automaton's behaviour.

# Probability Estimation

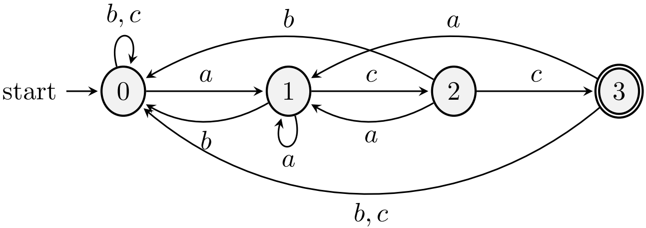
- Convert pattern automata into Markov chains to produce a probabilistic model for the automaton's behaviour.
- How long (number of transitions) will we have to wait before reaching the final state?
  - Calculate the waiting-time distribution for each state.

# Probability Estimation

- Convert pattern automata into Markov chains to produce a probabilistic model for the automaton's behaviour.
- How long (number of transitions) will we have to wait before reaching the final state?
  - Calculate the waiting-time distribution for each state.
- Scan the waiting-time distributions to produce forecasts.

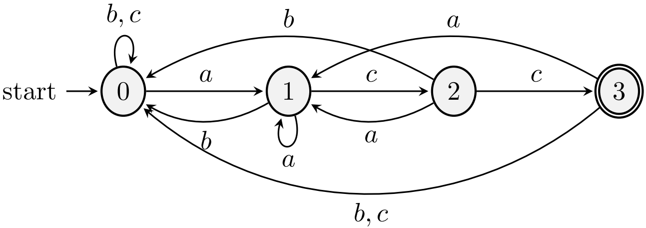
# Regular Expression → Pattern Markov Chain

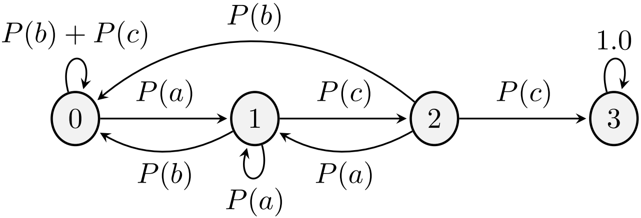
$R = a \cdot c \cdot c.$ $\Sigma = \{a, b, c\}.$ No memory.



# Regular Expression $\rightarrow$ Pattern Markov Chain

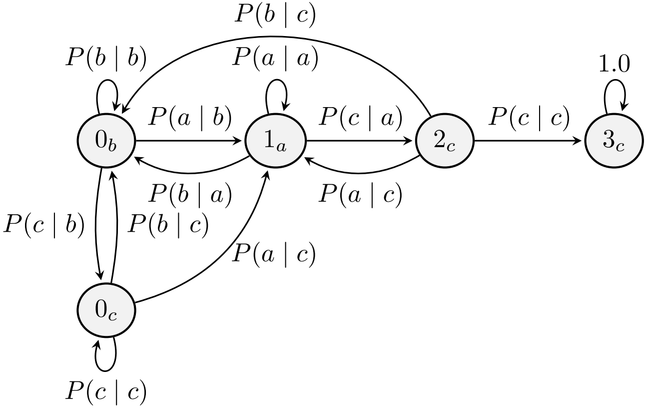
$R = a \cdot c \cdot c.$ $\Sigma = \{a, b, c\}.$ No memory.



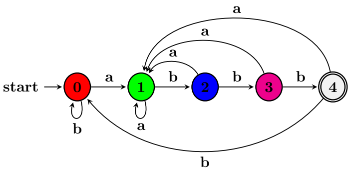
# Regular Expression $\rightarrow$ Pattern Markov Chain

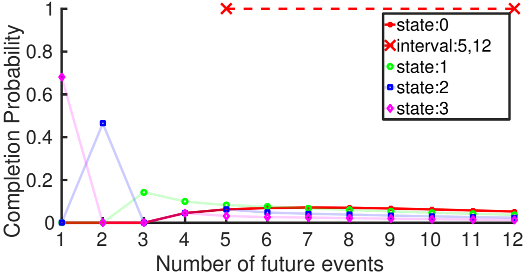
$R = a \cdot c \cdot c. \qquad \Sigma = \{a, b, c\}. \qquad m = 1.$



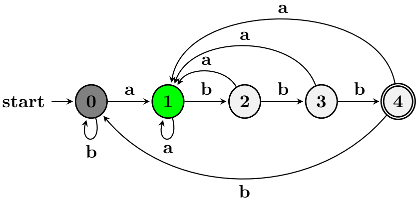
# Waiting-Time and Forecasts

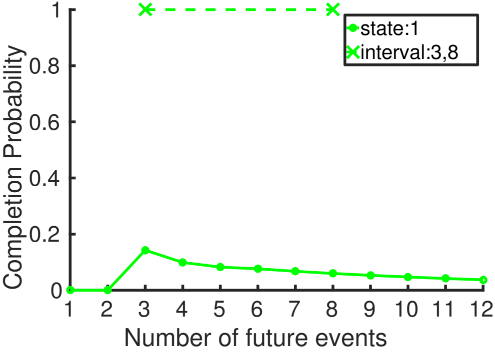
- Assume $R := a \cdot b \cdot b \cdot b$; $\Sigma = \{a, b\}$; and no memory.



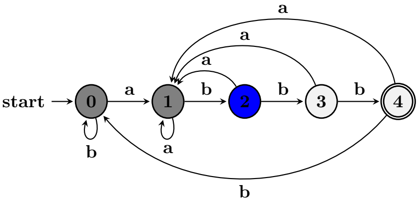



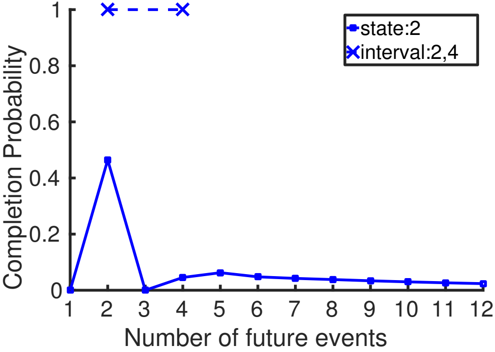
# Example: $R = a \cdot b \cdot b \cdot b$.



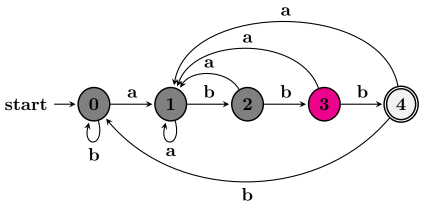



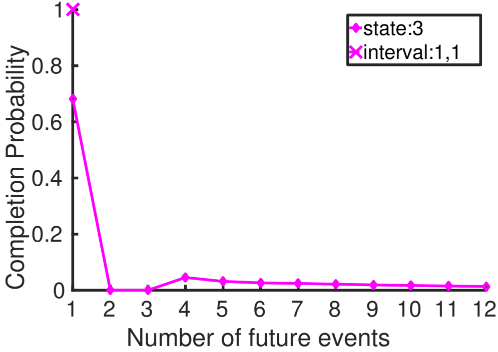
# Example: $R = a \cdot b \cdot b \cdot b$.



# Example: $R = a \cdot b \cdot b \cdot b$.



# Symbolic Automata & Variable-order Markov Models*

- Use symbolic automata.
  - Infinite alphabet.
  - Closed under concatenation, intersection, union, Kleene-star and complement; determinisable.

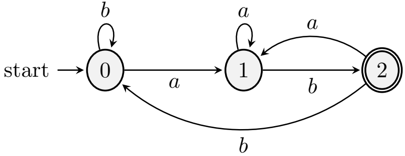
*Alevizos et al, Complex Event Forecasting with Prediction Suffix Trees. VLDB Journal, 2022.

# Symbolic Automata & Variable-order Markov Models*

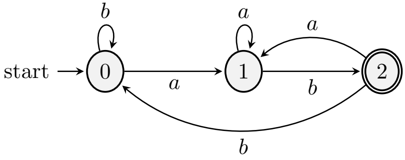
- Use symbolic automata.
  - Infinite alphabet.
  - Closed under concatenation, intersection, union, Kleene-star and complement; determinisable.
- Use a variable-order Markov model.
  - Prediction Suffix Trees.
  - Focus on the statistically significant properties of the stream.

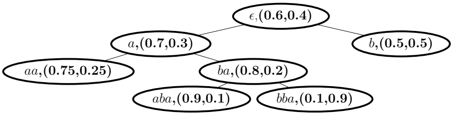
* Alevizos et al, Complex Event Forecasting with Prediction Suffix Trees. VLDB Journal, 2022.

# Complex Event Forecasting with Prediction Suffix Trees

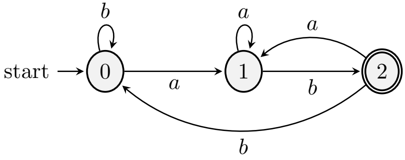


# Complex Event Forecasting with Prediction Suffix Trees

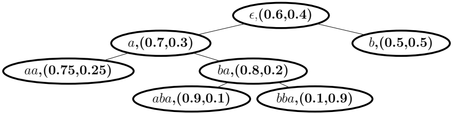


# Complex Event Forecasting with Prediction Suffix Trees

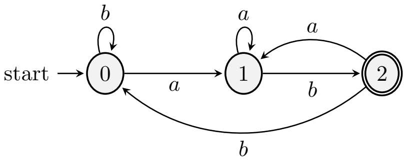


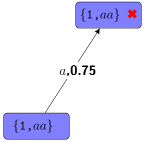

$\{1, aa\}$

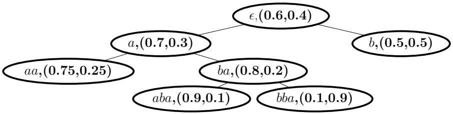



- Let's compute the waiting-time distributions.
- Assume automaton in state 1 and suffix is *aa*.
- Estimate distribution step by step, for $k = 1$, $k = 2$, etc.

# Complex Event Forecasting with Prediction Suffix Trees

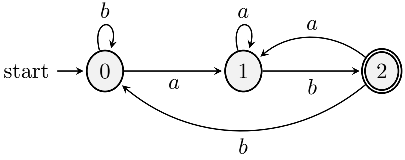


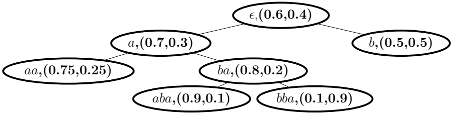



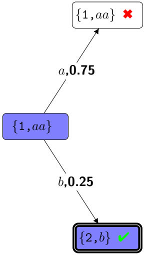




- $k = 1$.
- Assume another *a* arrives. Suffix still *aa*, automaton again in state 1.
- Reaches a non-final state. Not interested.

# Complex Event Forecasting with Prediction Suffix Trees

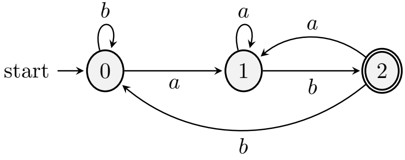


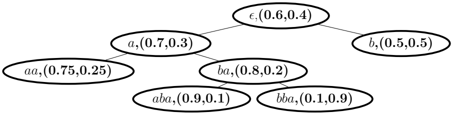



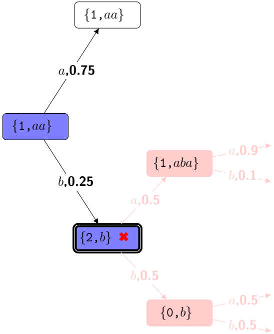




- $k = 1$.
- Assume $b$ arrives first. Suffix is $b$, automaton in state 2.
- Reaches a final state. No other options. Thus $P(W_{\{1,aa\}} = 1) = 0.25$.

# Complex Event Forecasting with Prediction Suffix Trees

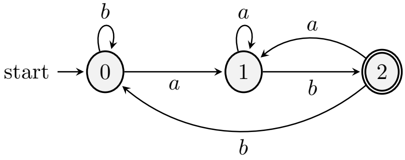


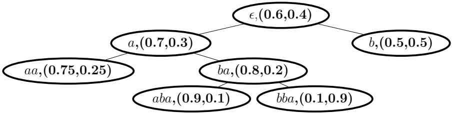



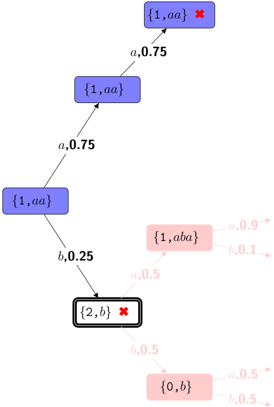




- $k = 2$.
- Assume $b$ arrives first. Suffix is $b$, automaton in state 2.
- Reaches a final state already at $k = 1$. All downstream paths pruned.

# Complex Event Forecasting with Prediction Suffix Trees

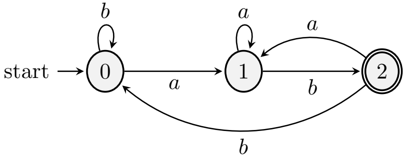


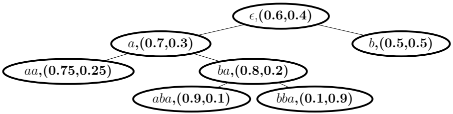



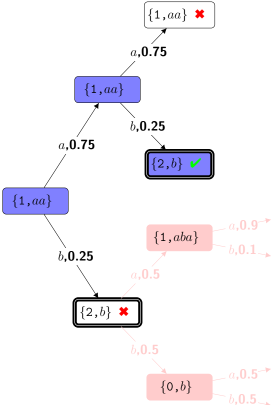




- $k = 2$.
- Assume *a* arrives first. Then another *a*.
- Reaches a non-final state. Not interested.

# Complex Event Forecasting with Prediction Suffix Trees




- $k = 2$.
- Assume $a$ arrives first. Then $b$.
- Reaches a final state at $k = 2$. Only path leading to a final at $k = 2$. Thus $P(W_{\{1,aa\}} = 2) = 0.75 * 0.25 = 0.1875$.

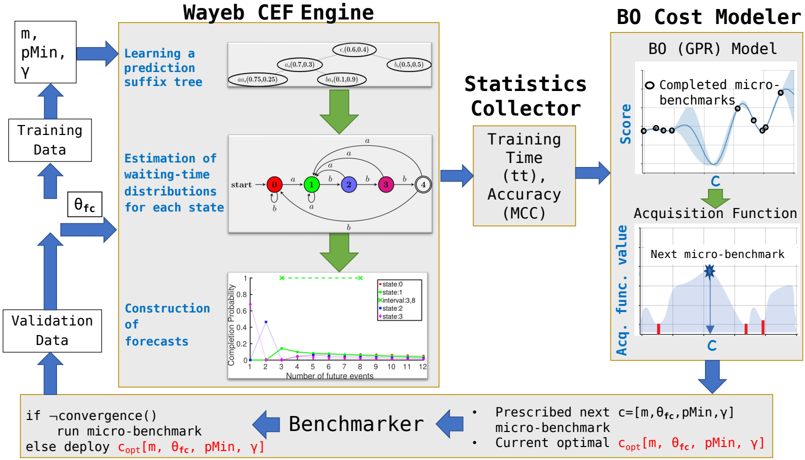
# Hyper-paremeter tuning for Complex Event Forecasting

- Complex Event Forecasting systems have hyper-parameters, eg model order, confidence threshold, etc.
- Finding the proper values through exhaustive search may be sub-optimal.

# Hyper-paremeter tuning for Complex Event Forecasting

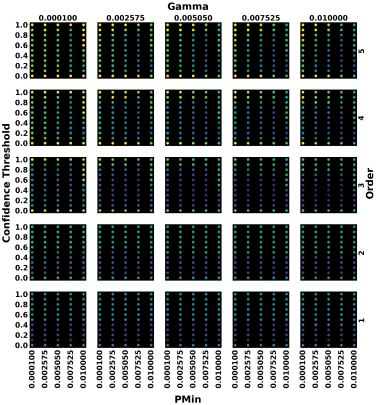
- Complex Event Forecasting systems have hyper-parameters, eg model order, confidence threshold, etc.
- Finding the proper values through exhaustive search may be sub-optimal.
- Efficient search of the hyper-parameter space with Bayesian Optimization.

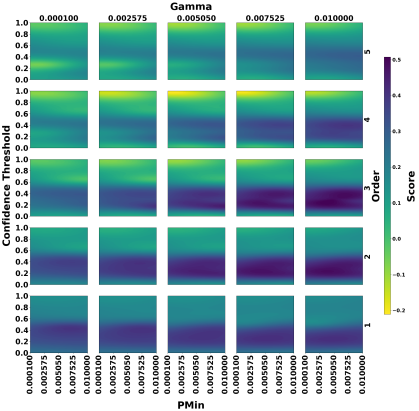
# Bayesian Optimization for Complex Event Forecasting*




*Stavropoulos et al, Optimizing complex event forecasting. Distributed and Event-based Systems (DEBS), 2022.

# Exhaustive search vs Bayesian Optimization



# Summary

Complex event forecasting:

- Symbolic automata for complex event patterns
  - Closure properties.
  - Formal compositional semantics.

# Summary

Complex event forecasting:

- Symbolic automata for complex event patterns
  - Closure properties.
  - Formal compositional semantics.
- Prediction suffix trees for long-term dependencies
  - Higher accuracy.
  - Comparable training time and acceptable throughput.

# Summary

Complex event forecasting:

- Symbolic automata for complex event patterns
  - Closure properties.
  - Formal compositional semantics.
- Prediction suffix trees for long-term dependencies
  - Higher accuracy.
  - Comparable training time and acceptable throughput.

Current work:

- CEF with Symbolic Register Automata*:
  - Symbolic automata with 'memory'.
  - Express *n*-ary relations between events.

*Alevizos et al, Complex Event Recognition with Symbolic Register Transducers. VLDB, 2024.