AGI-20 Tutorial
OpenCog, PLN and Pattern Miner

Nil Geisweiller, Matt Ikle

SingularityNET & OpenCog Foundations

1. Install docker
   - Debian/Ubuntu

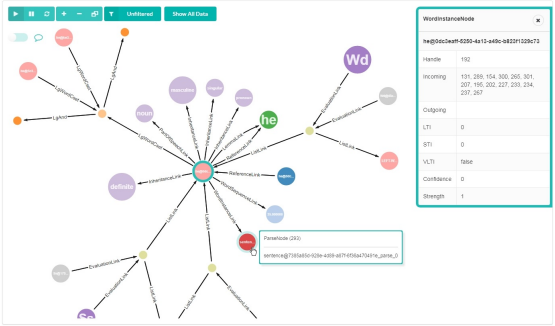
     ```
     sudo apt install docker.io
     ```
   - Arch/Manjaro

     ```
     sudo pacman -S docker
     ```
2. Download docker image (1.6GB)

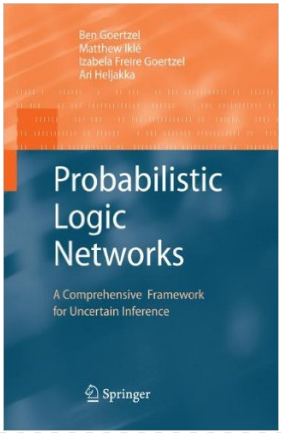
   ```
   sudo docker pull ngeiswei/opencog:agi20
   ```

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



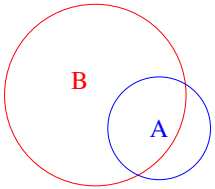
Download docker image: `sudo docker pull ngeiswei/opencog:agi20`



- 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)
```

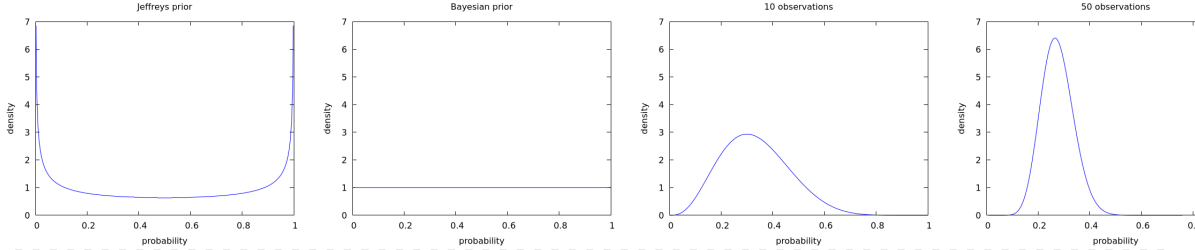


Definitions:

- `stv` = *Simple Truth Value*
- `Subset A B` = $P(B|A)$
- `s` = strength = $P(B|A) = \frac{|A \cap B|}{|A|}$
- `c` = 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)
```

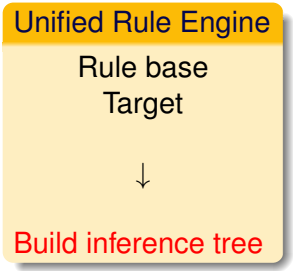
- Rule base:

$$\frac{\texttt{(Subset <tv1> X Y)} \qquad \texttt{(Subset <tv2> Y Z)}}{\texttt{(Subset <tv3> X Z)}} \text{ (Deduction)}$$

$$\frac{\texttt{X <tv1>} \qquad \texttt{(Subset <tv2> X Y)}}{\texttt{Y <tv3>}} \text{ (Modus Ponens)}$$

- Target: `C <?>`

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



$$\dfrac{\dfrac{\texttt{(Subset <?> A B)} \qquad \texttt{(Subset <?> A B)}}{\texttt{A <?>} \qquad \texttt{(Subset <?> A C)}}\ \text{(Deduction)}}{\texttt{C <?>}}\ \text{(Modus Ponens)}$$

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`

- 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 $S \leq \texttt{support}(Q, \mathcal{K})$
3. Add Q to $\mathcal{C}$
4. Repeat

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

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

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

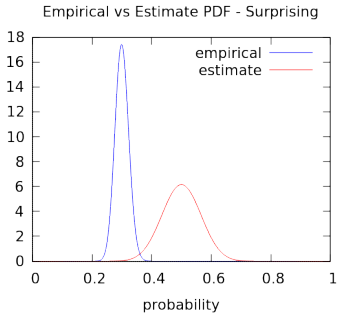
$$\Downarrow$$

$$\frac{\dfrac{S \leq \texttt{support}(Q, \mathcal{D}) \qquad \texttt{spec}(Q, P)}{S \leq \texttt{support}(P, \mathcal{D})}\ \text{(AP)} \qquad \texttt{spec}(P, \mathit{Top})}{S \leq \texttt{support}(\mathit{Top}, \mathcal{D})}\ \text{(AP)}$$

$$\Downarrow$$

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

surprising: contrary to expectation



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

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

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

$$\dfrac{\dfrac{\dfrac{P \quad \mathcal{D}}{\mathtt{emp}(P,\mathcal{D})}\,(\mathrm{DE}) \qquad \dfrac{P \quad \mathcal{D}}{\mathtt{est}(P,\mathcal{D})}\,(\mathrm{IS})}{\mathtt{jsd}(P,\mathcal{D}))}\,(\mathrm{JSD}) \qquad S \leq \mathtt{support}(P,\mathcal{D})}{\mathtt{surprising}(P,\mathcal{D},\mathtt{jsd}(P,\mathcal{D}))}\,(\mathrm{S})$$

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

$$\dfrac{S \leq \texttt{support}(P,\mathcal{D}) \qquad \dfrac{\dfrac{P \qquad \mathcal{D}}{\texttt{emp}(P,\mathcal{D})}\,(\text{DE}) \qquad \dfrac{\vdots \qquad \vdots}{\texttt{est}(P,\mathcal{D})}\,(\text{JSD})}{\texttt{jsd}(P,\mathcal{D}))}}{\texttt{surprising}(P,\mathcal{D},\texttt{jsd}(P,\mathcal{D}))}\,(\text{S})$$

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

$$\dfrac{S \leq \texttt{support}(P,\mathcal{D}) \qquad \dfrac{\dfrac{P \qquad \mathcal{D}}{\texttt{emp}(P,\mathcal{D})}\,(\text{DE}) \qquad \dfrac{\dfrac{\texttt{emp}(Q,\mathcal{D})}{\vdots \qquad \vdots} \qquad \dfrac{\vdots \qquad \vdots}{\texttt{est}(P,\mathcal{D})}}{\texttt{jsd}(P,\mathcal{D}))}\,(\text{JSD})}{\texttt{surprising}(P,\mathcal{D},\texttt{jsd}(P,\mathcal{D}))}\,(\text{S})$$

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

$$\dfrac{S \leq \texttt{support}(P,\mathcal{D}) \qquad \dfrac{\dfrac{P \qquad \mathcal{D}}{\texttt{emp}(P,\mathcal{D})}\,(\text{DE}) \qquad \dfrac{\dfrac{\texttt{emp}(Q,\mathcal{D})}{\vdots \qquad \vdots} \qquad \dfrac{\vdots \qquad \vdots}{\texttt{est}(P,\mathcal{D})}}{\texttt{jsd}(P,\mathcal{D}))}\,(\text{JSD})}{\texttt{surprising}(P,\mathcal{D},\texttt{jsd}(P,\mathcal{D}))}\,(\text{S})$$

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

Tutorial time

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