

AGI-20 Tutorial

OpenCog, PLN and Pattern Miner

Nil Geisweiller, Matt Ikle

SingularityNET & OpenCog Foundations



SingularityNET



1 Install docker

- Debian/Ubuntu

```
sudo apt install docker.io
```

- Arch/Manjaro

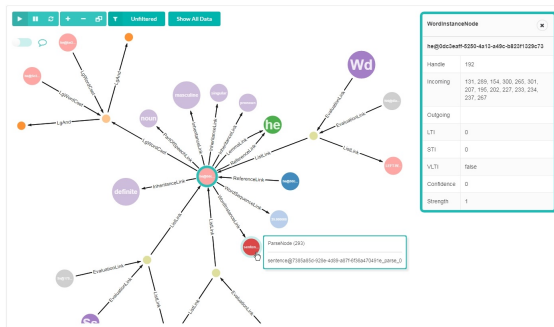
```
sudo pacman -S docker
```

2 Download docker image (1.6GB)

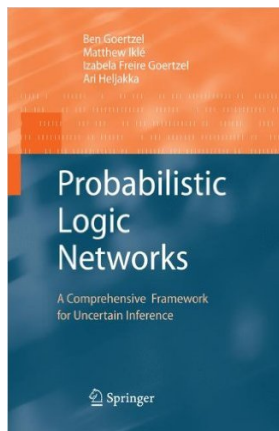
```
sudo docker pull ngeiswei/opencog:agi20
```

Framework for AGI

- 1 Hypergraph Database:
 - AtomSpace
 - Atomese: query, rewrite and more
- 2 Mind Agents (cognitive processes):
 - Reasoning: PLN, Miner
 - Learning: MOSES, Miner
 - Decision: OpenPsi (Bach's MicroPsi)
 - Language Processing
 - Attention Allocation



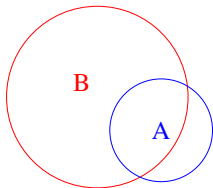
Download docker image: `sudo docker pull ngeiswei/opencog:agi20`



- Probability Theory
- Uncertainty management
- Common sense reasoning
- Mathematical reasoning
- Resource management

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

```
(Subset (stv s c)
  A
  B)
```

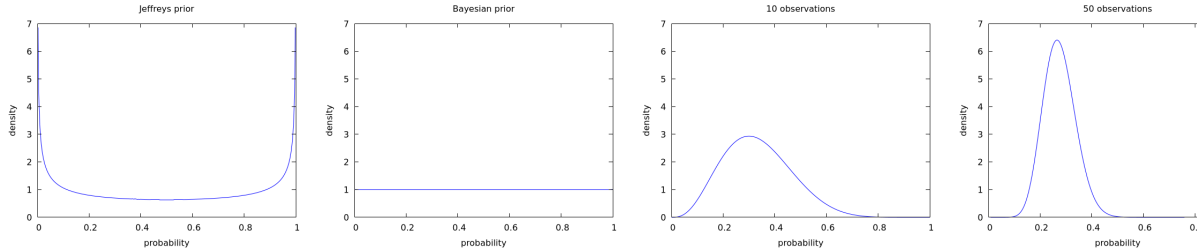


Definitions:

- $stv = \text{Simple Truth Value}$
- $\text{Subset } A \ B = P(B|A)$
- $s = \text{strength} = P(B|A) = \frac{|A \cap B|}{|A|}$
- $c = \text{confidence} = \frac{|A|}{|A| + K}$

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

Truth Value = Second Order Distribution



Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

- Knowledge base:

(Subset (stv 0.4 0.1) A B)

(Subset (stv 0.6 0.2) B C)

A (stv 0.1 0.6)

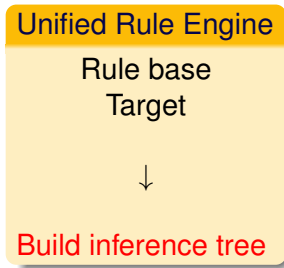
- Rule base:

$$\frac{(\text{Subset } \langle \text{tv1} \rangle \text{ X Y}) \quad (\text{Subset } \langle \text{tv2} \rangle \text{ Y Z})}{(\text{Subset } \langle \text{tv3} \rangle \text{ X Z})} \text{ (Deduction)}$$

$$\frac{\text{X } \langle \text{tv1} \rangle \quad (\text{Subset } \langle \text{tv2} \rangle \text{ X Y})}{\text{Y } \langle \text{tv3} \rangle} \text{ (Modus Ponens)}$$

- Target: C <?>

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`



$$\begin{array}{c}
 \frac{A \text{ <?>} \quad \frac{(\text{Subset } \text{<?>} \text{ A B}) \quad (\text{Subset } \text{<?>} \text{ A B})}{(\text{Subset } \text{<?>} \text{ A C})} \text{ (Deduction)} \\
 \hline
 C \text{ <?>} \text{ (Modus Ponens)}
 \end{array}$$

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

Basic:

- Deduction
- Modus Ponens
- Conjunction/Disjunction/Negation Introduction
- Universal Instantiation
- ...

Advanced:

- Intensional
- Contextual
- Temporal
- Spatial
- ...

Advanced built on basic

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

- Context-free control rule:

```
(Implication <TV>  
  <rule>  
  <good-inference>)
```

- Context-sensitive control rule:

```
(Implication <TV>  
  (And  
    <inference-context>  
    <rule>)  
  <good-inference>)
```

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

- Knowledge base:

(Subset A B)

(Subset A C)

(Subset A D)

(Subset B E)

(Subset C E)

(Subset D E)

...

- Frequent Pattern (minimum support = 3):

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

- Knowledge base:

(Subset A B)

(Subset A C)

(Subset A D)

(Subset B E)

(Subset C E)

(Subset D E)

...

- Frequent Pattern (minimum support = 3):

(Subset A X)

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

- Knowledge base:

(Subset A B)

(Subset A C)

(Subset A D)

(Subset B E)

(Subset C E)

(Subset D E)

...

- Frequent Pattern (minimum support = 3):

(Subset A X)

(Subset X E)

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

- Knowledge base:

(Subset A B)

(Subset A C)

(Subset A D)

(Subset B E)

(Subset C E)

(Subset D E)

...

- Frequent Pattern (minimum support = 3):

(Subset A X)

(Subset X E)

(And (Subset A X) (Subset X E))

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

Brute force algorithm:

- \mathcal{K} : *knowledge base*
 - S : *minimum support*
 - \mathcal{C} : *pattern pool*
 - P, Q : *patterns*
- 1 Select P from \mathcal{C}
 - 2 Select *specialization* Q of P such that $S \leq \text{support}(Q, \mathcal{K})$
 - 3 Add Q to \mathcal{C}
 - 4 Repeat

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

$$\frac{S \leq \text{support}(Q, \mathcal{D}) \quad \text{spec}(Q, P)}{S \leq \text{support}(P, \mathcal{D})} \text{ (AP=A Priory Property)}$$

$$\frac{S \leq \text{support}(P, \mathcal{D}) \quad \text{spec}(P, \text{Top})}{S \leq \text{support}(\text{Top}, \mathcal{D})} \text{ (AP)}$$

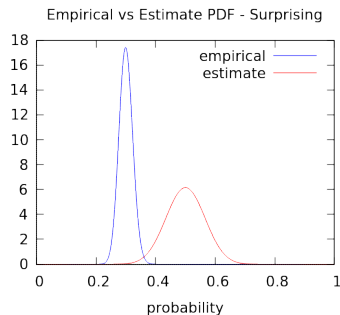
$$\Downarrow$$

$$\frac{\frac{S \leq \text{support}(Q, \mathcal{D}) \quad \text{spec}(Q, P)}{S \leq \text{support}(P, \mathcal{D})} \text{ (AP)} \quad \text{spec}(P, \text{Top}) \text{ (AP)}}{S \leq \text{support}(\text{Top}, \mathcal{D})} \text{ (AP)}$$

$$\Downarrow$$

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

surprising: **contrary to expectation**



Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

$$\frac{S \leq \text{support}(P, \mathcal{D})}{\text{surprising}(P, \mathcal{D}, \text{jsd}(P, \mathcal{D}))} \text{ (S)} \quad \frac{\frac{\frac{P}{\text{emp}(P, \mathcal{D})} \text{ (DE)} \quad \frac{?}{\text{est}(P, \mathcal{D})}}{\text{jsd}(P, \mathcal{D})} \text{ (JSD)}}{\text{surprising}(P, \mathcal{D}, \text{jsd}(P, \mathcal{D}))} \text{ (S)}$$

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

$$\frac{S \leq \text{support}(P, \mathcal{D})}{\text{surprising}(P, \mathcal{D}, \text{jsd}(P, \mathcal{D}))} \text{ (S)} \quad \frac{\frac{P \quad \mathcal{D}}{\text{emp}(P, \mathcal{D})} \text{ (DE)} \quad \frac{P \quad \mathcal{D}}{\text{est}(P, \mathcal{D})} \text{ (IS)}}{\text{jsd}(P, \mathcal{D})} \text{ (JSD)}$$

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

$$\frac{S \leq \text{support}(P, \mathcal{D})}{\text{surprising}(P, \mathcal{D}, \text{jsd}(P, \mathcal{D}))} \text{ (S)}$$

$$\frac{\frac{P \quad \mathcal{D}}{\text{emp}(P, \mathcal{D})} \text{ (DE)} \quad \frac{\frac{\vdots \quad \vdots}{\vdots} \quad \frac{\vdots \quad \vdots}{\text{est}(P, \mathcal{D})} \text{ (JSD)}}{\text{jsd}(P, \mathcal{D})}$$

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

$$\begin{array}{c}
 \frac{S \leq \text{support}(P, \mathcal{D})}{\text{surprising}(P, \mathcal{D}, \text{jsd}(P, \mathcal{D}))} \text{ (S)} \\
 \frac{\frac{P \quad \mathcal{D}}{\text{emp}(P, \mathcal{D})} \text{ (DE)} \quad \frac{\frac{\frac{\text{emp}(Q, \mathcal{D})}{\vdots} \quad \vdots}{\vdots} \quad \vdots}{\text{est}(P, \mathcal{D})} \text{ (JSD)}}{\text{jsd}(P, \mathcal{D})}
 \end{array}$$

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

$$\begin{array}{c}
 \frac{S \leq \text{support}(P, \mathcal{D})}{\text{surprising}(P, \mathcal{D}, \text{jsd}(P, \mathcal{D}))} \text{ (S)} \\
 \frac{\frac{P \quad \mathcal{D}}{\text{emp}(P, \mathcal{D})} \text{ (DE)} \quad \frac{\frac{\text{emp}(Q, \mathcal{D})}{\vdots} \quad \vdots}{\frac{\vdots}{\text{est}(P, \mathcal{D})} \text{ (JSD)}}}{\text{jsd}(P, \mathcal{D})}
 \end{array}$$

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`

Tutorial time

Download docker image: `sudo docker pull ngeiswei/opencog:agi20`