

## Ceaseless, Sequential-Case Based CBR

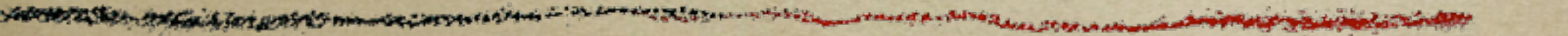
October 2003

```
No exact OS matches for host  
Nmap run completed -- 1 IP address (1 host up) scanned  
# sshnuke 10.2.2.2 -rootpw="210H0101"  
Connecting to 10.2.2.2:ssh ... successful  
Attempting to exploit SSHv1 CRC32 ... successful.  
Resetting root password to "210H0101".  
System open: Access Level (9)  
# ssh 10.2.2.2 -l root  
root@10.2.2.2's password: |
```



**CCIA2003**  
**Sisè Congrés Català d'Intel·ligència Artificial**  
Divendres, 23 d'octubre de 2003

# ***Alba - A Cognitive Assistant for Network Administration***

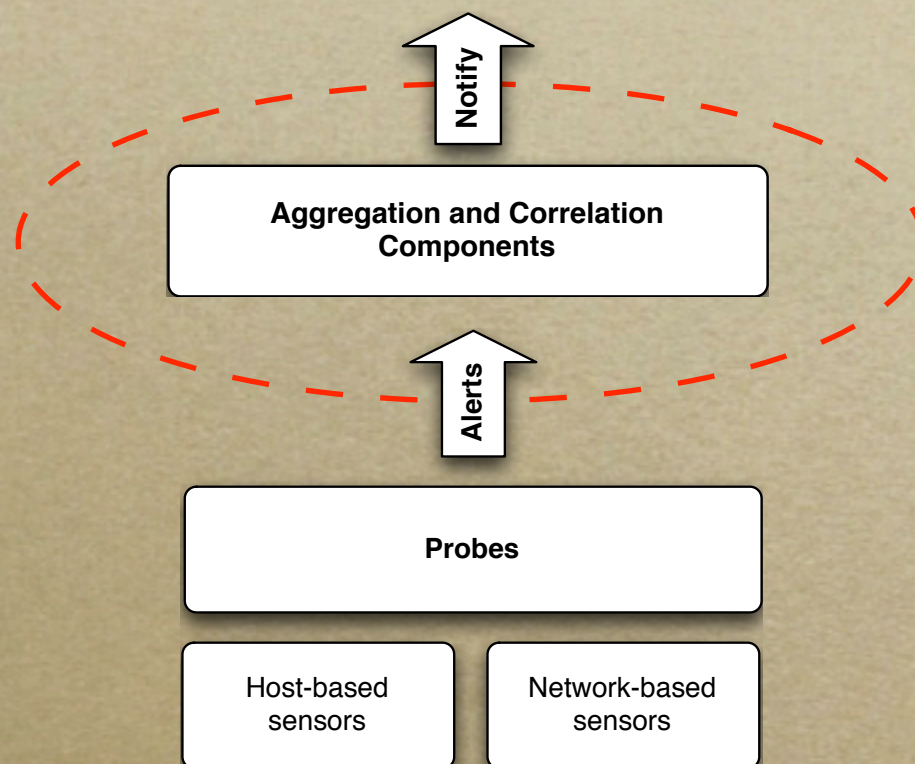


Francisco J Martin  
**EECS-OSU**  
**Corvallis, Oregon (USA)**

Enric Plaza  
**IIIA-CSIC**  
**Bellaterra, Catalonia (Spain)**

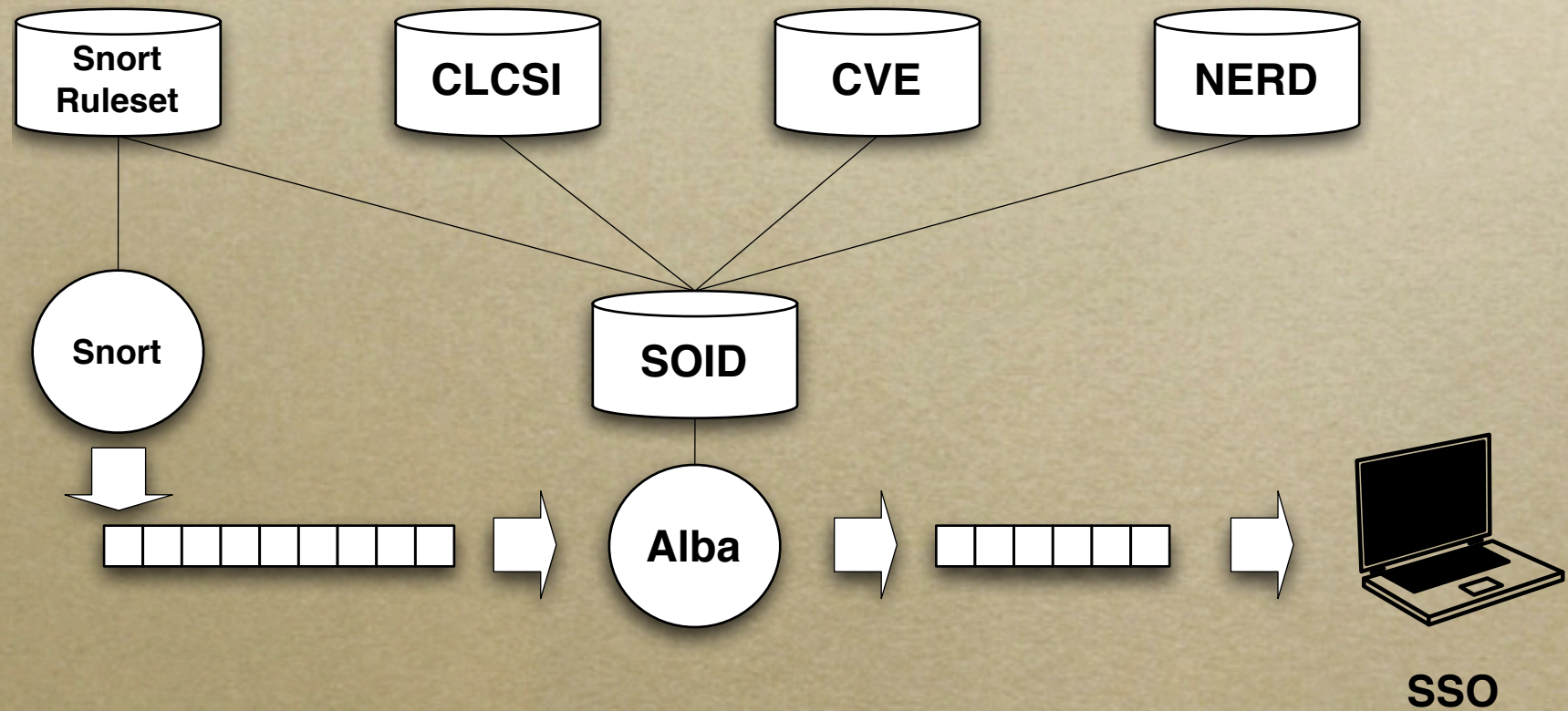
## *Our approach overview (I)*

- **Alert Triage (AT)** is the process of rapid and approximate prioritization for subsequent action of an IDS alert stream.



Our goal is to **increase the efficiency** of **current IDSes**.

## *Our approach overview (II)*

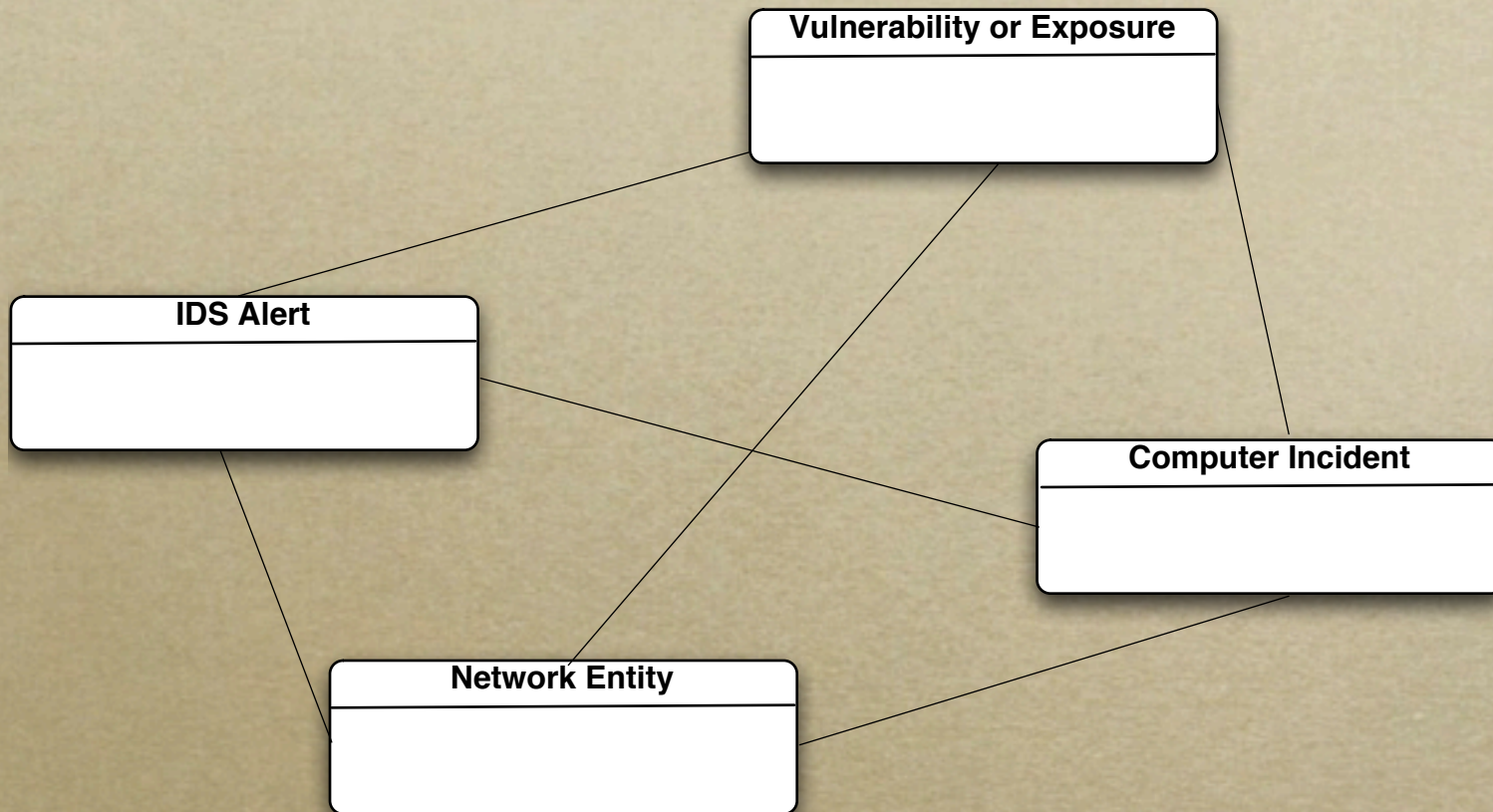


# Three-layered approach

---

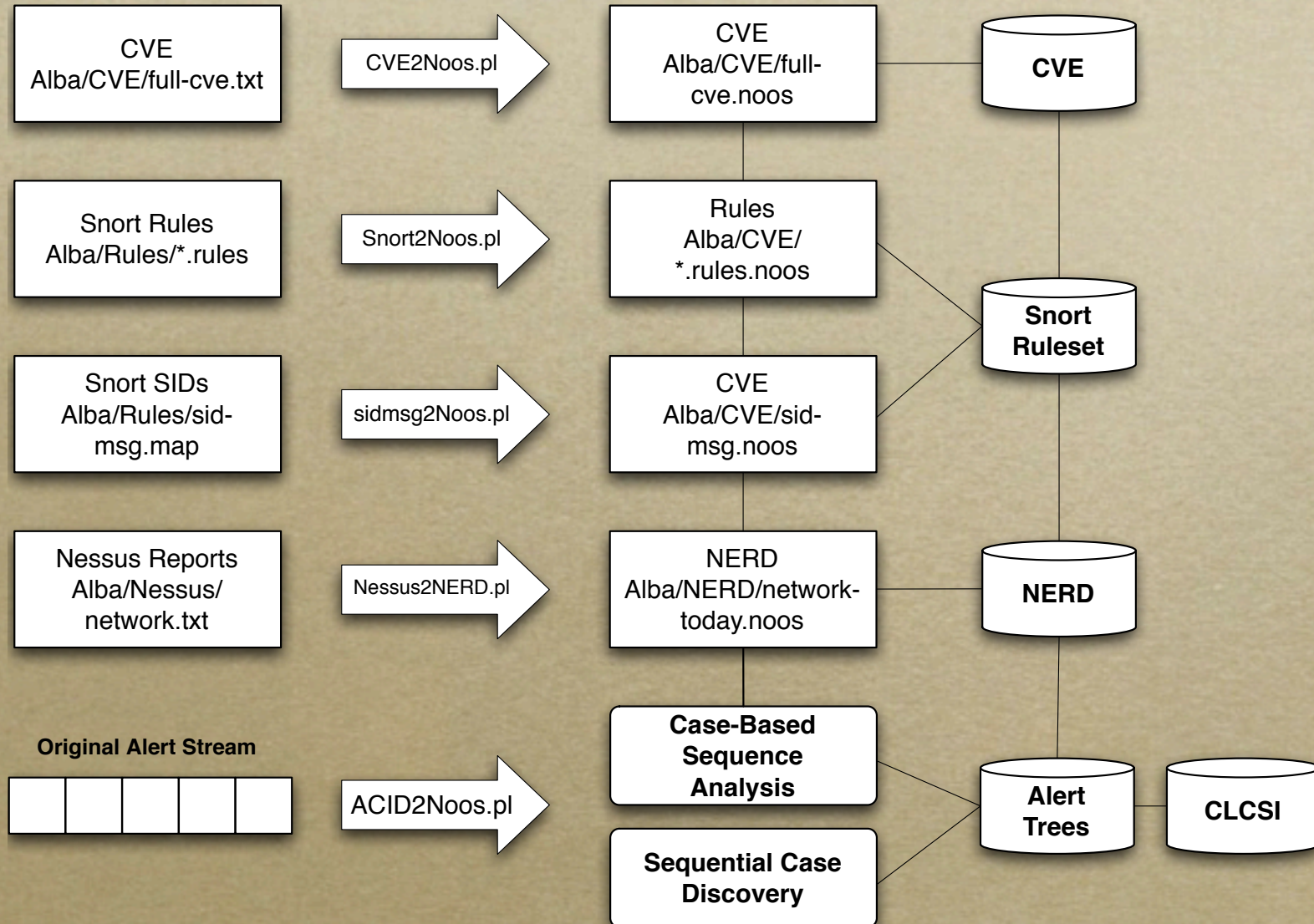
- **Perception layer**
  - *sensors emit alerts on suspicious actions in the network*
- **Recognition layer**
  - *SOID ontology models monitored actions*
  - *sequential cases (actionable trees)*
- **Planning layer**
  - *plan recognition to prioritise alerts and use them to anticipate final goals*

# ***SOID overview***



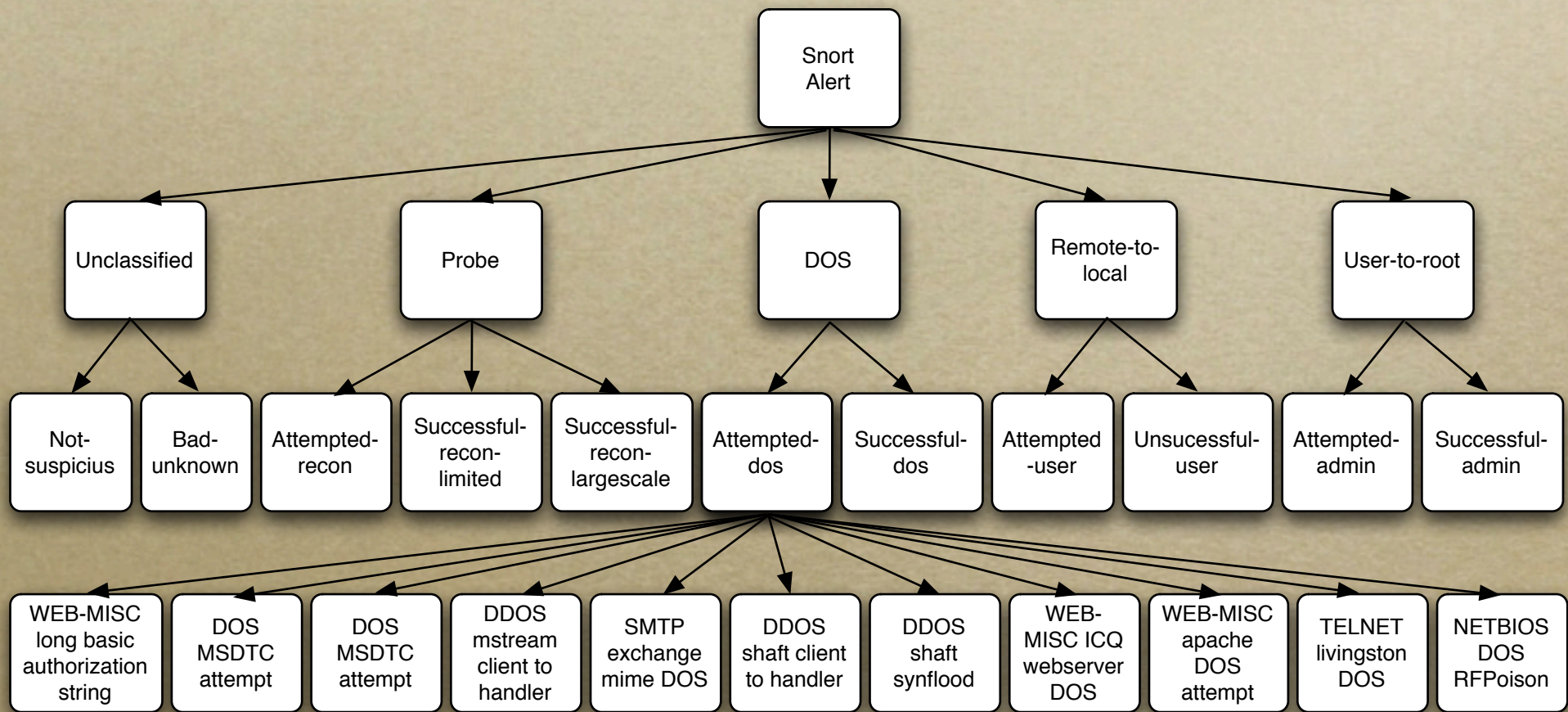
- For each knowledge source a **separate ontology** has been built.
- **SOID** merges those ontologies on top of the **Noos** knowledge representation language.

# ***SOID details***



# *Example of a taxonomy of Snort alerts*

- A simple taxonomy of Snort alerts.



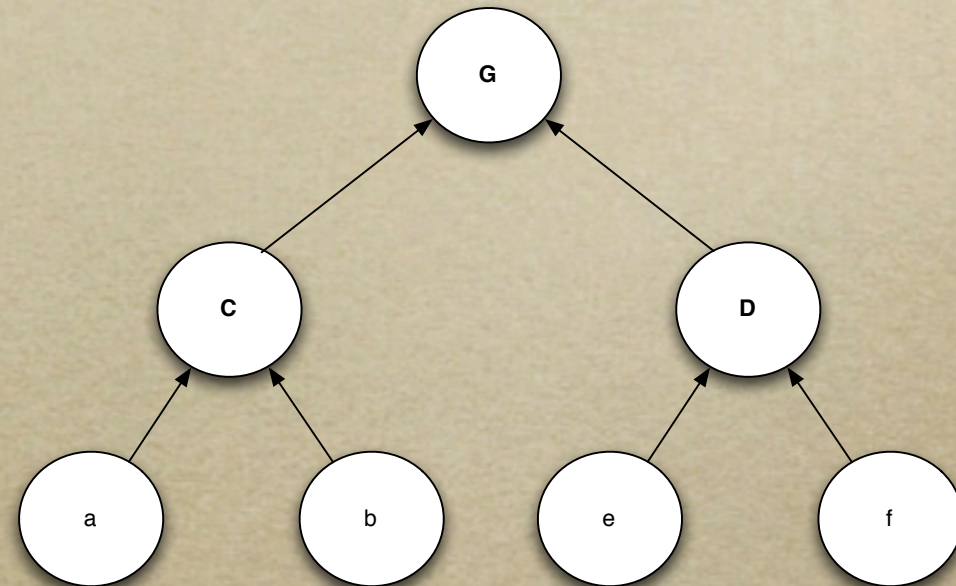
# *Outline*

---

- **Actionable Trees**
- **Case Activations**
- **Ceaseless Retrieve**
- **Ceaseless Reuse**
- **Ceaseless Revise**
- **Ceaseless Retain**

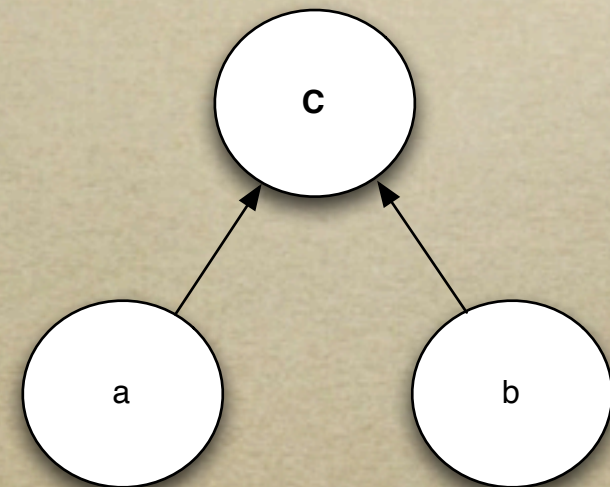
# Actionable Trees: Definition

- A highly intuitive and machine learnable knowledge structure that enables the representation of **sequential cases**.
- An **actionable tree** (AT) is a **Multi-Rooted Directed Acyclic Graph** (MDAG) with the semantics that **roots** represent **observable symptom events, intermediate nodes** (in the trunk and crown) represent **composite (serial or parallel) cases** and the **arcs** represent **part-whole relationships**.
- The **crown** made up of only one node represents the overall **case**. There is one and only one path from each root node to the crown.



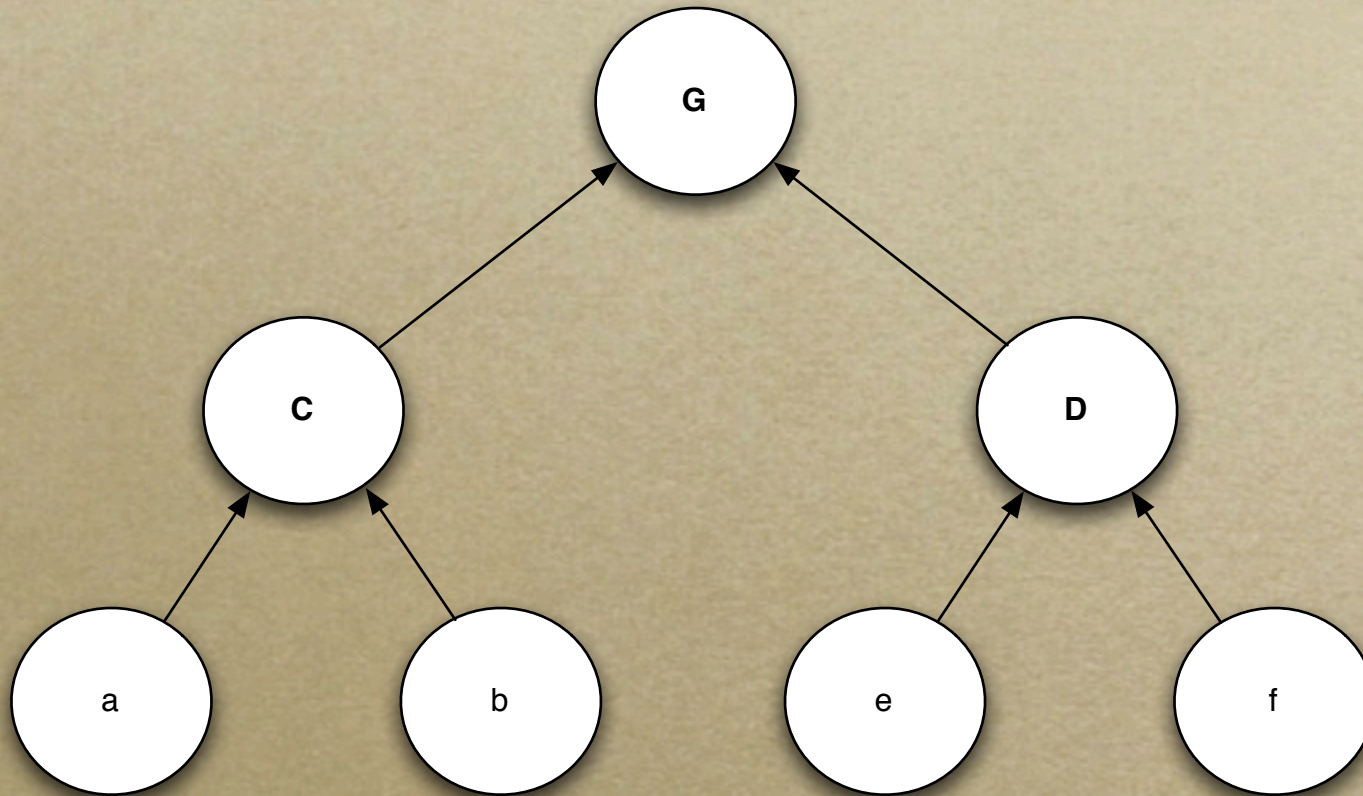
# Actionable Trees: Definition (II)

- 
- Roots represent observable symptom events (i.e. alerts)
  - nodes **a** and **b** are **complex objects** represented by means of **feature terms**.
- The crown represents a case:
  - node **c** stores information about:
    - the **risk** that supposes the occurrence of **a** and **b** together. **Risk** is a combination of **threat**, **exposure**, and **cost**.
    - **constraints** that limit the correlation of **a** and **b** (using a set of common features of **a** and **b** for which **path equality** must be hold)
    - the prioritization that received **a** and **b** (i.e. the case solution)



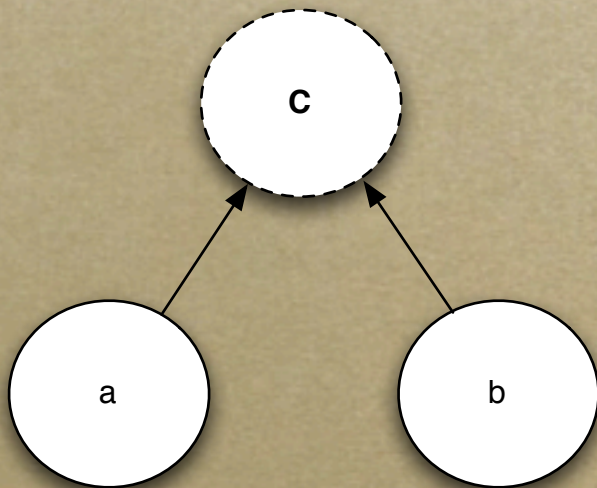
# *Actionable Trees: Compoundability*

- Actionable Trees are compoundable

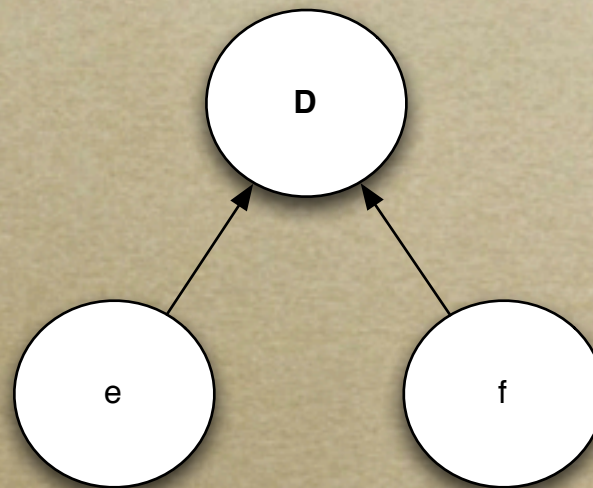


# Actionable Trees: types of intermediate nodes

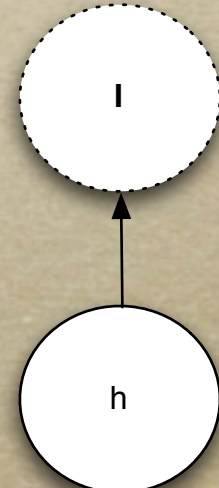
- There are three types of intermediate nodes:
  - a-nodes** (dashed nodes) represent **parallel cases**
  - s-nodes** represent **serial cases**
  - b-nodes** (doted nodes) represent **burst cases** (i.e. flood situations)



yields: {[a b] [b a]}



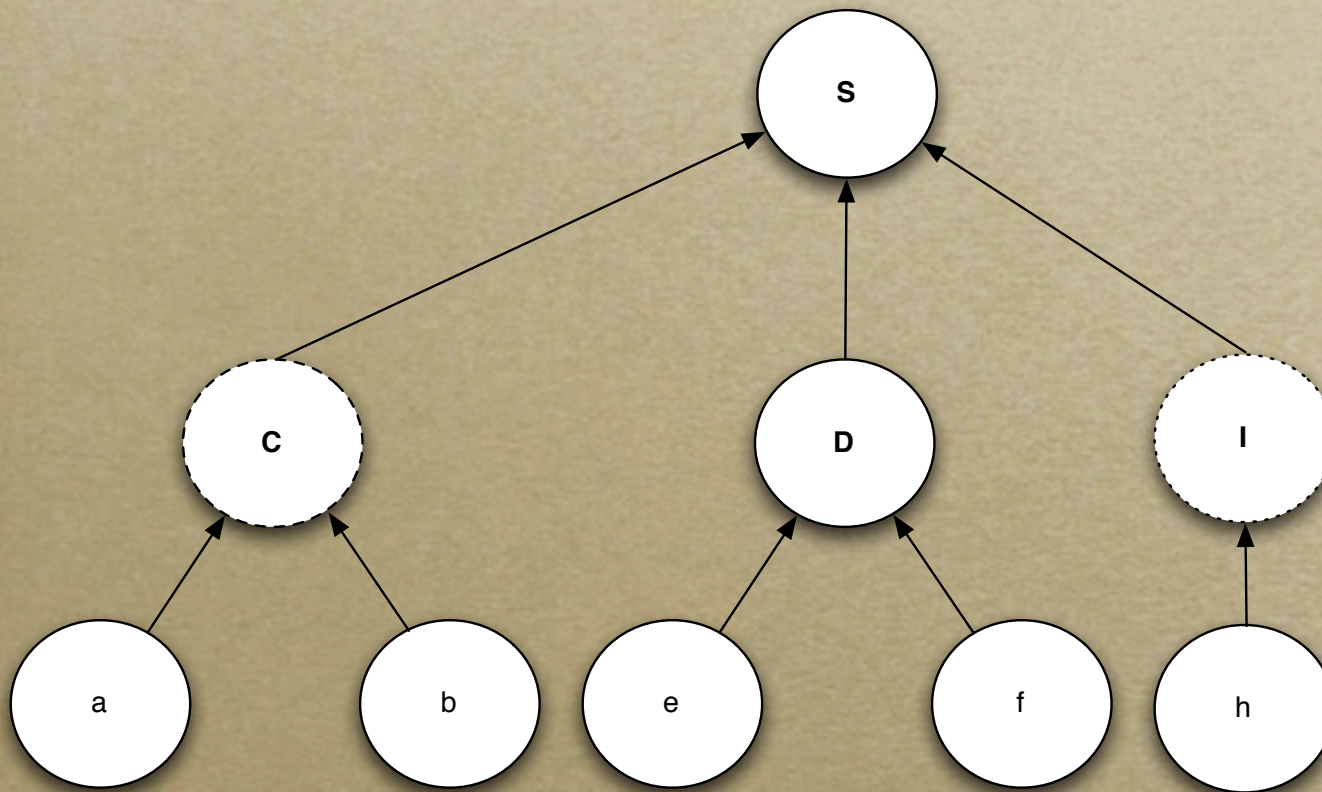
yields: {[e f]}



yields: {[h<sup>n</sup>] n > X/t}

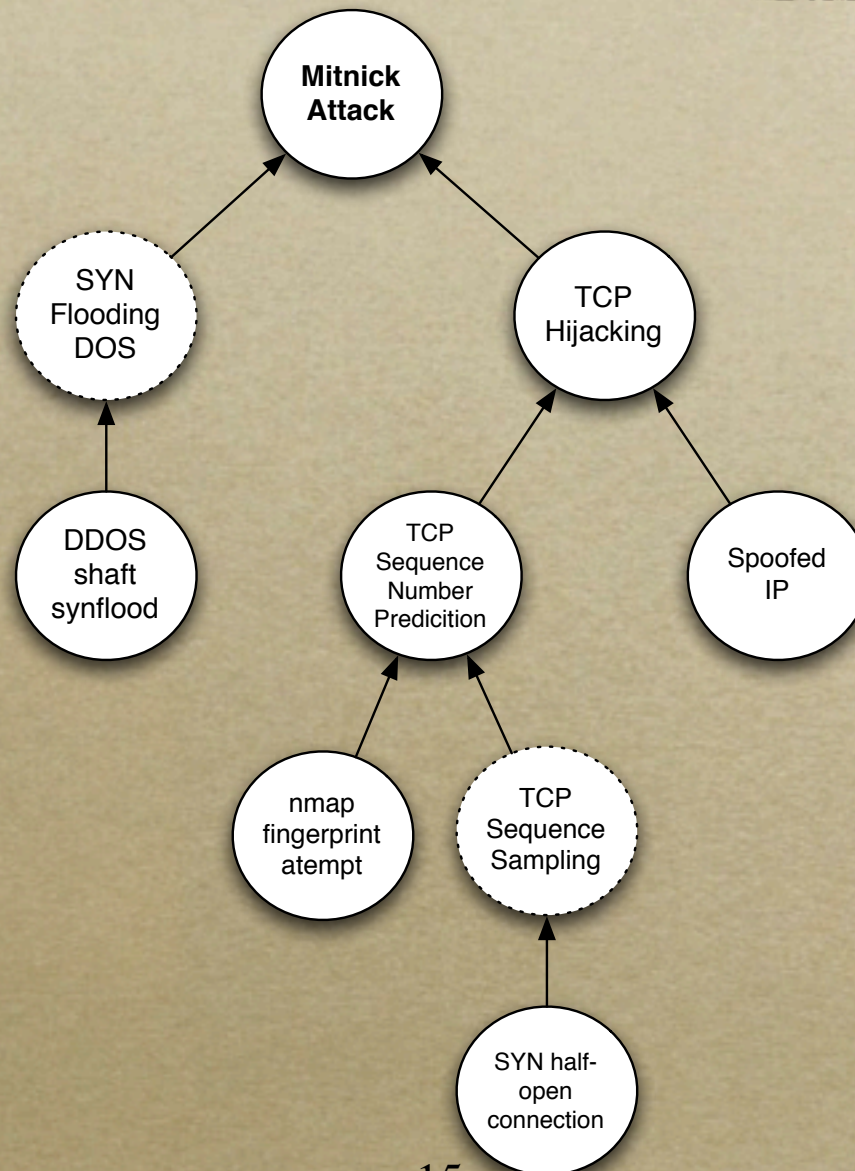
# Actionable Trees and Context-Free Grammar correspondence

- A direct mapping can be established between an **Actionable Tree** and a **Context-Free Grammar** that yields all the sequences represented by the Actionable Tree.



**G:**  $S \rightarrow C D I$   
 $C \rightarrow ab \mid ba$   
 $D \rightarrow ef$   
 $I \rightarrow h^{X/t_I}$

# Actionable Trees Example: Mitnick Attack



# *Dynamic Sequence Similarity*

- We have defined a **dynamic similarity** between two sequences of complex objects based on the following components:
  1. A **dynamic subsumption scoring scheme** that:
    - establishes the similarity between two individual alerts according to its probability of occurrence and its position in the hierarchy of sorts.
      - Rare alerts receive a high score and frequent alerts receive a low score.
    - is continuously updated upon arrival of new alerts.
  2. A **semi-global alignment** obtained by insertion of a number of dummy feature terms such that both sequences have the same length and in the individual alignment of the elements at least one of the two element isn't a dummy feature term.
  3. Two **operations** that allow a sequence to be altered so that corresponding elements in both sequences to be comparable.
    - **Abduction**: injecting an alert of sort *a* in the alert stream at a given position.
    - **Neglection**: ignoring an alert in the alert stream.
  4. A **dynamic programming** formulation that computes the score of the optimal alignment.

# Dynamic Sequence Similarity

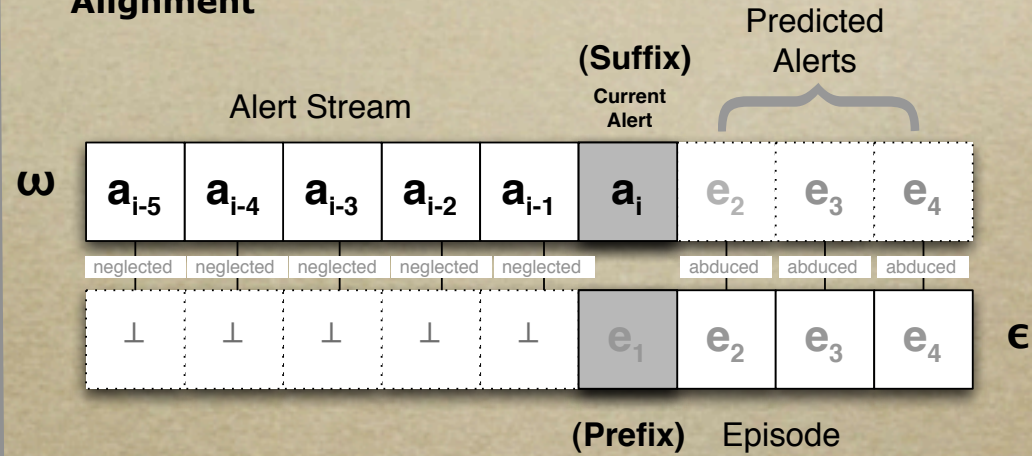
$$S_{I \sim S} S_2 = \max_{1 \leq j \leq |S_2|} S(|S_1|, j).$$

## 1. Dynamic Scoring Scheme

$$\mathcal{M}_{i,j} = \begin{cases} \frac{1-q_i}{q_j} & \text{if } \psi^{a_i} \sqsubseteq \psi^{a_j} \\ -1 & \text{otherwise} \end{cases}$$

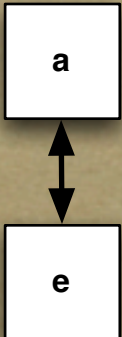
M	A	B	C	D	E	Z	Y	X	⊥
A	2.167	-1	-1	-1	-1	-1	-1	-1	-1
B	-1	2.8	-1	-1	-1	-1	-1	-1	-1
C	-1	-1	2.8	-1	-1	-1	-1	-1	-1
D	-1	-1	-1	18	-1	-1	-1	-1	-1
E	-1	-1	-1	-1	8.5	-1	-1	-1	-1
Z	2.167	-1	-1	-1	-1	2.167	-1	-1	-1
Y	-1	1.8	1.8	-1	-1	-1	0.9	-1	-1
X	-1	-1	-1	16	8	-1	-1	5.333	-1
⊥	0	0	0	0	0	0	0	0	0

## 2. Semi-Global Alignment

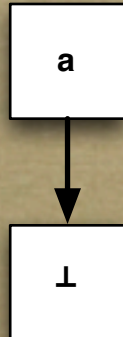


## 3. Operations

Subsumption



Neglection



Abduction



## 4. Dynamic Programming Formulation

$$S(0, 0) = 0$$

$$S(i, 0) = S(i - 1, 0)$$

$$S(0, j) = S(0, j - 1) + C^a(\vec{S}_2[j])$$

$$S(i, j) = \max \begin{cases} S(i - 1, j) & + C^n(\vec{S}_1[i]) \\ S(i, j - 1) & + C^a(\vec{S}_2[j]) \\ S(i - 1, j - 1) & + \mathcal{M}(\text{root}(i), \text{root}(j)) \end{cases}$$

# *Case Activations*

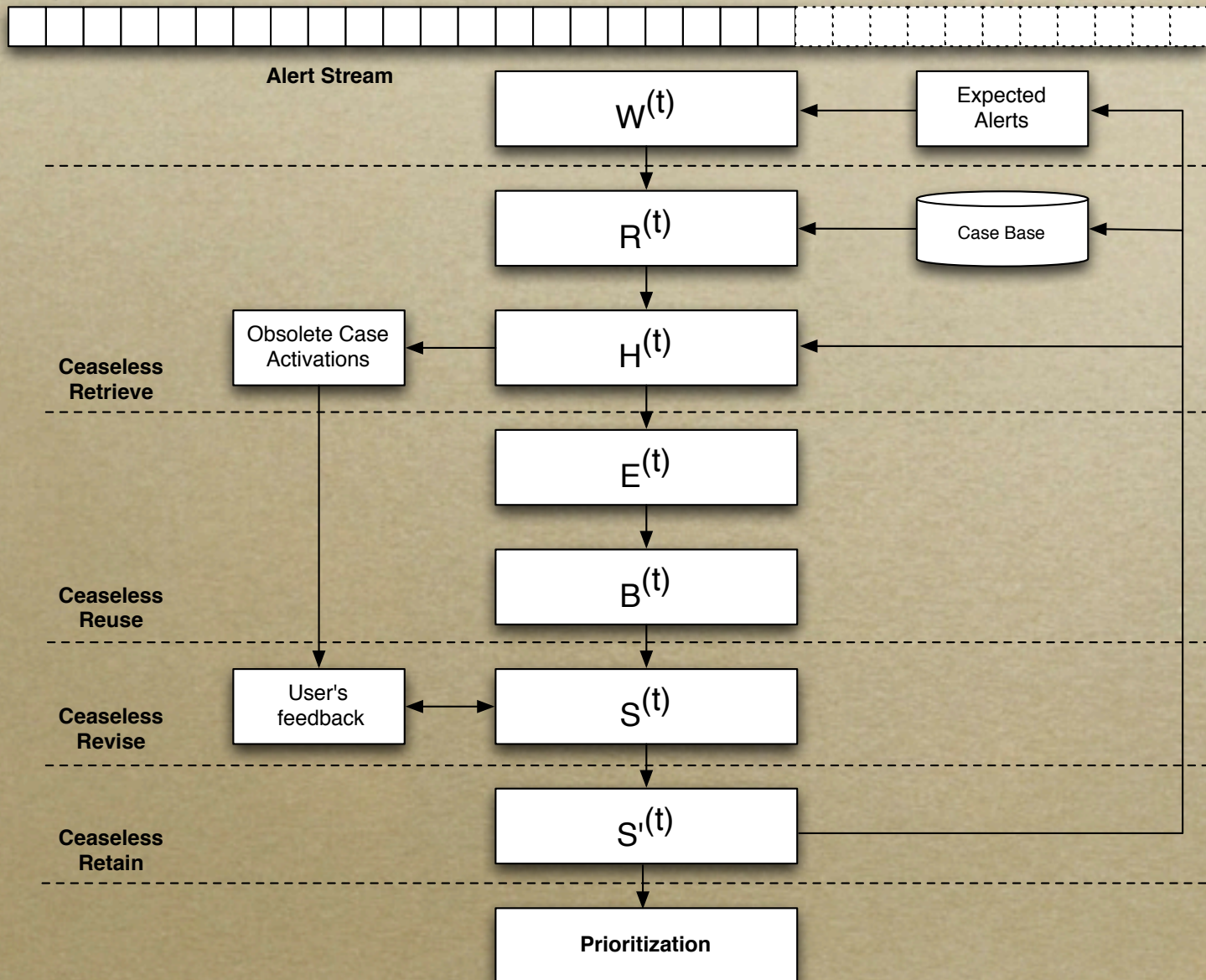
- A **case activation** represents a **hypothesis** on a case explaining the current **situation**.
- A case activation is composed of:
  - **Case** (risk(threat,exposure,cost), yields of observable symptom events, and prioritization).
  - **Observed** (O) symptoms that represent **observed alerts**.
  - **Abducted** (A) symptoms that represent **lost symptoms**.
  - **Neglected** (N) symptoms that represent **spurious symptoms**.
  - **Evidence** computed in terms of O, A, and N symptoms.
- Cases are activated using a **similarity-based likelihood judgment** (i.e. similarity between the current window of alerts and a case's yield of observable events).

# *Ceaseless CBR*

---

- A push-pull constructive situation awareness process governed ceaselessly by:
  1. Observational data. The sequence of events received pushes towards a situation
  2. The sequential case base pulls towards the best explanation of the current situation interpreted in terms of past cases.
- **Ceaseless CBR** is decomposed in four parallel processes:
  - **Ceaseless Retrieve**
  - **Ceaseless Reuse**
  - **Ceaseless Revise**
  - **Ceaseless Retain**

# Ceaseless CBR



# *Ceaseless Retrieve (I)*

- **Ceaseless Retrieve:**

- $\mathbf{R}^{(t)}$ : Using the sequence of alerts ( $\mathbf{O}$ ) returned by the corresponding window model ( $\mathbf{w}^{wm(t)}$ ) and a dynamic similarity measure, **retrieve** those cases that are similar to such sequence above a user-defined threshold ( $\theta$ ).
- $\mathbf{H}_i$ : A case activation ( $H_i$ ) is created for each retrieved case containing **observed, abducted and neglected** alerts as well as an estimation of its **evidence** and the **risk** that supposes.
- $\mathbf{H}^{(t)}$ : New case activations (**hypotheses**) are merged with previous case activations considering the **constraints** imposed by each case (**path equality checking**). For example, the same source and address in all the sequence of alerts.

## *Ceaseless Retrieve (II)*

- Initially:
  - $\mathbf{C}^{(0)} = \{C_1 \dots C_n\}$  (i.e. case-base)
  - $\mathbf{R}^{(0)} = \emptyset, \mathbf{H}^{(0)} = \emptyset,$
  - $\mathbf{E}^{(0)} = \emptyset, \max \mathbf{B}^{(0)} = 1$
  - $\mathbf{S}^{(0)} = \emptyset, \mathbf{S}'^{(0)} = \emptyset$
- $\mathbf{W}^{wm(t)}$  extracts the next sequence of alerts from the alert stream according to a given window model  $wm$  (**landmark**, **sliding**, **damped** or **alert-driven**).
- $\mathbf{R}^{(t)}(\mathbf{W}^{wm(t)}) = \{\text{Case Activations} : C_i \in C^{(t)} \text{ and } \text{sim}(\mathbf{W}^{wm(t)}, C_i) > \square\}$
- $\mathbf{H}^{(t)} = \mathbf{H}^{(t-1)} \cup \mathbf{R}^{(t)} \quad \Leftarrow \text{Current Situation}$
- $\mathbf{H}^{(t)}$  keeps a number of case activations for each pending alert (i.e. alerts that have not received an explanation/prioritization yet).

## *Ceaseless Reuse (I)*

- **$E^{(t)}$** : Computes a set of **explanations** (combinations of case activations that explain completely the current sequence of observed alerts (**O**)). This set is computed following a **minimum description length (MDL)** principle. Those explanations:
  - that contain case activations that appear in other explanations that are already in the set.
  - whose size is greater than the minimum size of the combinations above are not contemplated.
- **$B^{(t)}$** : An estimation of the goodness of each explanation in  **$E^{(t)}$**  is computed. This estimation considers the probability of occurrence and can also consider the risk and cost of each explanation.
- Explanations are **ranked** according to  **$B^{(t)}$**  and then proposed to the user for their revision.

## Ceaseless Reuse (II)

- $\mathbf{E}^{(t)}(S^{(t)}, O) = \{H' \subseteq H^{(t)} : H' \text{ explains all events in } O \text{ and } \nexists H'' : |H''| < |H'| \text{ and } H' \cap H'' \neq \emptyset\}$ 
  - $\mathbf{E}^{(t)}$  is computed following a **minimum description length (MDL)** principle.
  - The following observation: “**the probability of multiple coincidental sources is low**” induces the following heuristic:
  - $H'_i$  is not included in  $E^{(t)}$  if it contains a case activation that is already contained by  $H'_j \in E^{(t)}$  such that its size is lower and its risk is greater.
- $\mathbf{B}^{(t)}(E_i)$  is a **belief function** that represents the likelihood that all cases in  $E_i$  have occurred and  $E_i$  explains all events in  $O$ :
- $$B^{(t)}(E_i^{(t)}, \mathcal{O}) = \prod_{H_i, c_i \in E_i} p(c_i) \prod_{a_i \in \mathcal{O}} \left( 1 - \prod_{H_i, c_i \in E_i} (1 - p(a_i | c_i)) \right)$$
- $\mathbf{B}^{(t)}$  is computed incrementally:  $\mathbf{B}^{(t)}(E^{(t)}) = B^{(t-1)} \cup E^{(t)}(H^{(t)}, O)$
- Explanations are **ranked** according to  $B^{(t)}$
- **Best Explanations**  $\Rightarrow \{E_i \in E^{(t)} : B^{(t)}(E_i) \text{ is maximal}\}$

# Ceaseless Revise

- This process **continuously** provides a human (expert) **operator** with the set of **most likely explanations** given the alerts received so far (instead of presenting a solution periodically).
- The operator can define a **threshold  $\theta'$**  such that individual explanations whose likelihood is above it produce an automatic triage of the corresponding alerts and initiates the same process that above.

$$S^{(t)} = \{E_i \in E^{(t)} : B^{(t)}(E_i) \text{ is maximal}\} \cup \{H_i : B^{(t)}(H_i) > \theta'\}$$

- The **operator's feedback** may create a completely new case or update a past case:
  - adding, deleting or altering observable events or constraints among them.
  - altering its risk(threat, exposure, or cost) or the corresponding prioritization.
- The **operator's feedback** produces a set of **revised solutions** that in turn produces the triage of the corresponding alerts and initiates a **back-propagation** process that automatically updates  $H^{(t)}$  and the set of **expected** alerts (i.e. alerts that are probably to occur and have already received an explanation).

$$S'(t) = \text{feedback}(S^{(t)})$$

# *Ceaseless Retain*

- Once a solution has been revised by the user:
  - The probability of occurrence of each case is updated as well as the probability of occurrence of each alert in the cases that have been used in the solution.
  - Those cases whose probability of occurring together is above the probability of occurring separately are merged together in a new case.
  - Other features of intervening cases such as risk or cost can also be updated in this process.
  - New cases are created using alerts that do not appear in any other case.
- In other words, this process **ceaselessly** stores the solutions revised by the former process

$$C^{(t)} = C^{(t-1)} \cup S'(t)$$

## Alert Stream

a b a a b c d e c c b a b a c c e a b

## Window

a a b c d

## Expected Alerts

a

ee, a, cc

O = Window - Expected Alerts

$C_i = \langle \text{yield, priority, risk} \rangle$

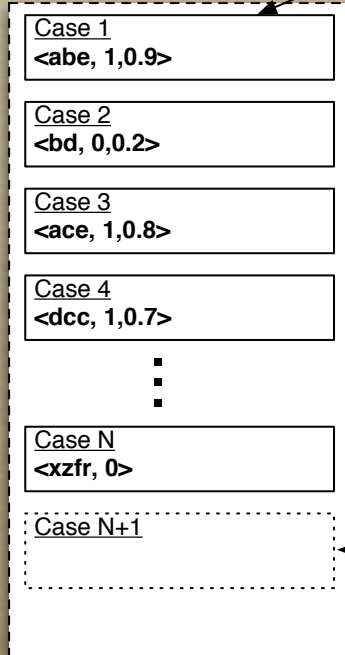
$H_i = \langle \text{case, observed, abducted, neglected} \rangle$

$E_i = \langle \{C_i\}, \{\text{abducted}\} \rangle$

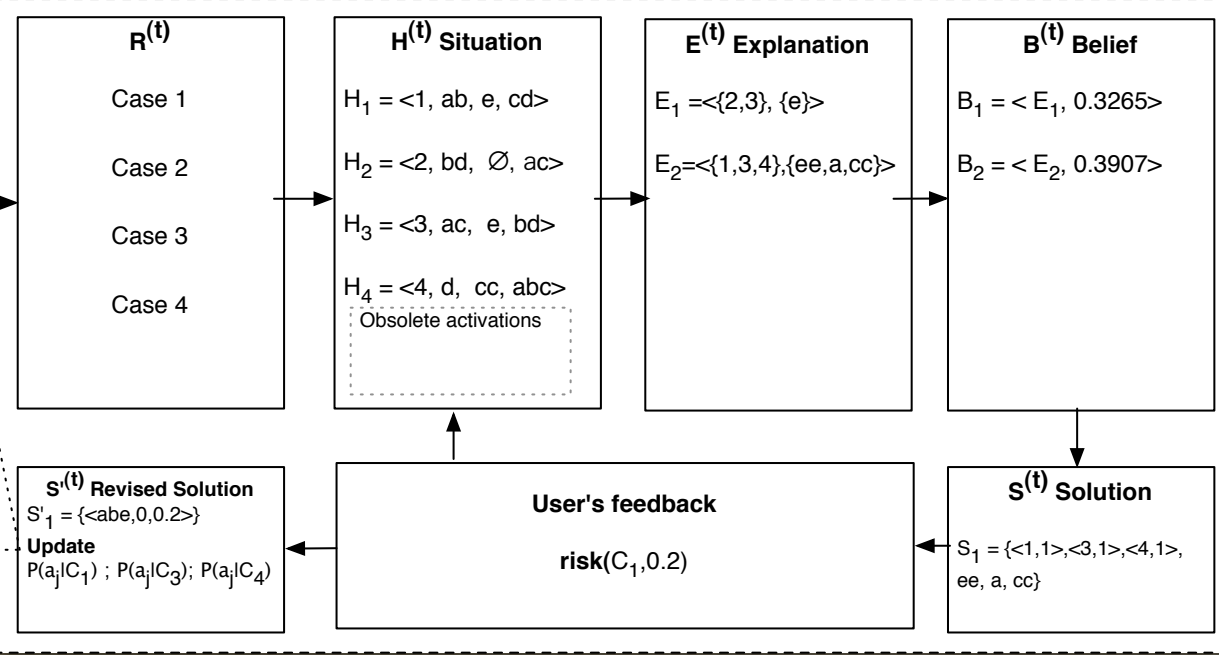
$B_i = \langle E_i, \text{belief} \rangle$

$S_i = \langle \{C_i, \text{priority}_i\}, \text{expected alerts} \rangle$

## Case Memory



## Working Memory



Case	Yield	O				Observed	Abducted	Neglected
		a	b	c	d			
1	abe	x	x	-	-	ab	e	cd
2	bd	-	x	-	x	bd	-	ac
3	ace	x	-	x	-	ac	e	bd
4	dcc	-	-	-	x	d	cc	abc
{1,2}	abe,bd	x	x	-	x	abd	e,b	c
{1,3}	abe,ace	x	x	x	-	abc	e,e,a	d
{1,4}	abe,dcc	x	x	-	x	abd	e,cc	c
{2,3}	bd,ace	x	x	x	x	abcd	e	-
{2,4}	bd,dce	-	x	x	x	bcd	cc,d	a
{3,4}	ace,dcc	x	-	x	x	acd	e,cc	b
{1,2,4}	abe,bd,dcc	x	x	-	x	abd	e,b,cc	c
{1,3,4}	abe,ace,dcc	x	x	x	x	abcd	e,e,a,cc	-

$C_i$	$P(C_i)$	alert	$P(a_j C_1)$	alert	$P(a_j C_3)$
1	0.91	a	0.89	a	0.67
2	0.62	b	0.65	c	0.73
3	0.53	e	0.65	e	0.53
4	0.81				
...	...				
N	0.42				

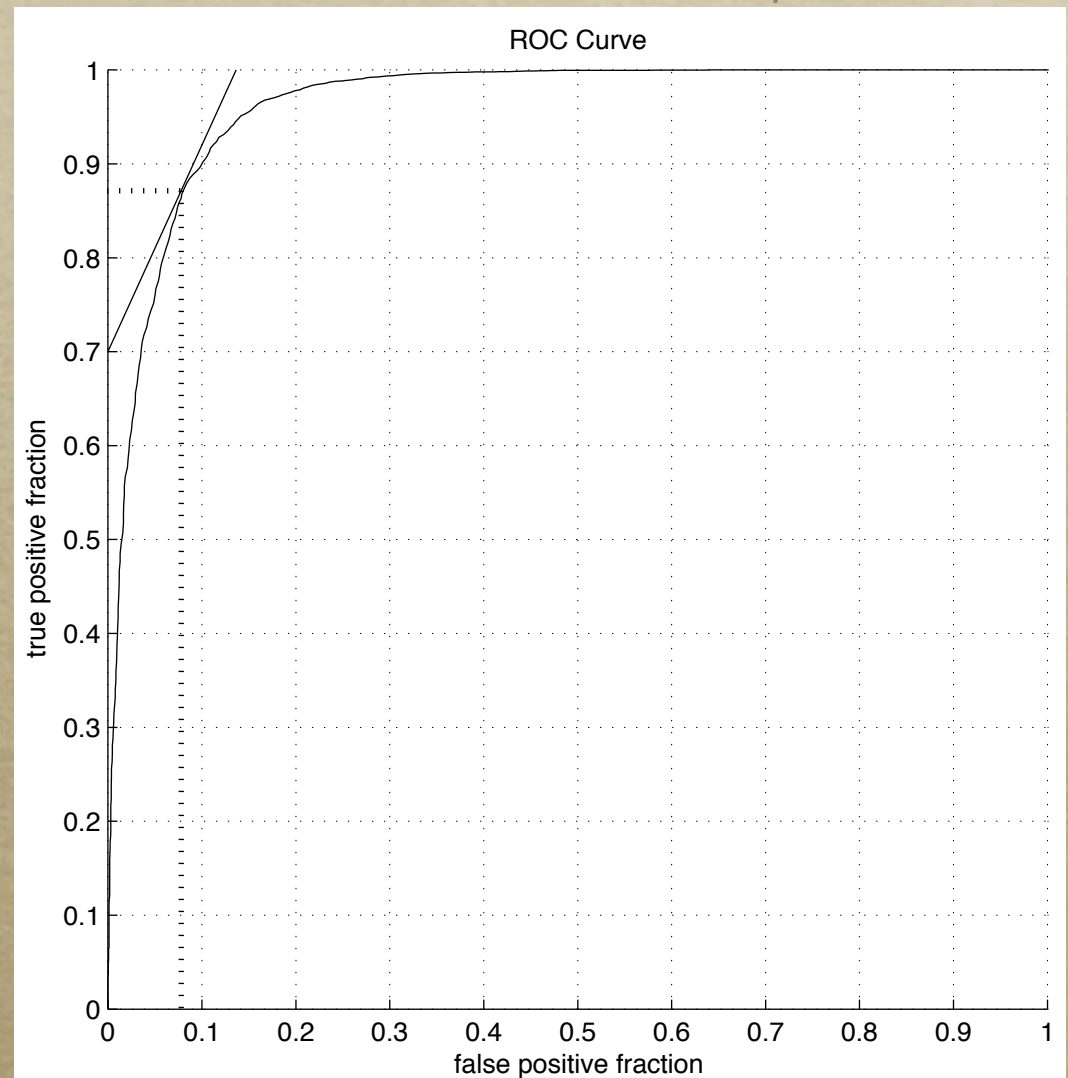
alert	$P(a_j C_2)$	alert	$P(a_j C_4)$
b	0.53	d	0.67
d	0.67	c	0.89
		c	0.89

$$B_1 = 0.62 * 0.53 * (1 - (1 - 0.53) * (1 - 0.67) * (1 - 0.67) * (1 - 0.73) * (1 - 0.54))$$

$$B_2 = 0.91 * 0.53 * 0.81 * (1 - (1 - 0.89) * (1 - 0.65) * (1 - 0.65) * (1 - 0.67) * (1 - 0.73) * (1 - 0.53) * (1 - 0.67) * (1 - 0.89) * (1 - 0.89))$$

# Preliminary Experiments

- ROC curve generated in a set of preliminary experiments where we employed an alert stream composed of **84168 alerts** coming from **8848 different IPs** that was generated after **four months of real surveillance** in a networked organization using **3 Snort sensors, 18 sequential cases** corresponding to well-known attack patterns, an **error type weighting of 1:500** (i.e. a cost of 1 for each false positive and cost of 500 for each false negative), and **12 variants of 3 different multi-stage attacks**. The optimal decision threshold corresponded to the iso-performance line with slope equal to 2.2 as shown in the Figure.



# Questions?

```
No exact OS matches for host  
Nmap run completed -- 1 IP address (1 host up) scanned  
# sshnuke 10.2.2.2 -rootpw="210N0101"  
Connecting to 10.2.2.2:ssh ... successful  
Attempting to exploit: root ... successful.  
Resetting root password to "210N0101".  
System open: Access Level (9)  
# ssh 10.2.2.2 -l root  
root@10.2.2.2's password: |
```



No exact OS matches for host

Nmap run completed -- 1 IP address (1 host up) scanned

```
# sshnuke 10.2.2.2 -rootpw="210H0101"
```

Connecting to 10.2.2.2:ssh ... successful.

Attempting to exploit shell (CVE-2002-1232) ... successful.

Resetting root password to "210H0101".

System open: Access Level (9)

```
# ssh 10.2.2.2 -l root
```

```
root@10.2.2.2's password: |
```

Back-up



# Sequence Similarity

---

- *Commonalities* -- Comparable elements that appear at the same position in both sequences.
- *Alignable Differences* -- Comparable elements that appear in both sequences but at a different position.
- *Non-alignable Differences* -- Non-comparable elements that appear in one sequence but not in the other.

# Alignment Example

E	B	D
---	---	---

A	B	A	C	C	E	B
---	---	---	---	---	---	---

$$C = \{B\} \quad A = \{B, E\} \quad N = \{A, C, D\}$$

E	B	D
---	---	---

A	B	A	C	C	E	B
---	---	---	---	---	---	---

$$C = \{E, B\} \quad A = \{B\} \quad N = \{A, C, D\}$$

# Sequence alignment

- *Definition 5. (Sequence Alignment)* Given a signature  $\Sigma = \langle S, \perp, F, \leq \rangle$  and two sequences  $S_1, S_2 \in \Sigma^*$ . An alignment of sequences  $S_1$  and  $S_2$  is a pair  $\{S'_1, S'_2\}$  attained by insertion of a number of dummy feature terms ( $\perp$ ) in both sequences such that:  $|S'_1| = |S'_2|$  and  $\forall_{1 \leq i \leq |S'_1|} S'_1[i]$  is aligned with  $S'_2[i]$  and either  $S'_1[i]$  or  $S'_2[i]$  is not a dummy feature term.

# Sequence alignment example

E	B	D
---	---	---

A	B	A	C	D	E	A	B
---	---	---	---	---	---	---	---

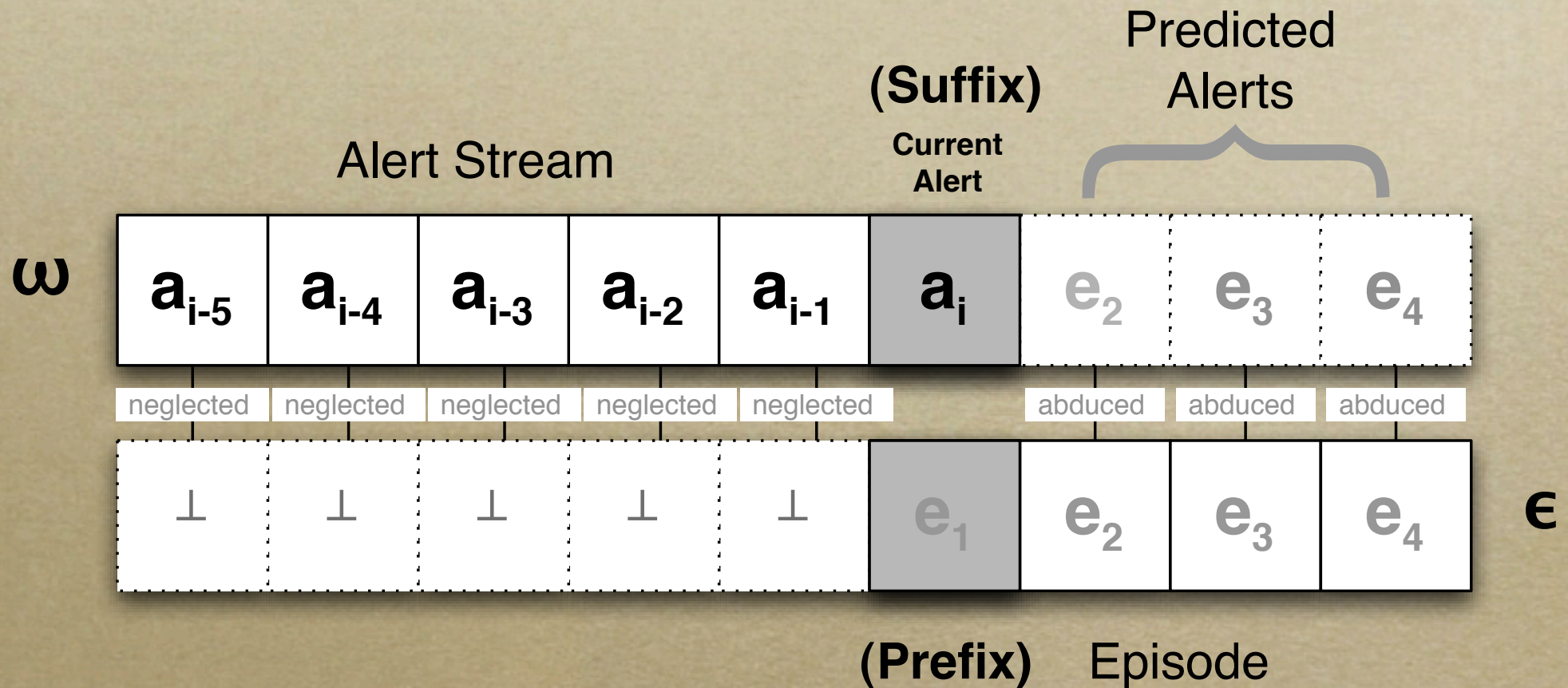
E	B	⊥	⊥	D	⊥	⊥	⊥
---	---	---	---	---	---	---	---

A	B	A	C	D	E	A	B
---	---	---	---	---	---	---	---

⊥	⊥	⊥	⊥	⊥	E	⊥	B	D
---	---	---	---	---	---	---	---	---




A	B	A	C	D	E	A	B	⊥
---	---	---	---	---	---	---	---	---

# Semi-global alignment



# Score of an alignment

$$S(S_1, S_2) = \sum_{1 \leq i \leq |S_1|} S(S_1[i], S_2[i])$$

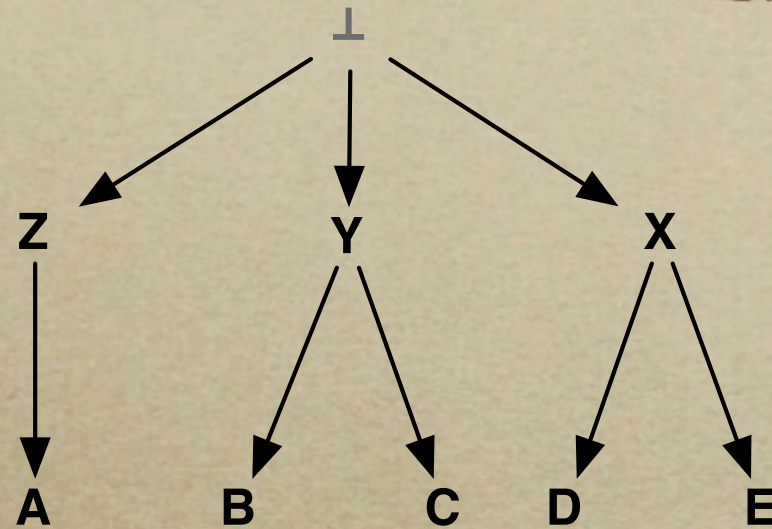
	Subsumption	Neglection	Abduction
$S_1$	<div><b>a</b></div>	<div><b>a</b></div>	<div><math>\perp</math></div>
	<div></div>	<div></div>	<div></div>
$S_2$	<div><b>e</b></div>	<div><math>\perp</math></div>	<div><b>e</b></div>

# Subsumption scoring scheme

- *Definition 7. (Subsumption Scoring Scheme).* Given the following signature  $\Sigma = \langle S, \perp, F, \leq \rangle$ , a subsumption scoring scheme  $\mathbf{M}$  is a square  $|S \cup \perp| \times |S \cup \perp|$  matrix such that:

$$\mathcal{M}_{i,j} = \begin{cases} \frac{1-q_i}{q_j} & \text{if } \psi^{a_i} \sqsubseteq \psi^{a_j} \\ -1 & \text{otherwise} \end{cases}$$

# Dynamic scoring scheme example



**W=3secs**

alerts

A	B	A	A	B	C	D	E	C	C	B	A	B	A	C	C	E	A	B
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

sorts

Z	Y	Z	Z	Y	Y	X	X	Y	Y	Y	Z	Y	Z	Y	Y	X	Z	Y
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

timestamp

1	1	2	3	4	4	5	5	5	6	7	9	9	9	10	11	11	12	12
---	---	---	---	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----

# Dynamic scoring scheme example

<i>M</i>	A	B	C	D	E	Z	Y	X	$\perp$
A	2.167	-1	-1	-1	-1	-1	-1	-1	-1
B	-1	2.8	-1	-1	-1	-1	-1	-1	-1
C	-1	-1	2.8	-1	-1	-1	-1	-1	-1
D	-1	-1	-1	18	-1	-1	-1	-1	-1
E	-1	-1	-1	-1	8.5	-1	-1	-1	-1
Z	2.167	-1	-1	-1	-1	2.167	-1	-1	-1
Y	-1	1.8	1.8	-1	-1	-1	0.9	-1	-1
X	-1	-1	-1	16	8	-1	-1	5.333	-1
$\perp$	0	0	0	0	0	0	0	0	0

$q_i$

A	0.316
B	0.263
C	0.263
D	0.053
E	0.105
Z	0.316
Y	0.526
X	0.158
$\perp$	1

# Abduction ( $C^a$ ) and Neglection ( $C^n$ ) Costs

- *Abduction( $S, a, i$ ): injects an alert  $a$  in alert stream  $S$  at position  $i$ .*
- $$C^a(a) = - \sum_{a' \in S: \text{root}(a) \sqsubseteq a'} \rho_\alpha(a')$$
- $$\rho_\alpha(a) = \frac{\alpha - \text{rare}(a)}{\#distinct}$$
- *Neglection( $S, i$ ): ignores an alert at position  $i$  from alert stream  $S$ .*

$$C^n(a) = -\rho_\alpha(a)^{-1}$$

# Abduction ( $C^a$ ) and Neglection ( $C^n$ ) Costs

	$C^a$	$C^n$
A	-1.2	-0.83
B	-1	-1
C	-1	-1
D	-0.2	-5
E	-0.4	-2.5
Z	-2.4	-0.83
Y	-4	-0.5
X	-1.2	-1.67
$\perp$	-11.4	-0.26

# Sequence Similarity

- **Definition 6. (Sequence Similarity).** The similarity between two sequences  $S_1$  and  $S_2$  is the score of the optimal alignment between a suffix of  $S_1$  and a prefix of  $S_2$ :  $S_1 \sim_s S_2 = \max_{1 \leq j \leq |S_2|} S(|S_1|, j)$ .

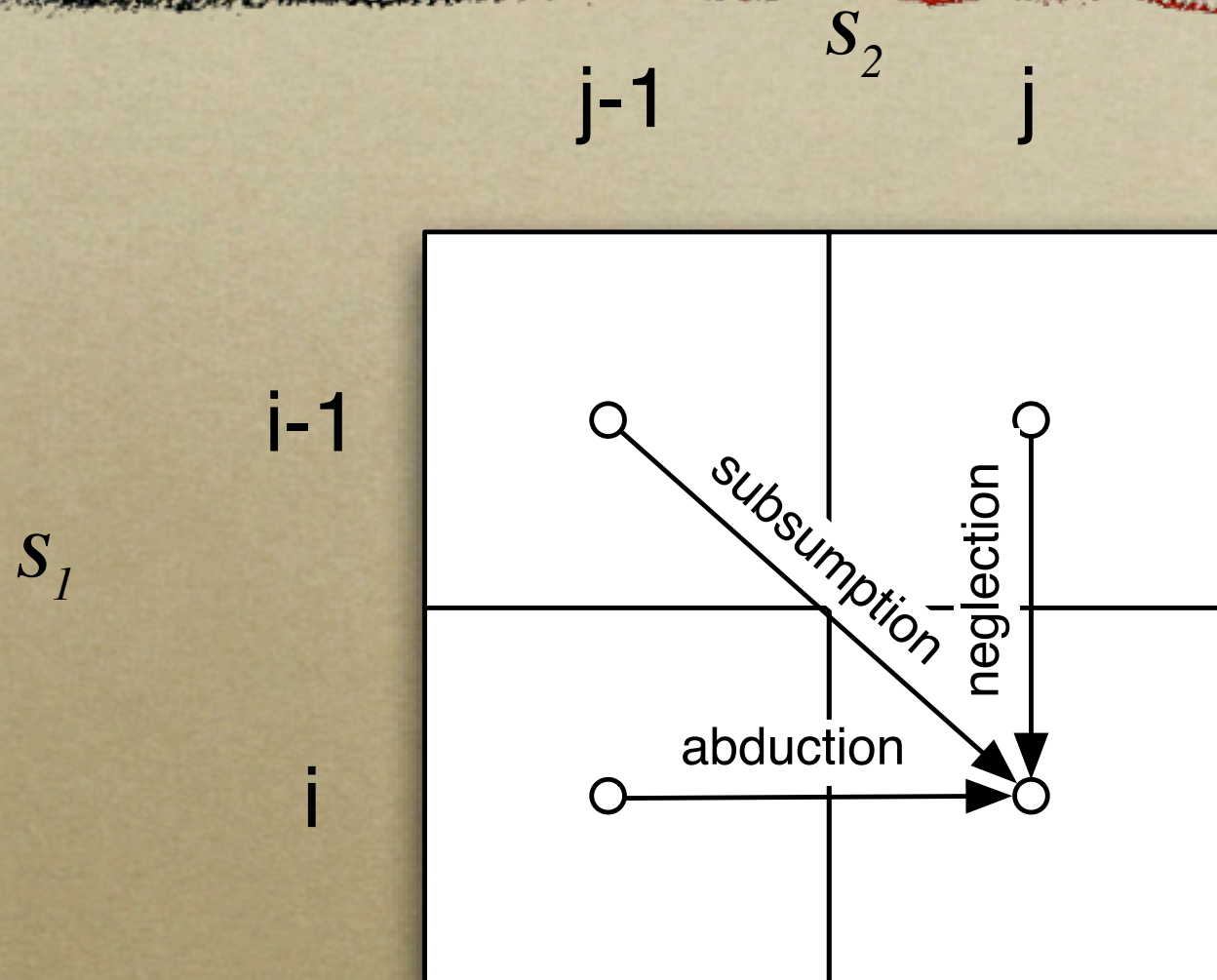
$$S(0, 0) = 0$$

$$S(i, 0) = S(i - 1, 0)$$

$$S(0, j) = S(0, j - 1) + C^a(\vec{S}_2[j])$$

$$S(i, j) = \max \begin{cases} S(i - 1, j) & + C^n(\vec{S}_1[i]) \\ S(i, j - 1) & + C^a(\vec{S}_2[j]) \\ S(i - 1, j - 1) & + \mathcal{M}(\text{root}(i), \text{root}(j)) \end{cases}$$

# Sequence similarity



# Dynamic programming trace

E B D

10.467 > □

neglecting(A)

neglecting(B)

neglecting(A)

neglecting(C)

neglecting(C)

subsuming(E)

neglecting(A)

subsuming(B)

		$S_2[0]$	$S_2[1]$	$S_2[2]$	$S_2[3]$
	$S_1[0]$	0	-0.4	-1.4	-1.6
A	$S_1[1]$	0	-0.4	-1.4	-1.6
B	$S_1[2]$	0	-0.4	2.4	2.2
A	$S_1[3]$	0	-0.4	1.567	1.4
C	$S_1[4]$	0	-0.4	0.567	0.5
C	$S_1[5]$	0	-0.4	-0.433	-0.433
E	$S_1[6]$	0	8.5	7.5	7.30
A	$S_1[7]$	0	7.67	7.5	7.30
B	$S_1[8]$	0	6.67	10.467	10.267

	$C^a$	$C^n$
A	-1.2	-0.83
B	-1	-1
C	-1	-1
D	-0.2	-5
E	-0.4	-2.5
Z	-2.4	-0.83
Y	-4	-0.5
X	-1.2	-1.67
⊥	-11.4	-0.26