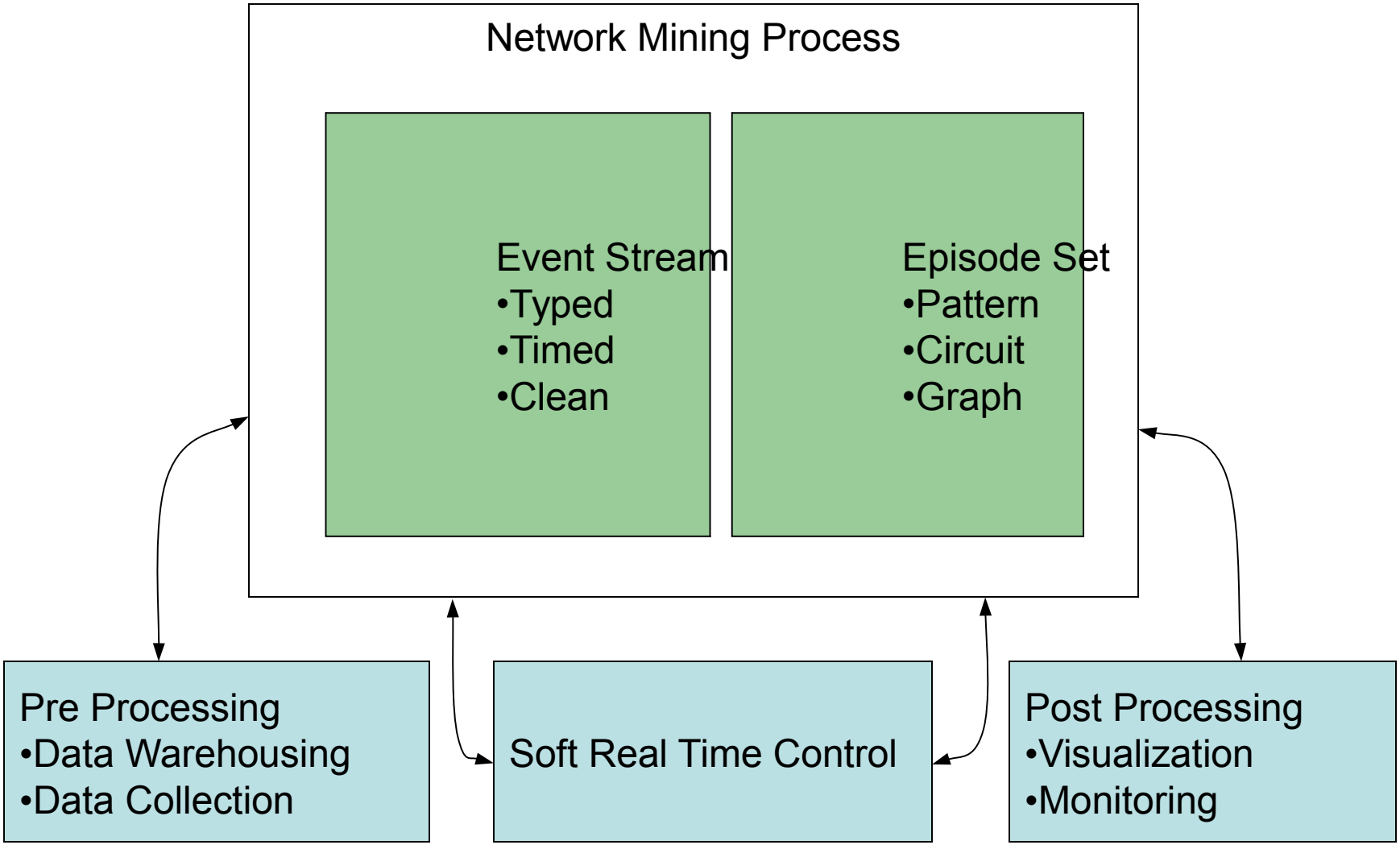
# Collaborative Network Mining

Discovering patterns in sequential event streams using distributed processing

# Network Mining Components

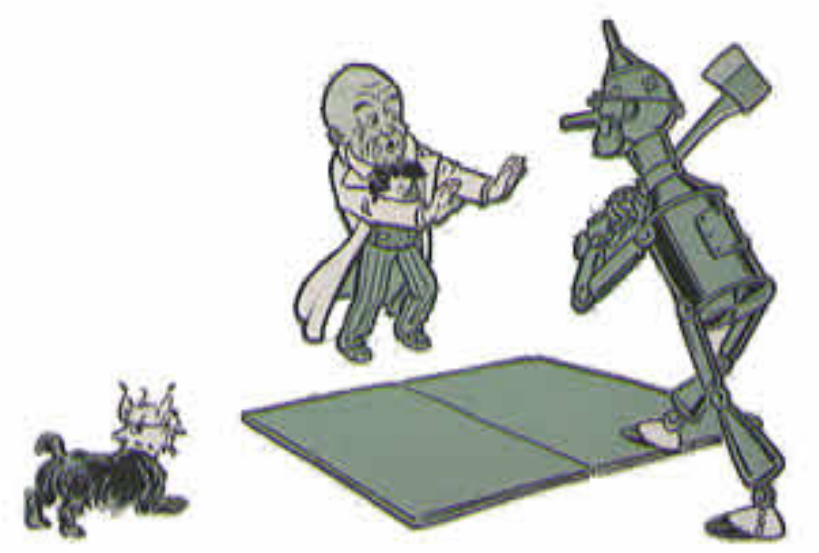


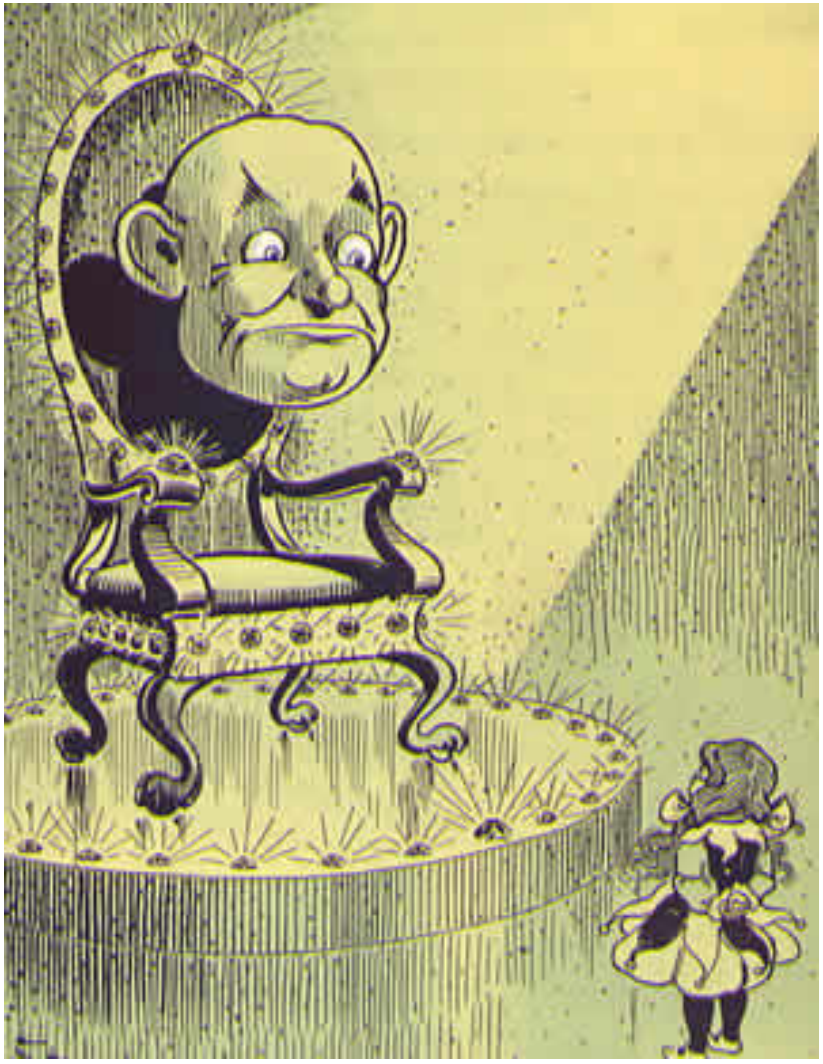
# An Ancient Problem

- To see reality when all we have is perception
- This is prospecting

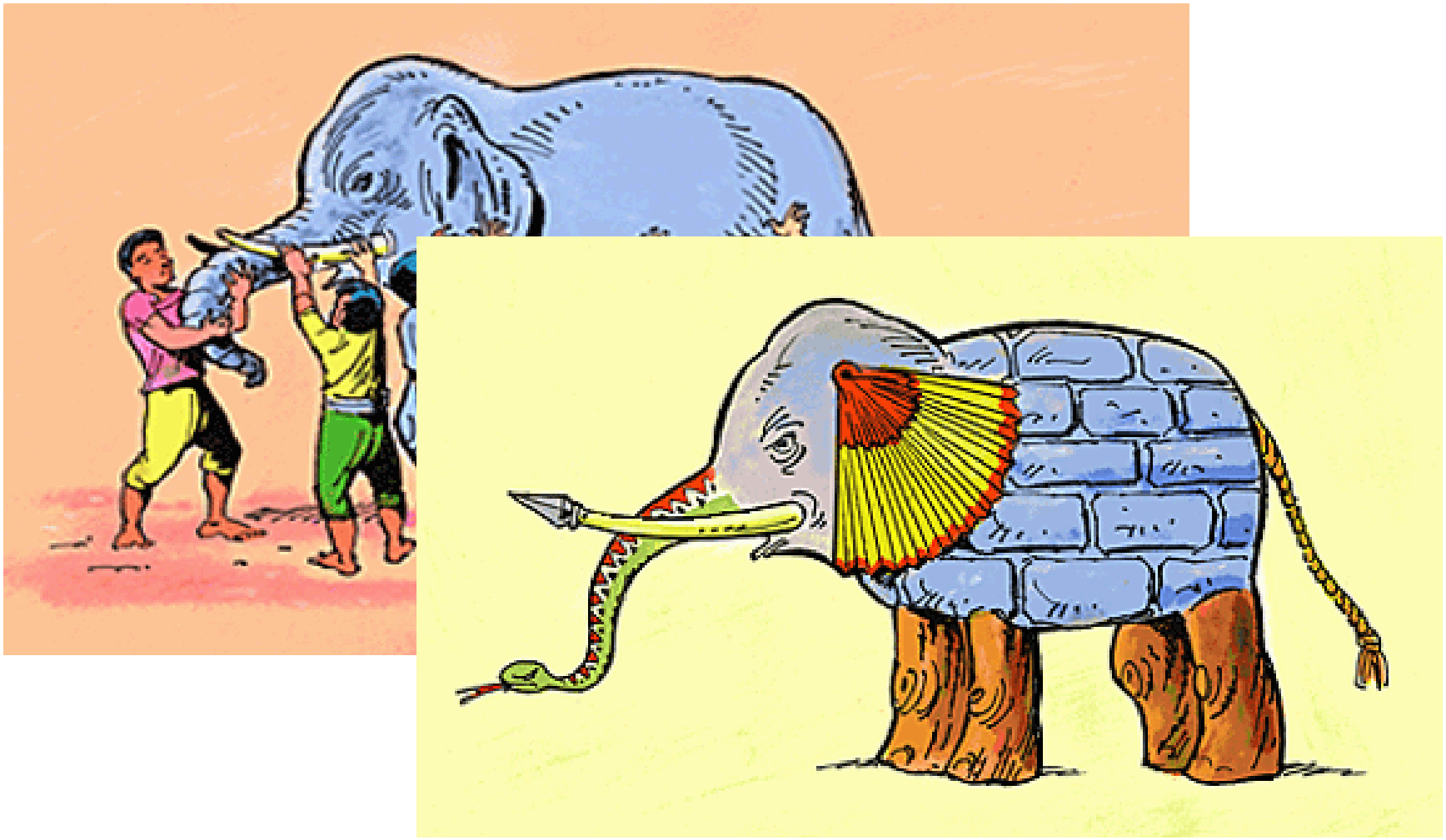
# An Ancient Problem

- To see reality when all we have is perception
- “Perception is reality”?

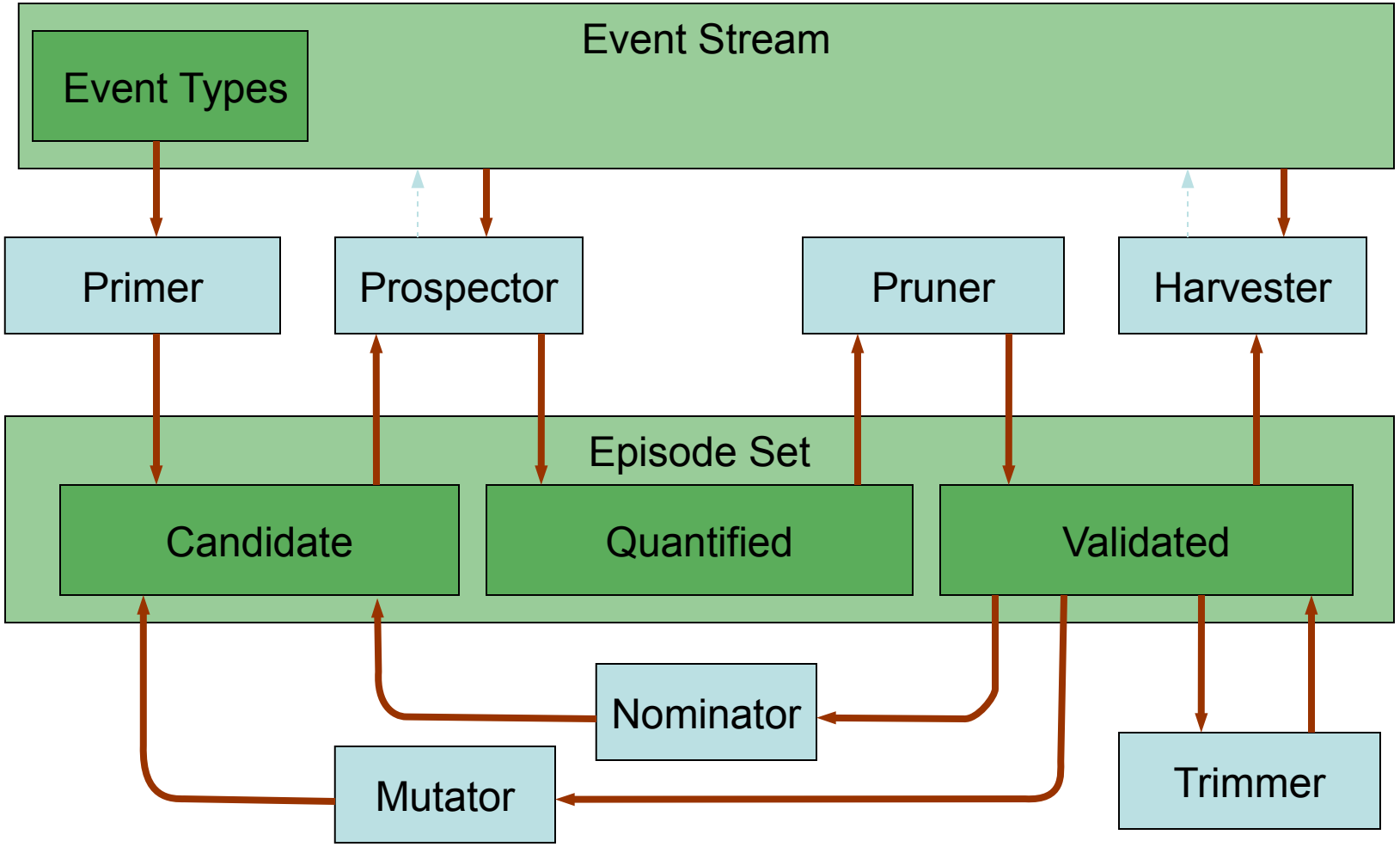




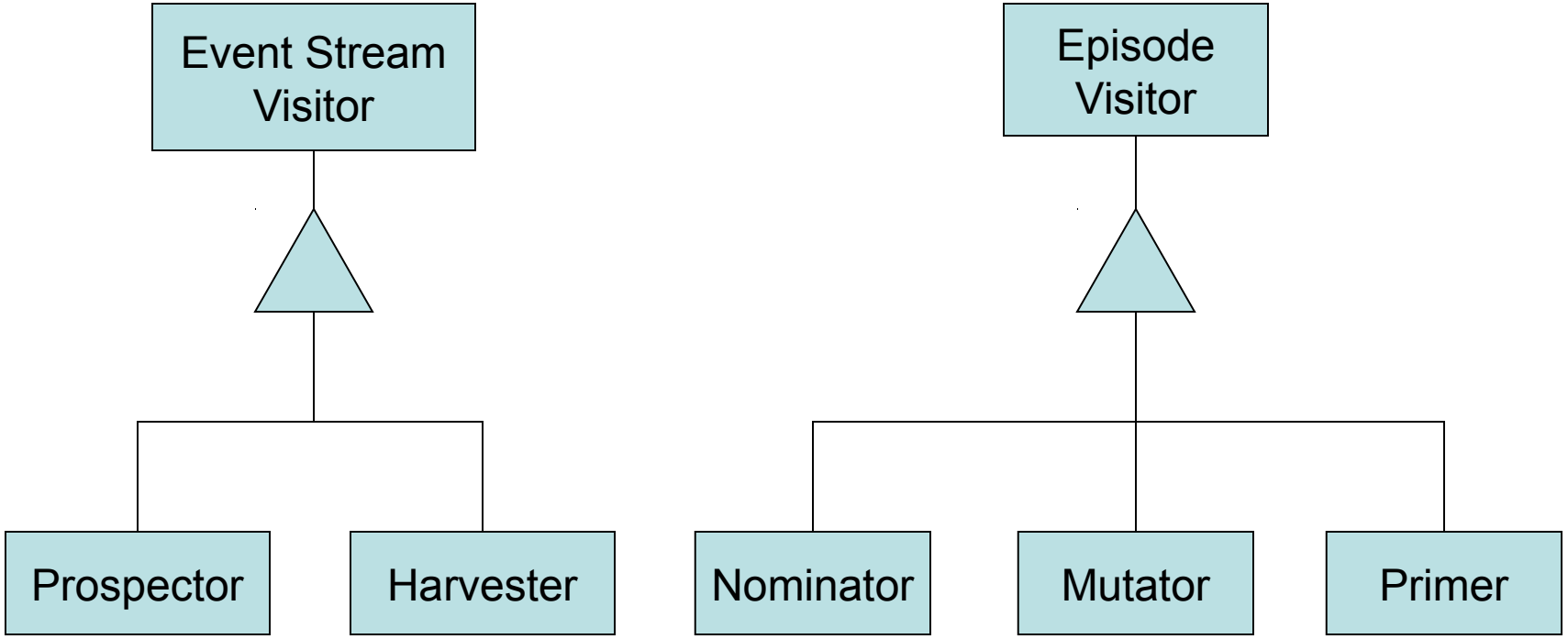
# Perception v. Reality



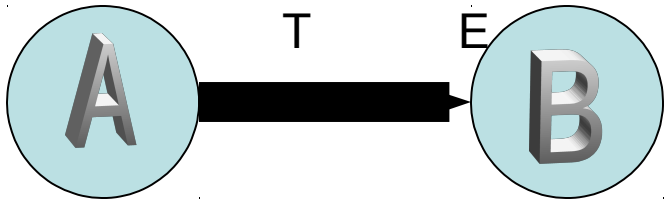
# Network Mining Process



# Type Relationships

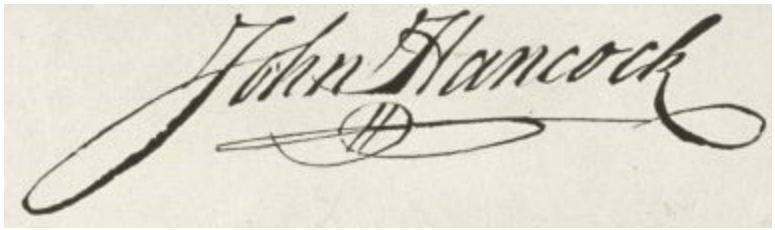


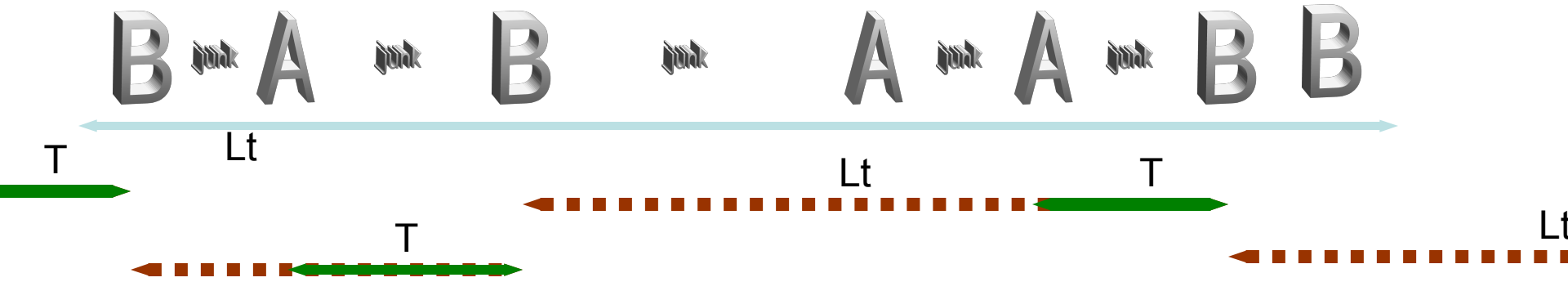
# The Key to Graphs




The non-overlapping episode

- Signature: (A [T:E] B)
- Total Length: Lt



# Total Non-overlapped Length

1. Develop a no connection formulation
    1. Expected value (first moment)
    2. Variation (second moment)
2. Develop a connected formulation
    1. Expected value (first moment)
    2. Variation (second moment)
    - 

# Chain Motif

If there is no connection between A and B

$$\left\langle \tilde{L}_{AB} \right\rangle = \frac{1}{\Pr(A)} + \frac{1}{\Pr(B)};$$

If there is a connection between A and B, expressed as a conditional probability. The probability that A causes B.

$$\gamma = \Pr(A \Rightarrow B) = \Pr(B \mid A) - \Pr(B);$$

$$\left\langle \overline{L}_{AB} \right\rangle = (1-\gamma)\left[\left\langle \tilde{L}_{AB} \right\rangle\right] + \gamma\left[\frac{1}{\Pr(A)} + \frac{1}{\gamma}\right];$$

$$\left\langle \overline{L}_{AB} \right\rangle = 2 + \frac{1}{\Pr(A)} + \frac{1}{\Pr(B)} - \frac{\Pr(A \cap B)}{\Pr(A)\Pr(B)};$$

# Chained Interval Motif

The chained interval is an episode where there is an occurrence of A followed by an occurrence of B after an interval of time T.

If there is no connection between A and B.
How many times will A be repeated before a B appears in time T?

$$\Pr\left(A \xrightarrow{T} B\right) = 1 - e^{-\frac{1}{\Pr(B)}x}\Big|_{T0}^{T1} = e^{-\frac{T0}{\Pr(B)}} - e^{-\frac{T1}{\Pr(B)}};$$

$$\left\langle \tilde{L}_{AB} \right\rangle = \frac{1}{\Pr(A)\Pr(A \xrightarrow{T} B)} + T;$$

If there is a connection between A and B with a delay T.

$$\gamma = \Pr\left(A \Rightarrow B\right) = \frac{1}{\lambda_{AB}};$$

$$\left\langle \overline{L}_{AB} \right\rangle = (1-\gamma)\left[\left\langle \tilde{L}_{AB} \right\rangle\right] + \gamma\left[\frac{1}{\Pr(A)} + T\right];$$

# Branching Interval Motif

The branch interval is an episode where there is an occurrence of A followed by an occurrence of B and an occurrence of C after intervals of time Tb and Tc, respectively.

There is no connection between A and B.

How many times will A be repeated before a B appears in time T?

$$\beta = \Pr(\frac{1}{\Pr(B)} = T)?;$$

$$\tilde{L}_{AB} = \frac{\beta}{\Pr(A)} + T;$$

There is a connection between A and B, expressed as a conditional probability.

$$\gamma = \Pr(B \mid A) - P(B);$$

$$\overline{L}_{AB} = (1-\gamma)\left[\tilde{L}_{AB}\right] + \gamma\left[\frac{1}{\Pr(A)} + \frac{1}{\gamma}\right];$$

# Joining Interval Motif

The join interval is an episode where there is an occurrence of A and an occurrence of B both followed by the same occurrence of C after intervals of time Tac and Tbc, respectively.

There is no connection between A and B.
How many times will A be repeated before a B appears in time T?

$$\beta = \Pr(\frac{1}{\Pr(B)} \geq T)?;$$

$$\tilde{L}_{AB} = \frac{\beta}{\Pr(A)} + T;$$

There is a connection between A and B, expressed as a conditional probability.

$$\gamma = \Pr(B \mid A) - P(B);$$

$$\overline{L}_{AB} = (1-\gamma)\left[\tilde{L}_{AB}\right] + \gamma\left[\frac{1}{\Pr(A)} + \frac{1}{\gamma}\right];$$

# Porting Components

- Define episode set structure in candidate environment
- Provide a typical episode prospector as prototypical C function
- Provide a corresponding nominator as an environment native function
- Provide a simple threshold pruner as an environment native function

# Demonstration Problem

- Specify the demonstration problem
- Demonstrate the use of the components against the problem

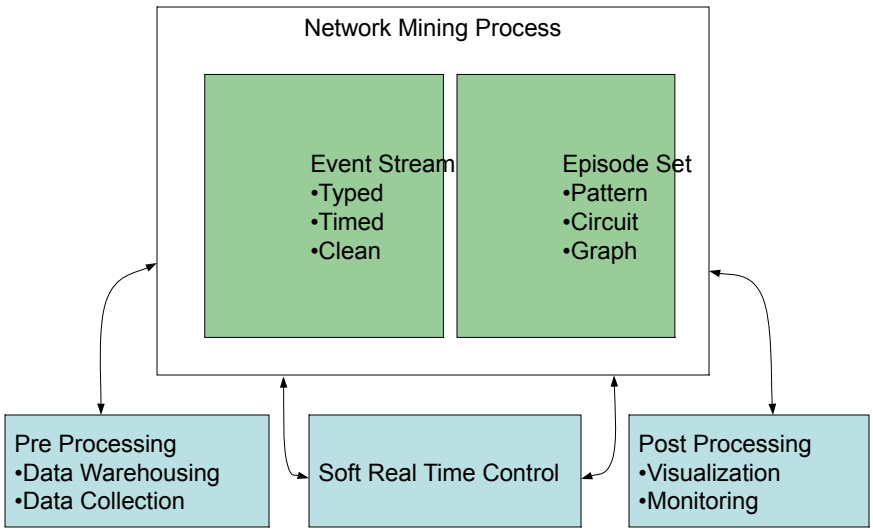
# Rationalization

- Examine/evaluate current (sub) frameworks in the target environment as possible integration points.

# Collaborative Network Mining

Discovering patterns in sequential event streams using distributed processing

# Network Mining Components




Why Prospecting and not Mining?

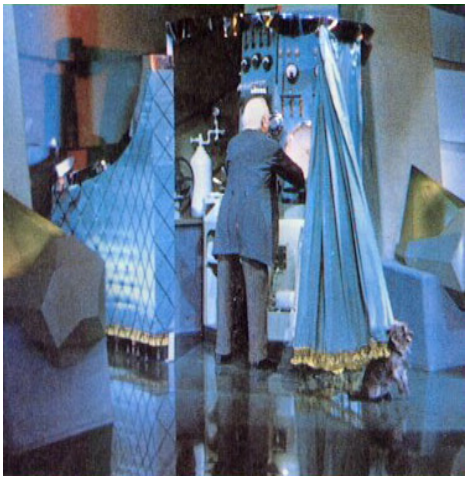
Mining is the extraction of specific resources from a place containing a deposit.

Prospecting is locating the place showing signs of containing a deposit.

Soft real time systems

# An Ancient Problem

- To see reality when all we have is perception
- This is prospecting



This is prospecting to identify the actor (and by extension the directory, script writer, etc.) when all we have is the play.

A miner is interested in extracting a known resource from a known location is large quantities, cheaply.

A prospector is interested in surveying large tracts of land to identify the locations of known resources, cheaply.

Clearly prospecting will extract some quantity of a resource but that is not its primary goal

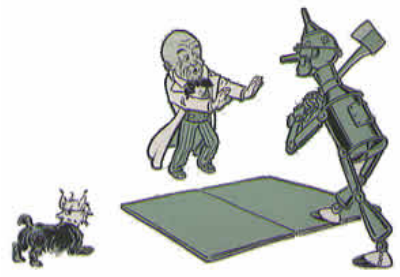
Whereas for Mining resource, episode instance, extraction is exactly the goal.

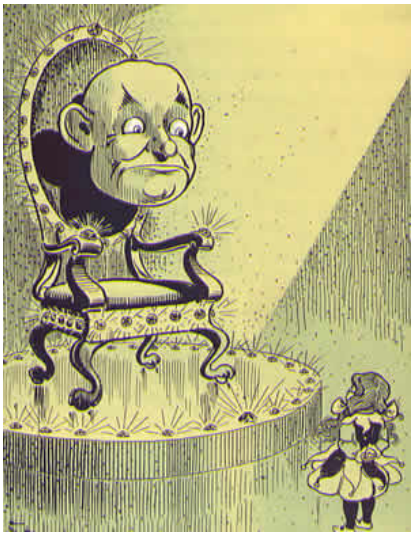
The algorithm used for mining needs to be through while

The algorithm used for prospecting must be efficient and capture episode parameters.

# An Ancient Problem

- To see reality when all we have is perception
- “Perception is reality”?





This is prospecting to identify the actor (and by extension the directory, script writer, etc.) when all we have is the play.

A miner is interested in extracting a known resource from a known location is large quantities, cheaply.

A prospector is interested in surveying large tracts of land to identify the locations of known resources, cheaply.

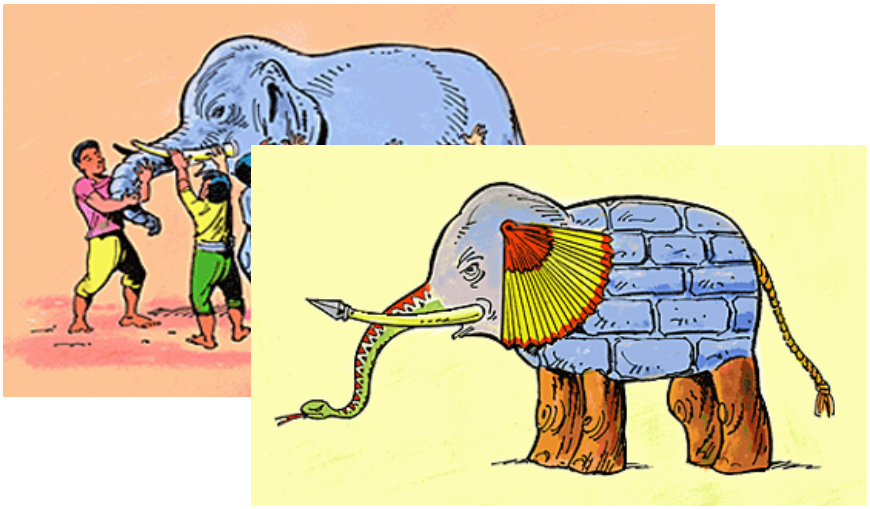
Clearly prospecting will extract some quantity of a resource but that is not its primary goal

Whereas for Mining resource, episode instance, extraction is exactly the goal.

The algorithm used for mining needs to be through while

The algorithm used for prospecting must be efficient and capture episode parameters.

# Perception v. Reality



**Blind Men and the Elephant (a.k.a., "Blindmen")**
**(by John Godfrey Saxe)**

American poet John Godfrey Saxe (1816-1887) based this poem,
"The Blind Men and the Elephant", on a fable that was told in India many years ago.
It is a good warning about how our sensory perceptions can lead to misinterpretations.

It was six men of Indostan
To learning much inclined,

Who went to see the Elephant
(Though all of them were blind),

That each by observation
Might satisfy his mind

The First approached the Elephant,
And happening to fall

Against his broad and sturdy side,
At once began to bawl:

“God bless me! but the Elephant
Is very like a wall!”

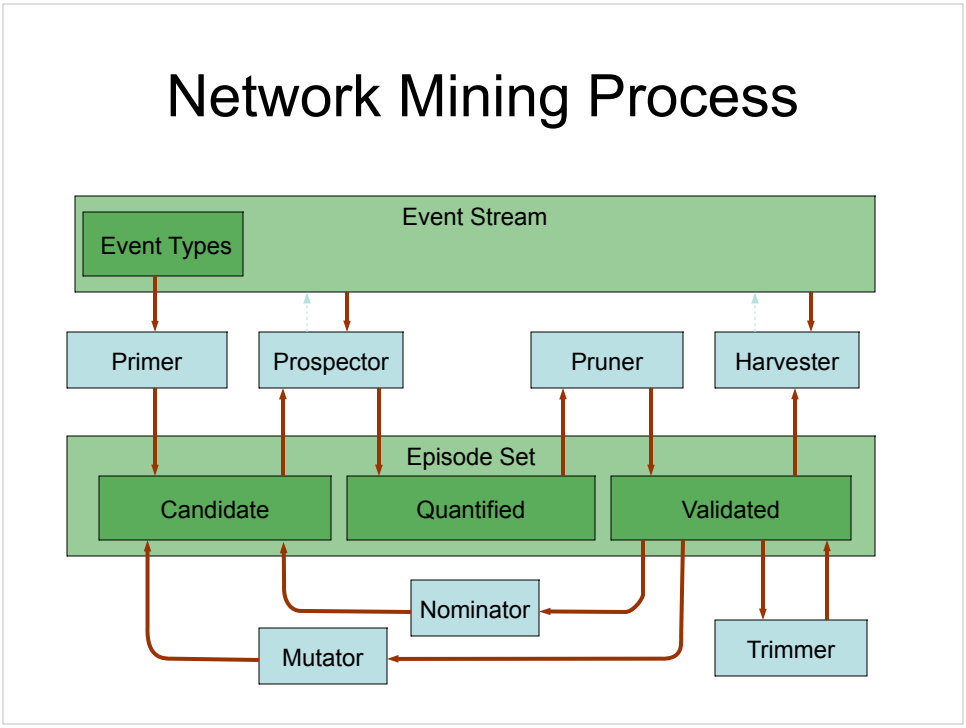
The Second, feeling of the tusk,
Cried, “Ho! what have we here

So very round and smooth and sharp?
To me ’tis mighty clear

This wonder of an Elephant
Is very like a spear!”



The Third approached the animal




(Green represents stateful objects)
(Blue represents functional objects)

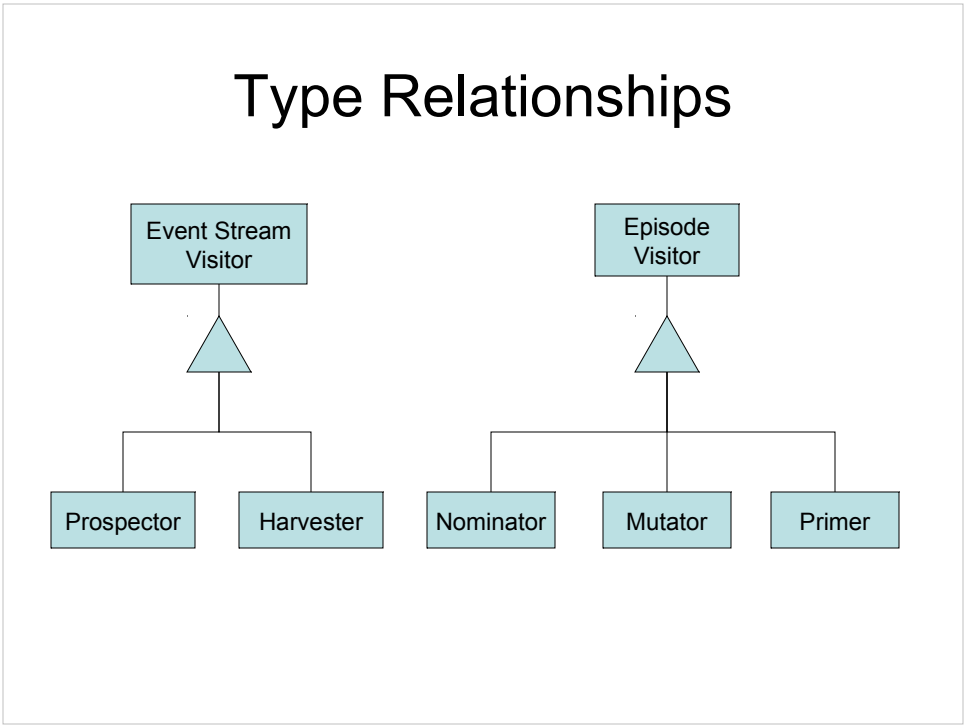
Episode/Pattern/Circuit Set/Rule/Miner:
The information that describes the things being mined.

Prospector/Counter/Miner/Search:
Transverses an Event Stream, in order, looking for a set (possibly of size 1) of episode instances
It has a visitor object that identifies each object as they are found.
Typically, the Prospector will traverse the Event Stream one time. It need not start at the beginning and stop at the end, but this is typical.

Nominator/Candidate Generator/Miner:
Creates new episode candidates.
Typically from previously identified episodes.
A special type of nominator generates a starting set of candidates from the event types identified in the event stream.
It is probably a good idea for the episode to retain an association with those objects used in its nomination in particular its ancestor episodes.

In general the arrows/edges represent a producer/consumer relationship.
This is presently enforced via a single threaded loop starting with the X-Nominator and concluding when the Puner produces no additional validated episodes.

The Prospector may also be halted due to feedback from the counter indicating that it has enough information to permit the Pruner (or whatever) to perform its job.



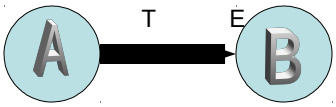

The Episode Visitor can be further sub-typed into

Nominator which builds higher order (i.e. longer) candidates from episodes.

Mutator converts from one type of episode to another.  Serial to branch or join.

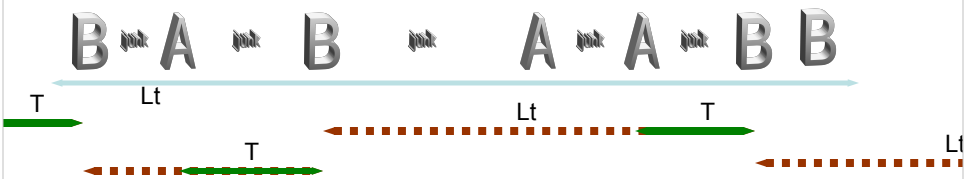
Primer creates episodes from event types.

## The Key to Graphs




The non-overlapping episode

- Signature: (A [T:E] B)
- Total Length: Lt


So, given that prospecting is about identifying reality from perceptions.

The key to effective prospecting is the key to reality.

What is a non-overlapping episode?

A graph is identified by its episodes.

Just as the Declaration of Independence may be recognized by a single signature

So to a graph may be characterized by its signatures.

A non-overlapping episode is defined by its signature

Event type names separated by intervals

# Total Non-overlapped Length

1. Develop a no connection formulation
    1. Expected value (first moment)
    2. Variation (second moment)
2. Develop a connected formulation
    1. Expected value (first moment)
    2. Variation (second moment)
    •

The development of a episode motif.

First determine the formulation for the frequency of fake episodes.

This is necessary as it forms the basis for the formulation for true episodes.

Once the expected values are formulated the variation follows from the derivation of the second moment (as expected value is the first moment).

# Chain Motif

If there is no connection between A and B

$$\left\langle \tilde{L}_{AB} \right\rangle = \frac{1}{\Pr(A)} + \frac{1}{\Pr(B)};$$

If there is a connection between A and B, expressed as a conditional probability. The probability that A causes B.

$$\gamma = \Pr(A \Rightarrow B) = \Pr(B \mid A) - \Pr(B);$$

$$\left\langle \overline{L}_{AB} \right\rangle = (1-\gamma)\left[\left\langle \tilde{L}_{AB} \right\rangle\right] + \gamma\left[\frac{1}{\Pr(A)} + \frac{1}{\gamma}\right];$$

$$\left\langle \overline{L}_{AB} \right\rangle = 2 + \frac{1}{\Pr(A)} + \frac{1}{\Pr(B)} - \frac{\Pr(A \cap B)}{\Pr(A)\Pr(B)};$$

The Apriori episode is of this type.

The amount of time between A and B is not constrained.

The chain episode is a special case of the chain interval episode where beta is 1 and T = 1/Pr(B).

# Chained Interval Motif

The chained interval is an episode where there is an occurrence of A followed by an occurrence of B after an interval of time T.

If there is no connection between A and B.
How many times will A be repeated before a B appears in time T?

$$\Pr\left(A \xrightarrow{T} B\right) = 1 - e^{-\frac{1}{\Pr(B)}\tau}\Big|_{T0}^{T1} = e^{-\frac{T0}{\Pr(B)}} - e^{-\frac{T1}{\Pr(B)}};$$

$$\left\langle \bar{L}_{AB} \right\rangle = \frac{1}{\Pr(A)\Pr(A \xrightarrow{T} B)} + T;$$

If there is a connection between A and B with a delay T.

$$\gamma = \Pr(A \Rightarrow B) = \frac{1}{\lambda_{AB}};$$

$$\left\langle \bar{L}_{AB} \right\rangle = (1-\gamma)\left[\left\langle \bar{L}_{AB} \right\rangle\right] + \gamma\left[\frac{1}{\Pr(A)} + T\right];$$

The chained interval motif is the general case of the chain motif.

In the independent case, the question is how many times must A fire before B fires at time Tab after A?

Of course, this presumes a certain range around Tab where an occurrence of B would be taken as occurring then.

Then lets take T0 as the start of Tab and T1 as the end.

Given that B is an independent process the Poisson assumption holds and the probability is the area under the exponential.

There are different definitions of an episode.

One definition allows for an episode A->B where only A fires. This formulation is not consistent with this definition.

Another definition asserts that in order for an A event to be part of an A->B episode instance B must fire.

I this formulation gamma is the probability that A causes B.

# Branching Interval Motif

The branch interval is an episode where there is an occurrence of A followed by an occurrence of B and an occurrence of C after intervals of time Tb and Tc, respectively.

There is no connection between A and B.

How many times will A be repeated before a B appears in time T?

$$\beta = \Pr(\frac{1}{\Pr(B)} = T)?;$$

$$\tilde{L}_{AB} = \frac{\beta}{\Pr(A)} + T;$$

There is a connection between A and B, expressed as a conditional probability.

$$\gamma = \Pr(B \mid A) - P(B);$$

$$\overline{L}_{AB} = (1-\gamma)\left[\tilde{L}_{AB}\right] + \gamma\left[\frac{1}{\Pr(A)} + \frac{1}{\gamma}\right];$$

The branching interval has a special type where the common node A is hidden.

This is not generally a problem as A is typically preceded by a node Z which can serve as A's proxy.

# Joining Interval Motif

The join interval is an episode where there is an occurrence of A and an occurrence of B both followed by the same occurrence of C after intervals of time Tac and Tbc, respectively.

There is no connection between A and B,
How many times will A be repeated before a B appears in time T?

$$\beta = \Pr(\overline{A \cap B} = T) = \frac{1}{\Pr(B)};$$

$$\tilde{L}_{AB} = \frac{\beta}{\Pr(A)} + T;$$

There is a connection between A and B, expressed as a conditional probability.

$$\gamma = \Pr(B \mid A) - P(B);$$

$$\overline{L}_{AB} = (1-\gamma)\left[\tilde{L}_{AB}\right] + \gamma\left[\frac{1}{\Pr(A)} + \frac{1}{\gamma}\right];$$

# Porting Components

- Define episode set structure in candidate environment
- Provide a typical episode prospector as prototypical C function
- Provide a corresponding nominator as an environment native function
- Provide a simple threshold pruner as an environment native function

Essentially, three R functions, one being implemented as a wrapped C/C++ function.

The other two be R native functions.

One of the key deliverables would be a description of the episode qualifiers and an identification of the other related components.

This activity should take about two months.

# Demonstration Problem

- Specify the demonstration problem
- Demonstrate the use of the components against the problem

Validation criteria.

This should be worked out BEFORE any coding work is done.

This demonstration problem should not be something where the answer is unknown but rather a well studied problem that may benefit from even a bit more understanding.

# Rationalization

- Examine/evaluate current (sub) frameworks in the target environment as possible integration points.

In particular arules and arulesSequence.