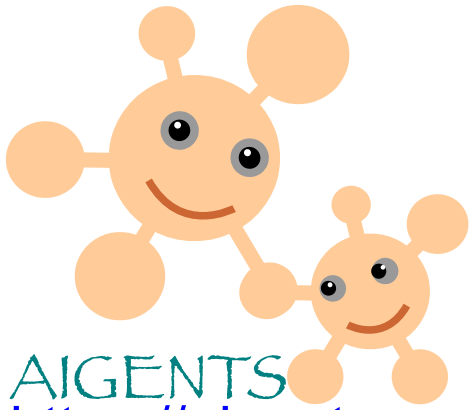
Data Structures and Languages for General Intelligence

Anton Kolonin
akolonin@aigents.com
Facebook: akolonin
Telegram: akolonin



https://aigents.com

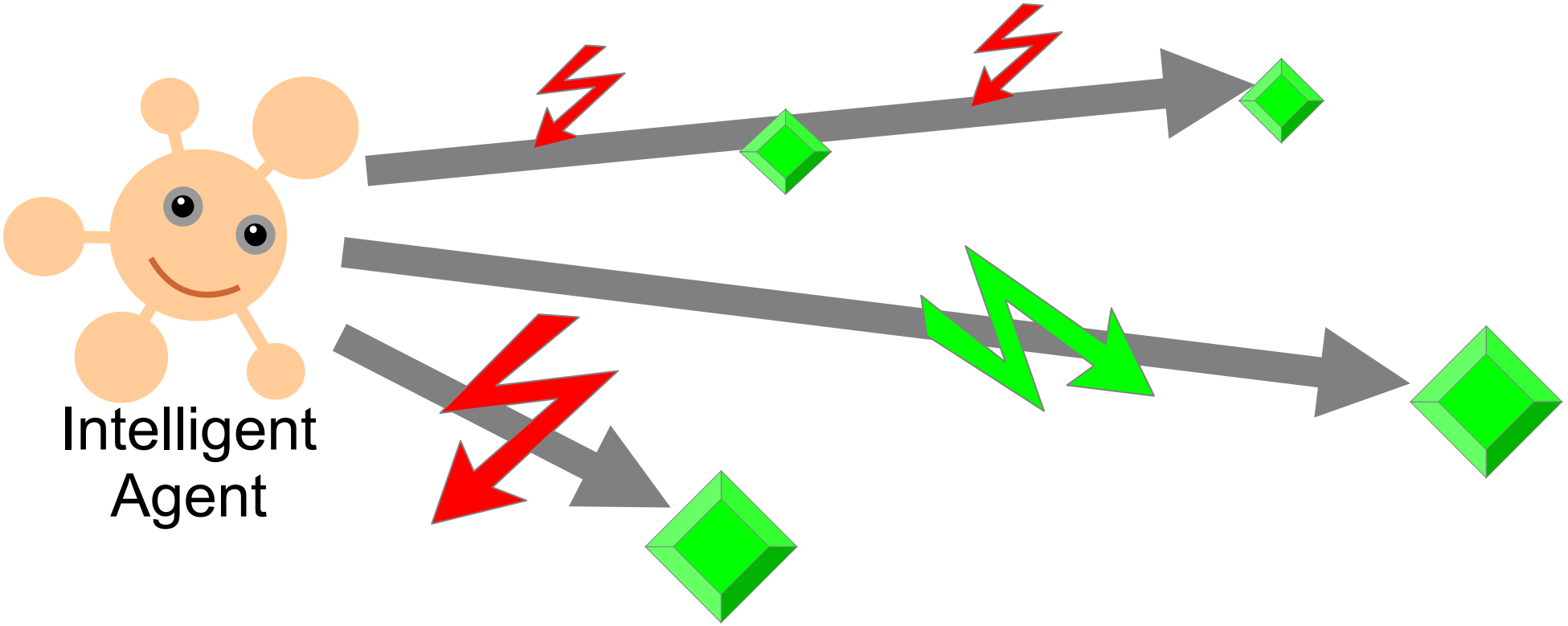
https://facebook.com/groups/agirussia
https://t.me/agirussia

Novosibirsk State University
*THE REAL SCIENCE

SingularityNET
https://singularitynet.io

General Intelligence:
Reaching complex goals in different complex environments, using limited resources and minimizing risks

(Ben Goertzel + Pei Wang + Shane Legg + Marcus Hutter)

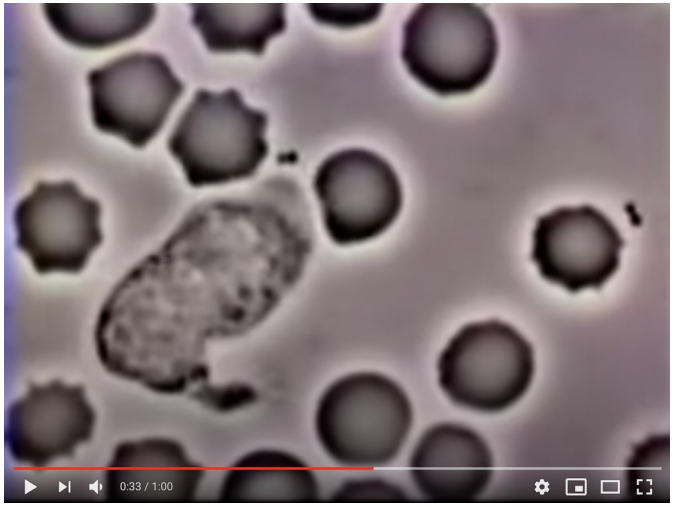


What is the minimally viable natural system capable to satisfy the requirement?

Complex nervous system

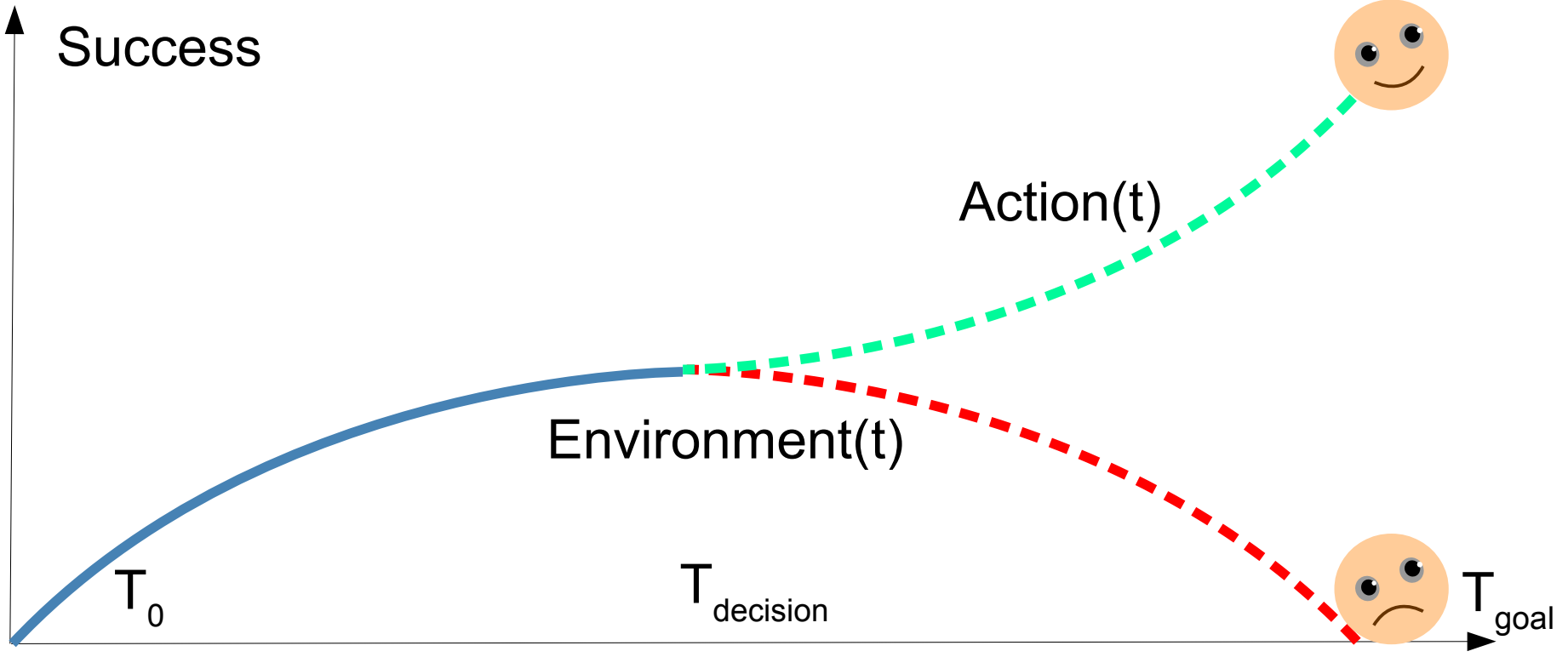
Simple nervous system

Single cell organism



Consciousness:

Ability to build models of the environment based on the past to predict the future scenarios and act “consciously” towards the desired ones



Acting consciously:
Agent being able to direct the sequence of behavioral acts (at least) to itself by any (internal or external) **language** (ontology/lexicon/grammar)

Push legs,
pull right hand

push(right_leg);
push(left_leg);
pull(right_hand).

Push left leg,
pull both hands

push(left_leg);
pull(right_hand);
pull(left_hand).

Pull left hand,
look right.

look(right);
pull(left_hand).

Pull right hand,
bend both legs.

pull(right_hand);
bend(right_leg);
bend(left_leg).

https://www.youtube.com/watch?v=oYWjqTP0AKM

Learning with experiential feedback

push(right_leg);
push(left_leg);
push(right_hand).
	feel(upset).			

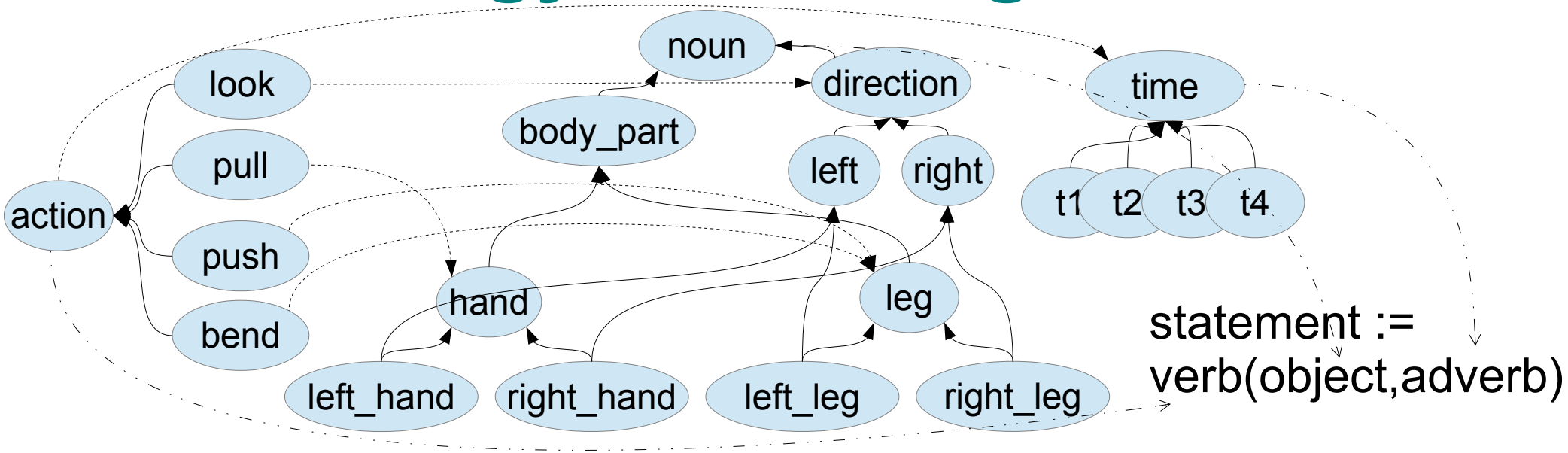
push(right_leg);
push(left_leg);
pull(right_hand).
	push(left_leg);
	pull(right_hand).
		feel(upset).		

push(right_leg);
push(left_leg);
pull(right_hand).
	push(left_leg);
	pull(right_hand);
	pull(left_hand).
		pull(right_hand).	feel(upset).	

push(right_leg);
push(left_leg);
pull(right_hand).
	push(left_leg);
	pull(right_hand);
	pull(left_hand).
		look(right);
		pull(left_hand).
			pull(right_hand);
			bend(right_leg);
			bend(left_leg).
				feel(lucky).

https://www.youtube.com/watch?v=oYWjqTP0AKM

Ontology, lexicon, grammar



push(right_leg, t1);
push(left_leg, t1);
pull(right_hand, t1).

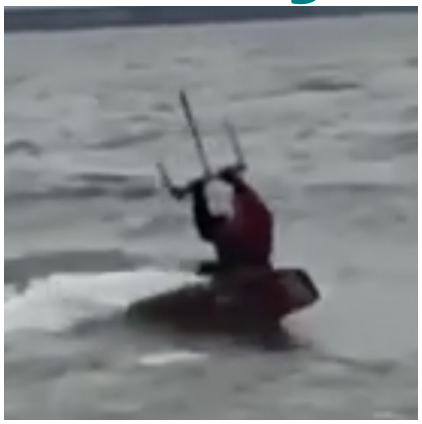
push(left_leg, t2);
pull(right_hand, t2);
pull(left_hand, t2);
greater(t2,t1);

look(right,t3);
pull(left_hand,t3);
greater(t3,t2);

pull(right_hand,t4);
bend(right_leg,t4);
bend(left_leg,t4);
greater(t4,t3).

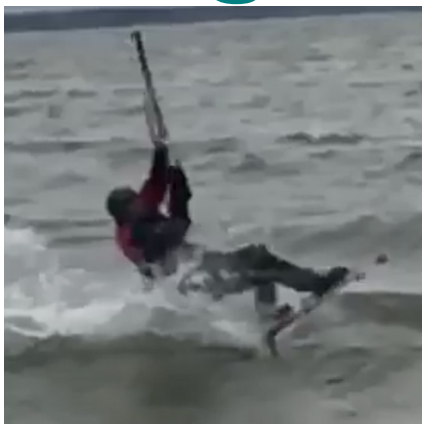
https://www.youtube.com/watch?v=oYWjqTP0AKM

N-ary and higher-order predicates



push(right_leg, t1);
push(left_leg, t1);
pull(right_hand, t1).

push(right, leg, t1);
push(left, leg, t1);
pull(right ,hand, t1).



push(left_leg, t2);
pull(right_hand, t2);
pull(left_hand, t2);
greater(t2,t1);

push(left, leg, t2);
pull(right, hand, t2);
pull(left, hand, t2);
greater(t2,t1);

look(right,t3);
pull(left_hand,t3);
greater(t3,t2);

look(right,t3);
pull(left, hand,t3);
greater(t3,t2);

pull(right_hand,t4);
bend(right_leg,t4);
bend(left_leg,t4);
greater(t4,t3).

pull(right, hand,t4);
bend(right, leg,t4);
bend(left, leg,t4);
greater(t4,t3).

```
before(
 and(
  push(right, leg),
  push(left, leg),
  pull(right ,hand)),
 and(
  push(left, leg),
  pull(right, hand),
  pull(left, hand))).
```

```
before(
 and(
  push(left, leg),
  pull(right, hand),
  pull(left, hand)),
 and(
  look(right),
  pull(left, hand))).
```

```
before(
 and(
  look(right),
  pull(left, hand)),
 and(
  pull(right, hand,t4),
  bend(right, leg,t4),
  bend(left, leg,t4))).
```

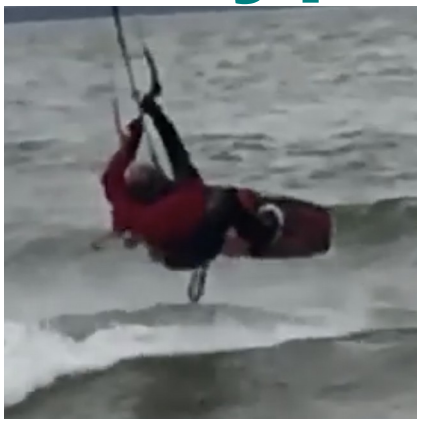
Making predicates untyped binary

push(right_leg, t1);
push(left_leg, t1);
pull(right_hand, t1).

($A,push**);**
($A,right_leg);
($A,t1);
($B,push**);**
($B,left_leg);
($B,t1);
($C,pull**);**
($C,right_hand);
($C,t2).

push(left_leg, t2);
pull(right_hand, t2);
pull(left_hand, t2);
greater(t2,t1).

($D,push);
($D,left_leg);
($D,t2);
($E,pull);
($E,right_hand);
($E, t2);
($F,pull);
($F,left_hand);
($F,t2);
(t2,t1).



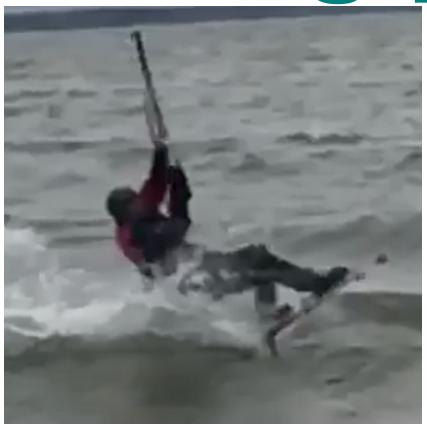
look(right,t3);
pull(left_hand,t3);
greater(t3,t2).

($G,look);
($G,right)
($G,t3);
($H,pull);
($H,left_hand)
($H,t3);
(t3,t2).

pull(right_hand,t4);
bend(right_leg,t4);
bend(left_leg,t4);
greater(t4,t3).

($I,pull);
($I,right_hand);
($I,t4);
($J,bend);
($J,right_leg)
($J,t4);
($K,bend);
($K,left_leg);
($K,t4);
(t4,t3).

Attaching probabilities

push(right_leg, t1);
push(left_leg, t1);
pull(right_hand, t1).

push(left_leg, t2);
pull(right_hand, t2);
pull(left_hand, t2);
greater(t2,t1).

look(right,t3);
pull(left_hand,t3);
greater(t3,t2).

pull(right_hand,t4);
bend(right_leg,t4);
bend(left_leg,t4);
greater(t4,t3).

push(right_leg, t1)
<0.7>;
push(left_leg, t1)
<0.7>;
pull(right_hand, t1)
<1.0>.

push(left_leg, t2)
<1.0>;
pull(right_hand, t2)
<1.0>;
pull(left_hand, t2)
<0.8>;
greater(t2,t1,1.0)<1.0>;

look(right,t3)
<0.9>;
pull(left_hand,t3)
<1.0>;
greater(t3,t2)<1.0>;

pull(right_hand,t4);
<1.0>
bend(right_leg,t4)
<0.6>;
bend(left_leg,t4)
<0.6>;
greater(t4,t3)<1.0>.

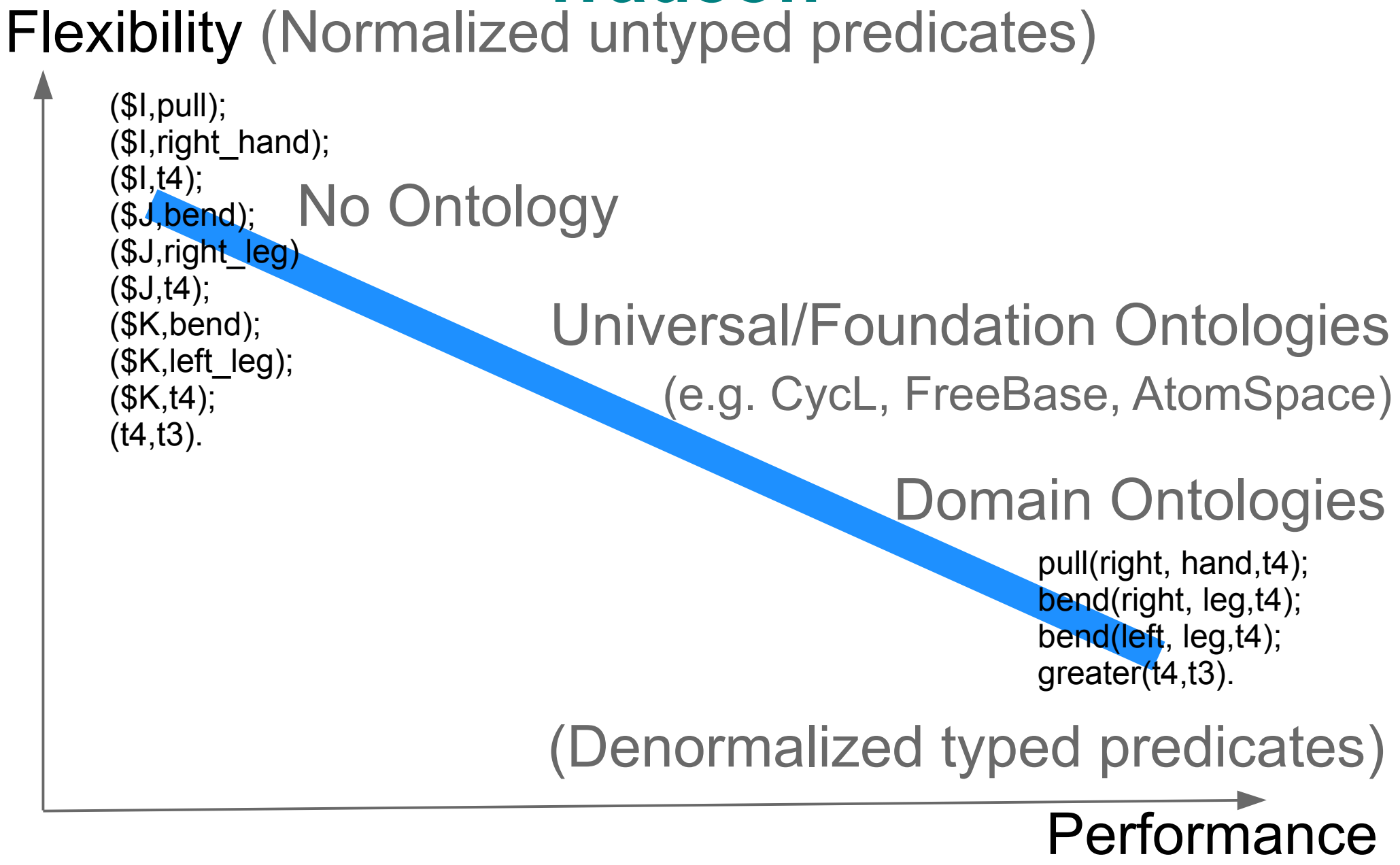
push(right_leg,t1,0.7);
push(left_leg,t1,0.7);
pull(right_hand,t1,1.0).

push(left_leg, t2,1.0);
pull(right_hand, t2,1.0);
pull(left_hand, t2,0.8);
greater(t2,t1,1.0);

look(right,t3,0.9);
pull(left_hand,t3,1.0);
greater(t3,t2,1.0);

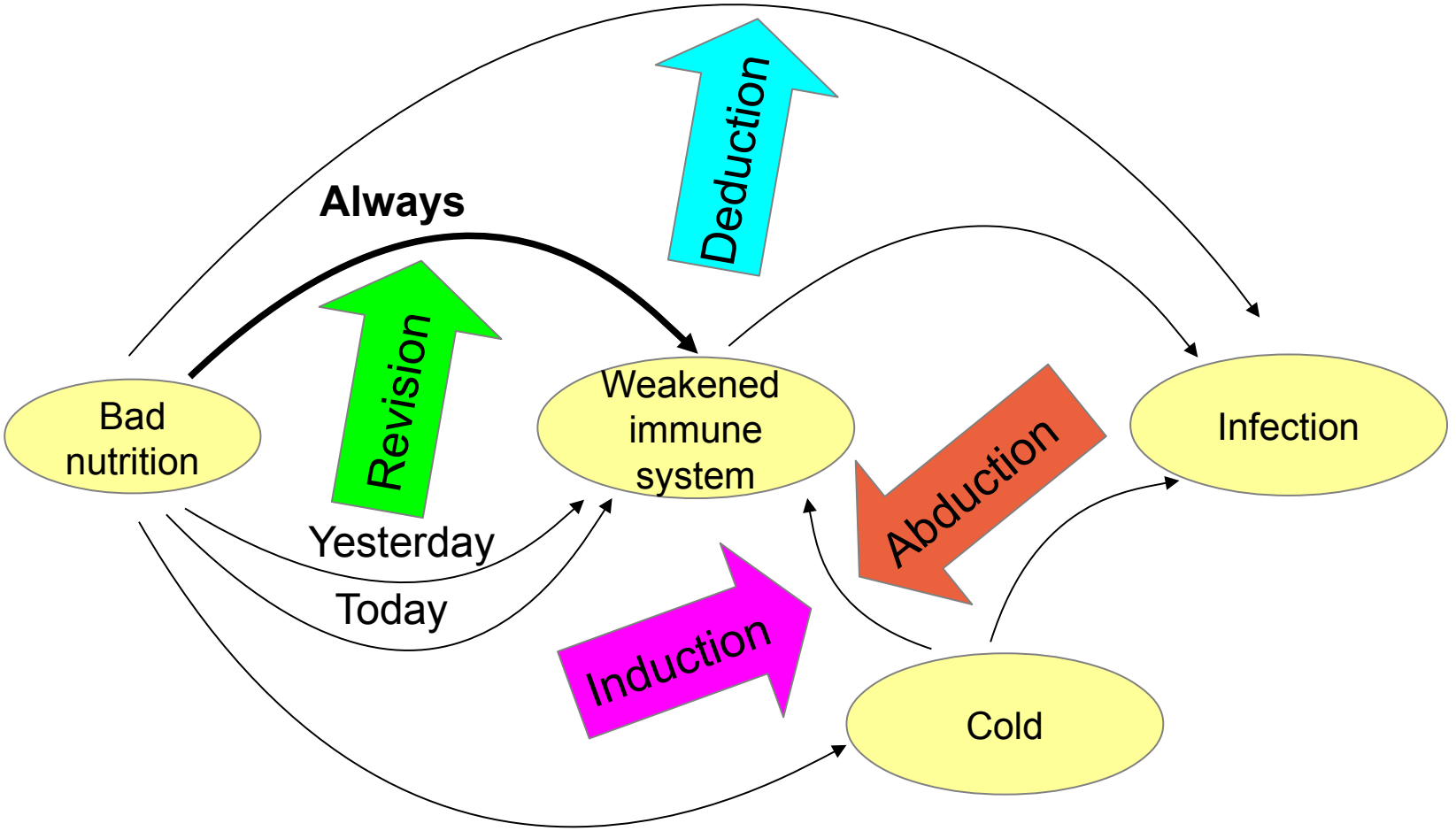
pull(right_hand,t4,1.0);
bend(right_leg,t4,0.6);
bend(left_leg,t4.0.6);
greater(t4,t3,1.0).

Predicate/Link Arity and Normalization Tradeoff



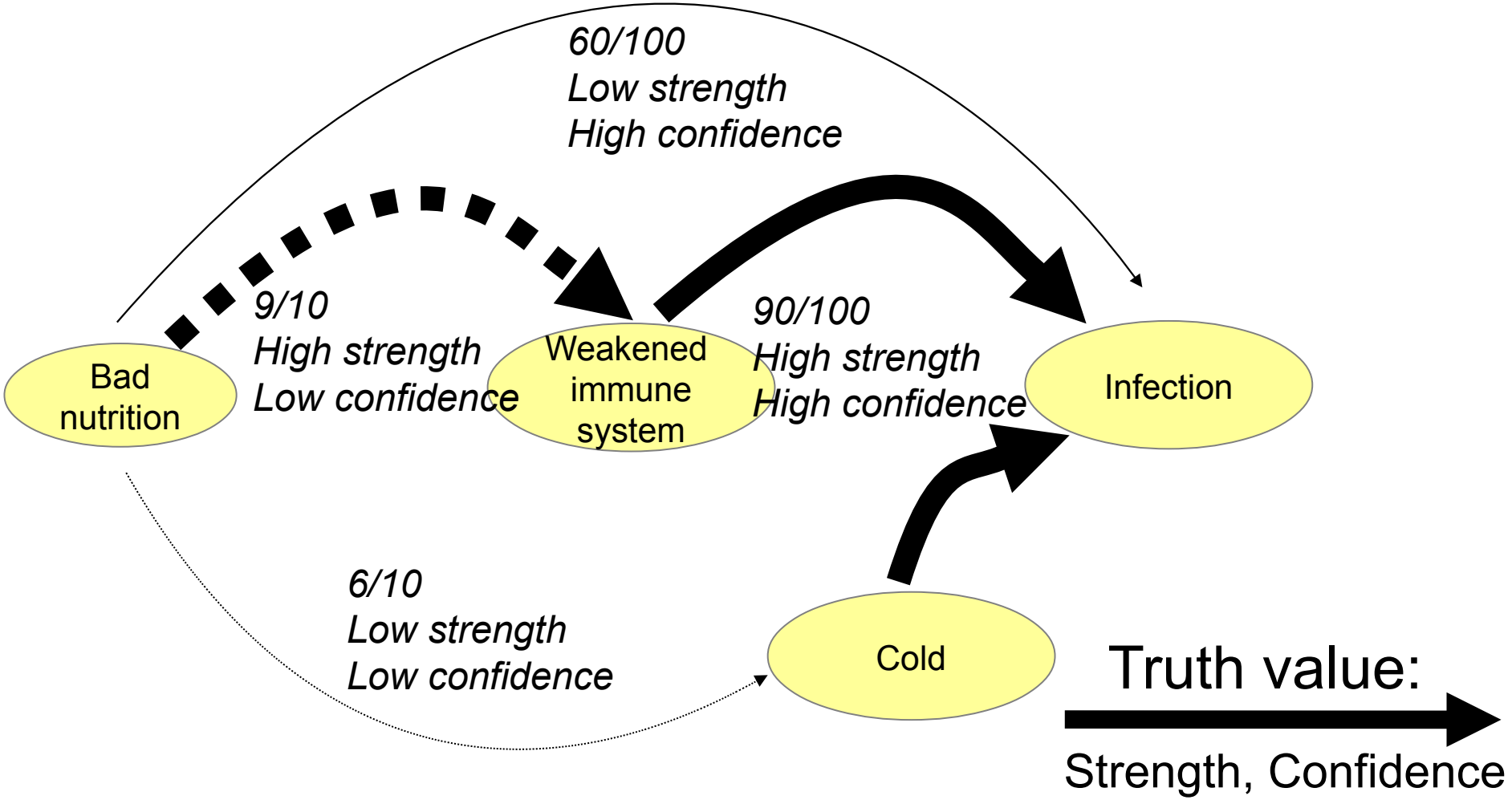
Making Probabilistic Inferences based on Predicates

P.Wang, et. al. (1993-2020) Non-Axiomatic Reasoning System (NARS)
E.Vityaev et.al. (2006-2020) Logical Prediction System ("Discovery")
B. Goertzel et. al. (2008-2020) Probabilistic Logic Network (PLN)



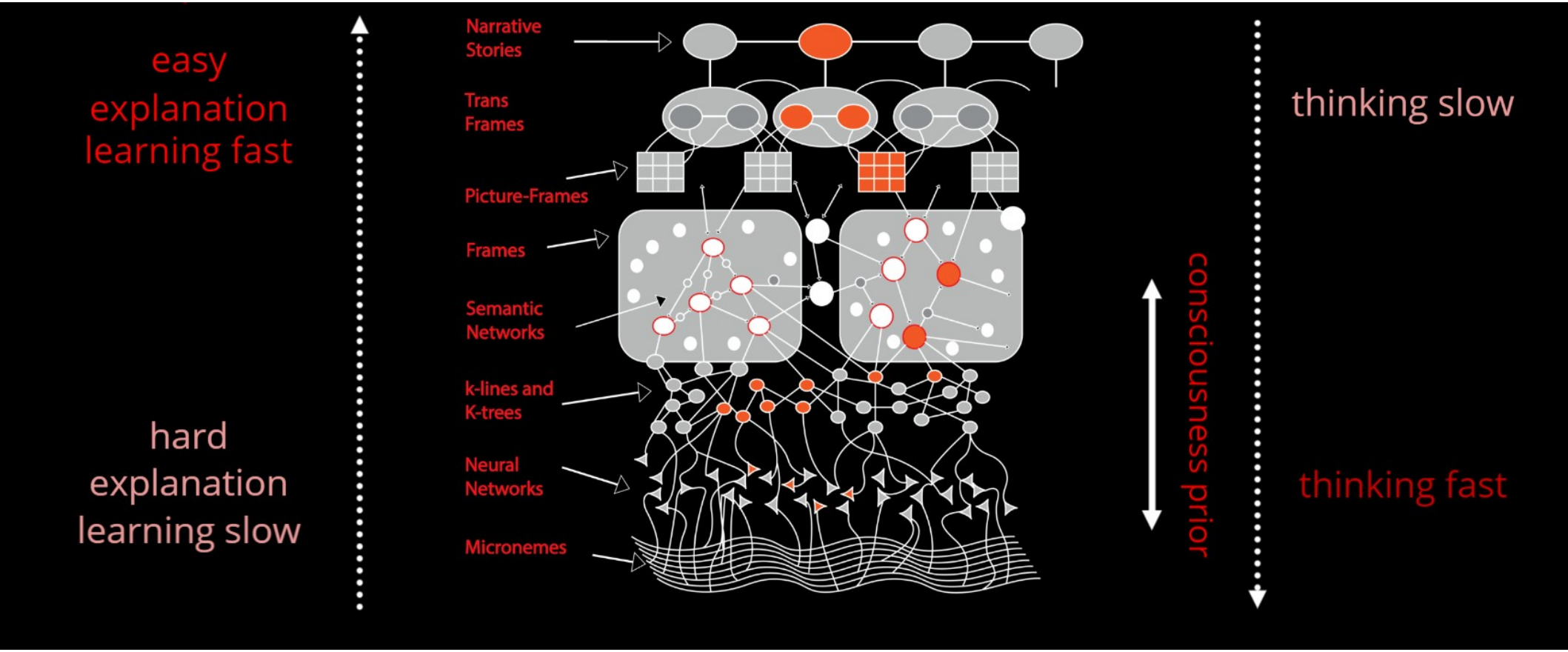
Replacing Probabilities with Compound Truth Values

P.Wang, et. al. (1993-2020) Non-Axiomatic Reasoning System (NARS)
B. Goertzel et. al. (2008-2020) Probabilistic Logic Network (PLN)



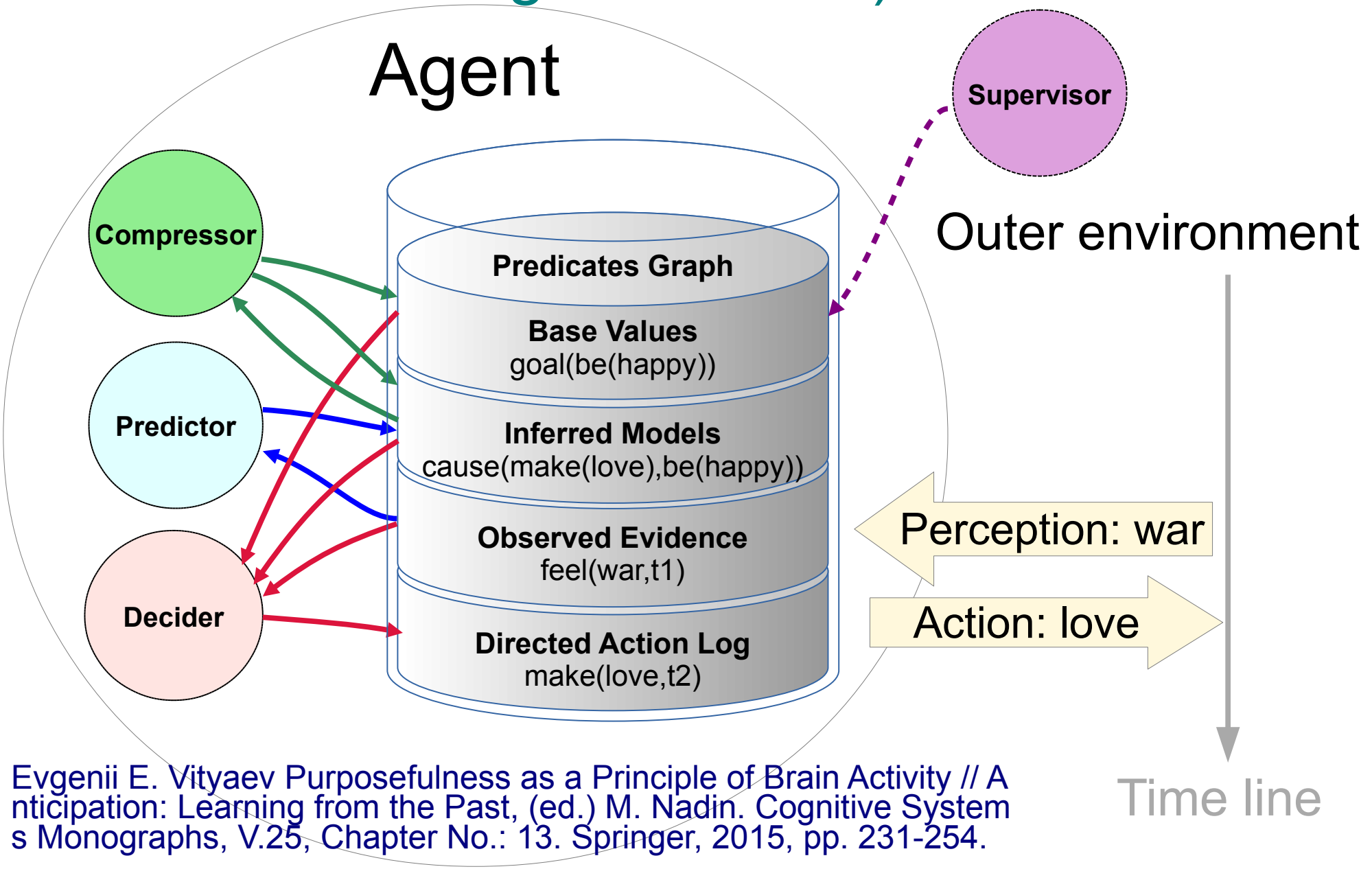
Hybrid Neuro-Symbolic Cognitive Architectures

Society of Mind – Marvin Minsky
Thinking, Fast and Slow – Daniel Kahneman



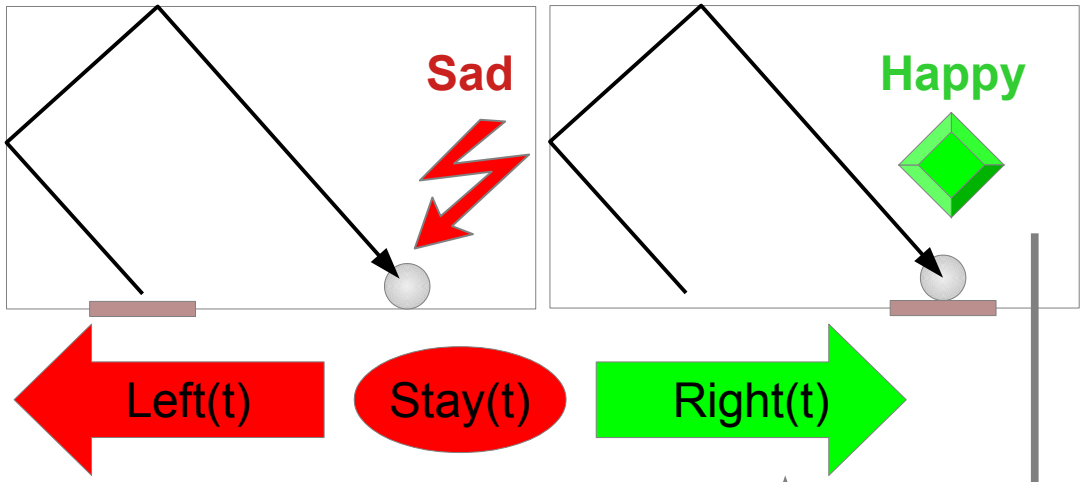
https://towardsdatascience.com/explainable-ai-vs-explaining-ai-part-1-d39ea5053347

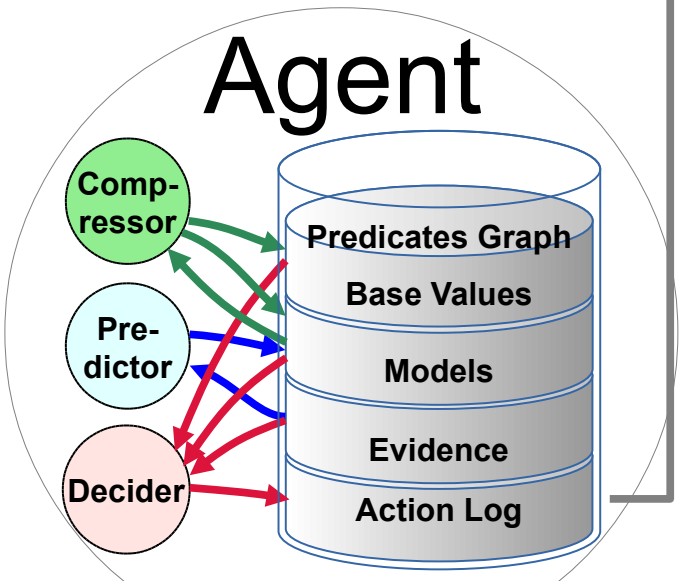
AGI Agent Cognitive Architecture (and AGI-unit testing framework) based on TFS



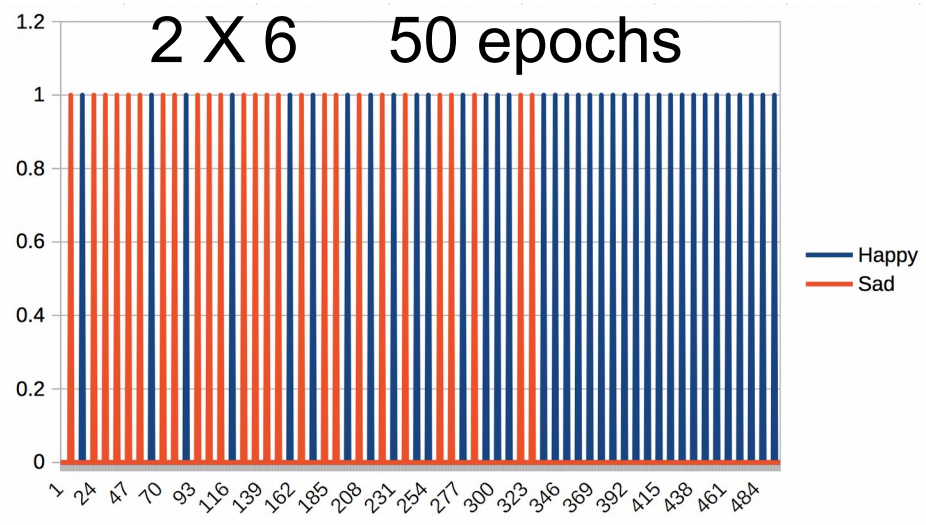
Evgenii E. Vityaev Purposefulness as a Principle of Brain Activity // Anticipation: Learning from the Past, (ed.) M. Nadin. Cognitive Systems Monographs, V.25, Chapter No.: 13. Springer, 2015, pp. 231-254.

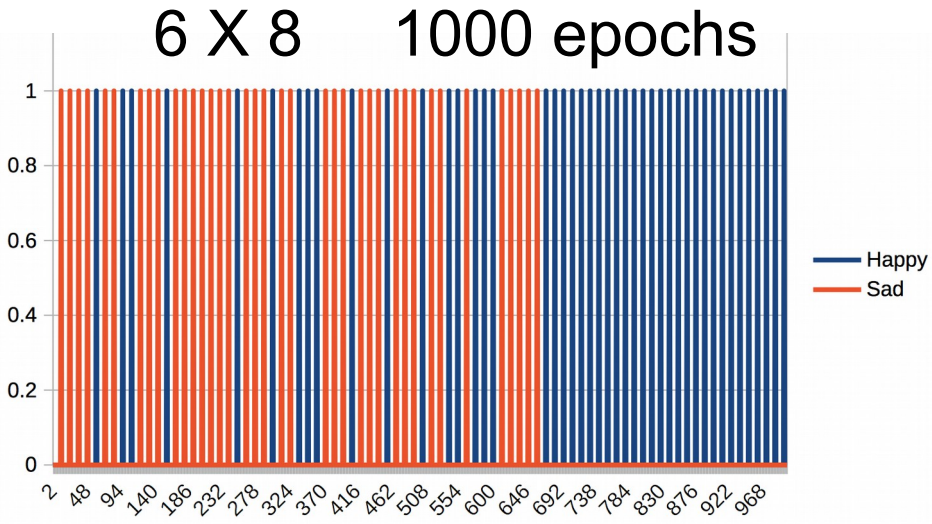
AGI Agent Cognitive Architecture
learning simplified “Atari Breakdown” game





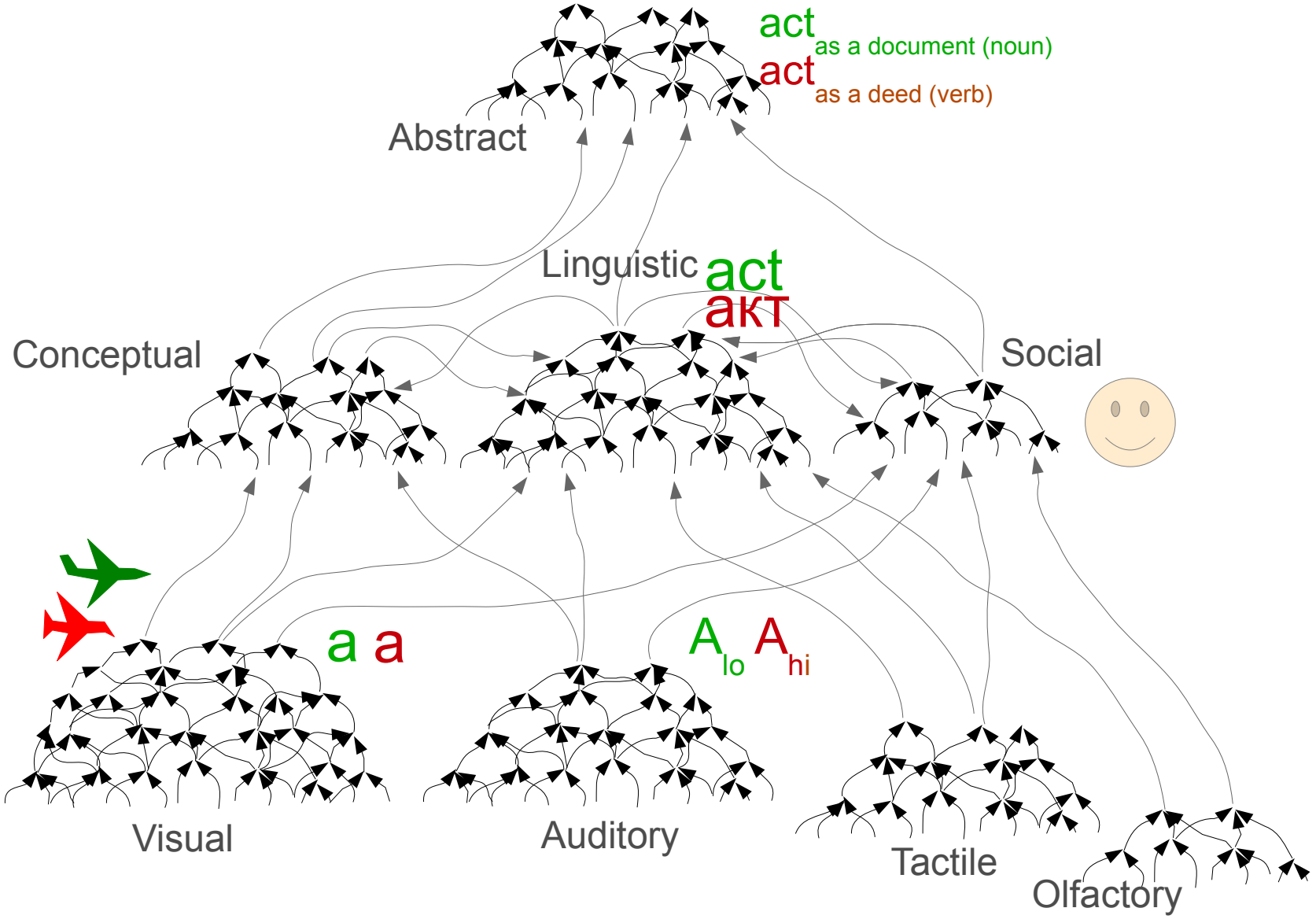
Xball(t)
Yball(t)
Xrocket(t)
Sad(t)
Happy(t)



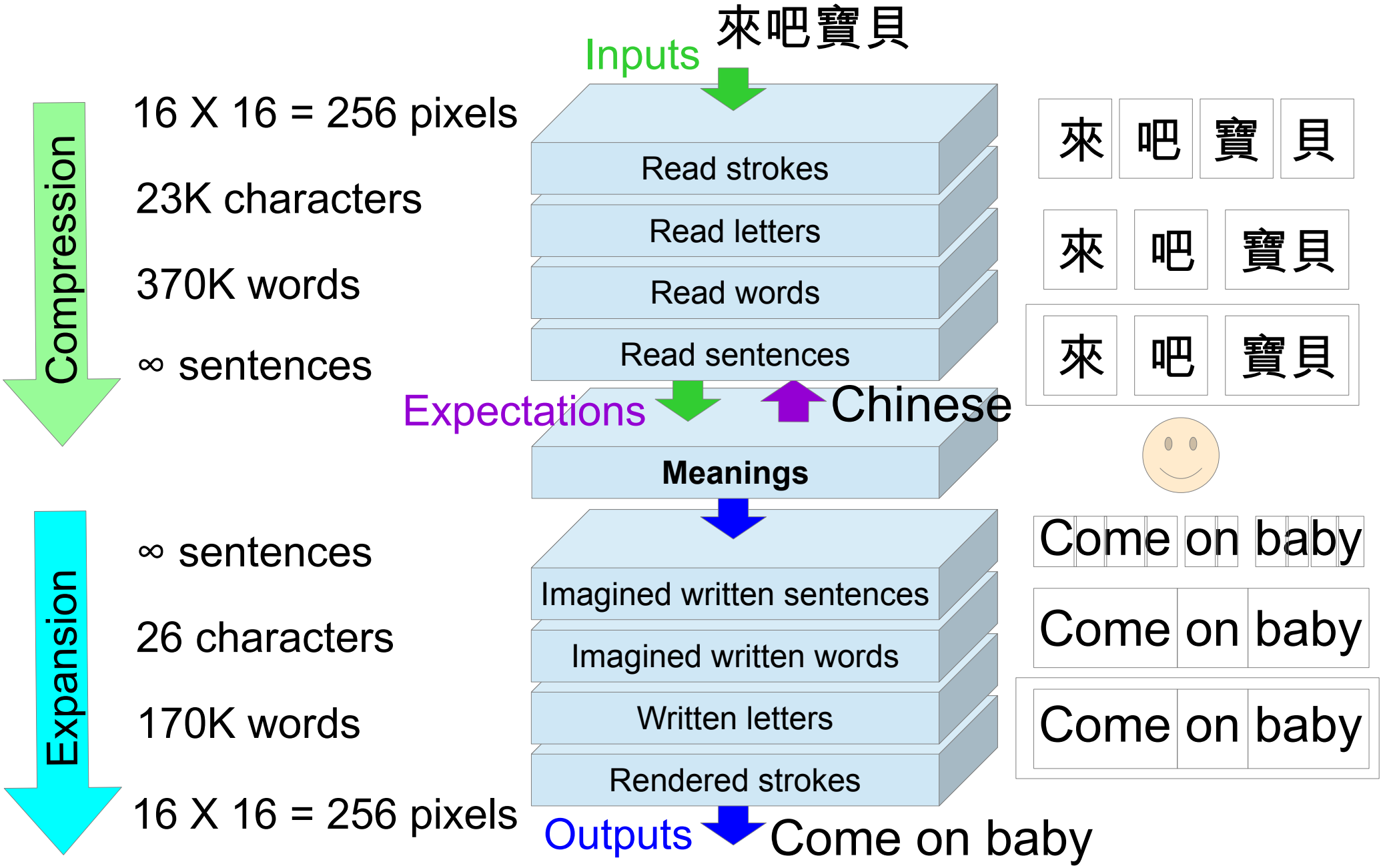


https://arxiv.org/abs/1807.02072
https://github.com/aigents/aigents-java/blob/master/src/main/java/net/webstructor/util/AgiTester.java

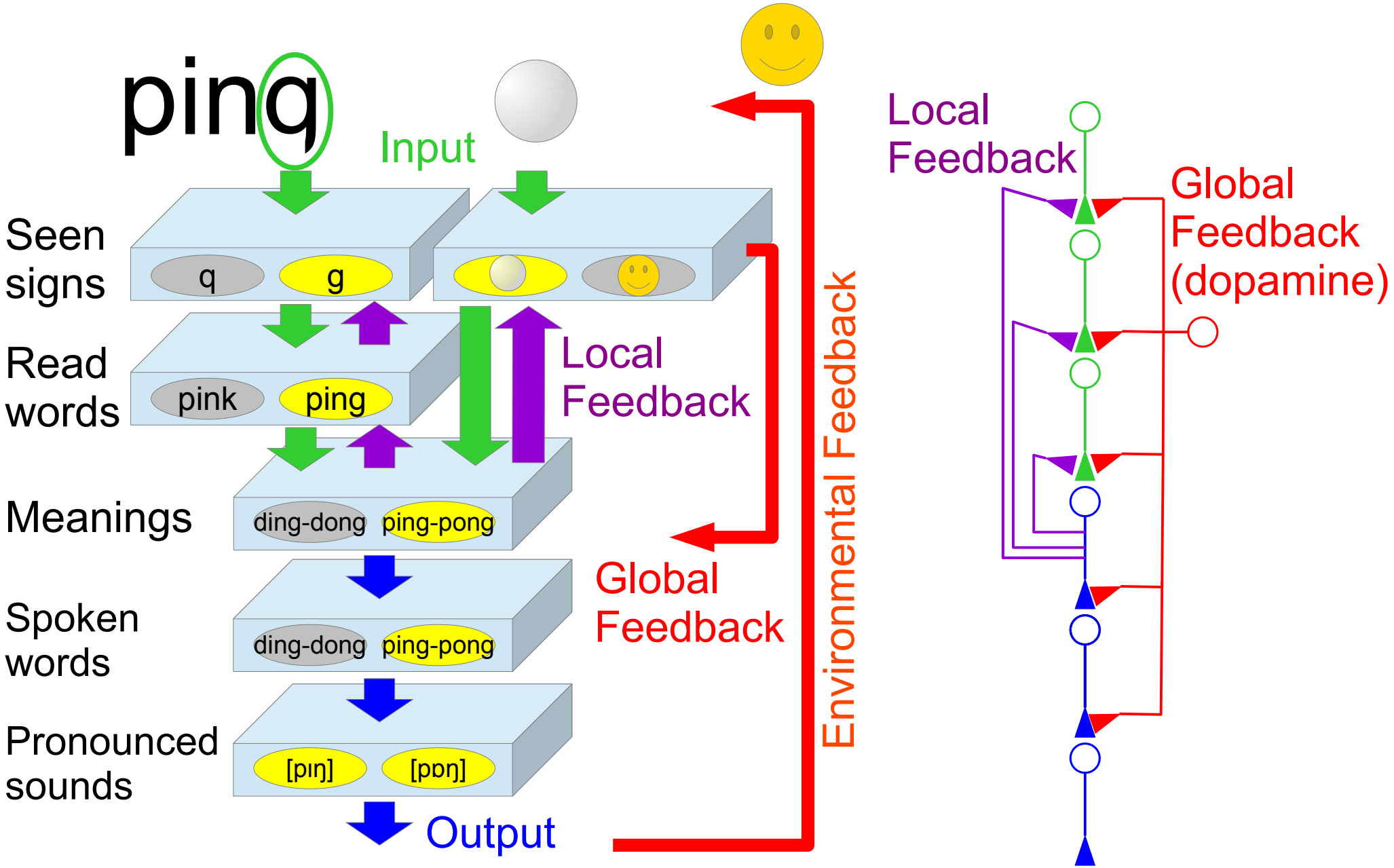
Physiological Plausibility – different segments of cortex responsible for different level of abstractions



Multi-layer transformation of complexity

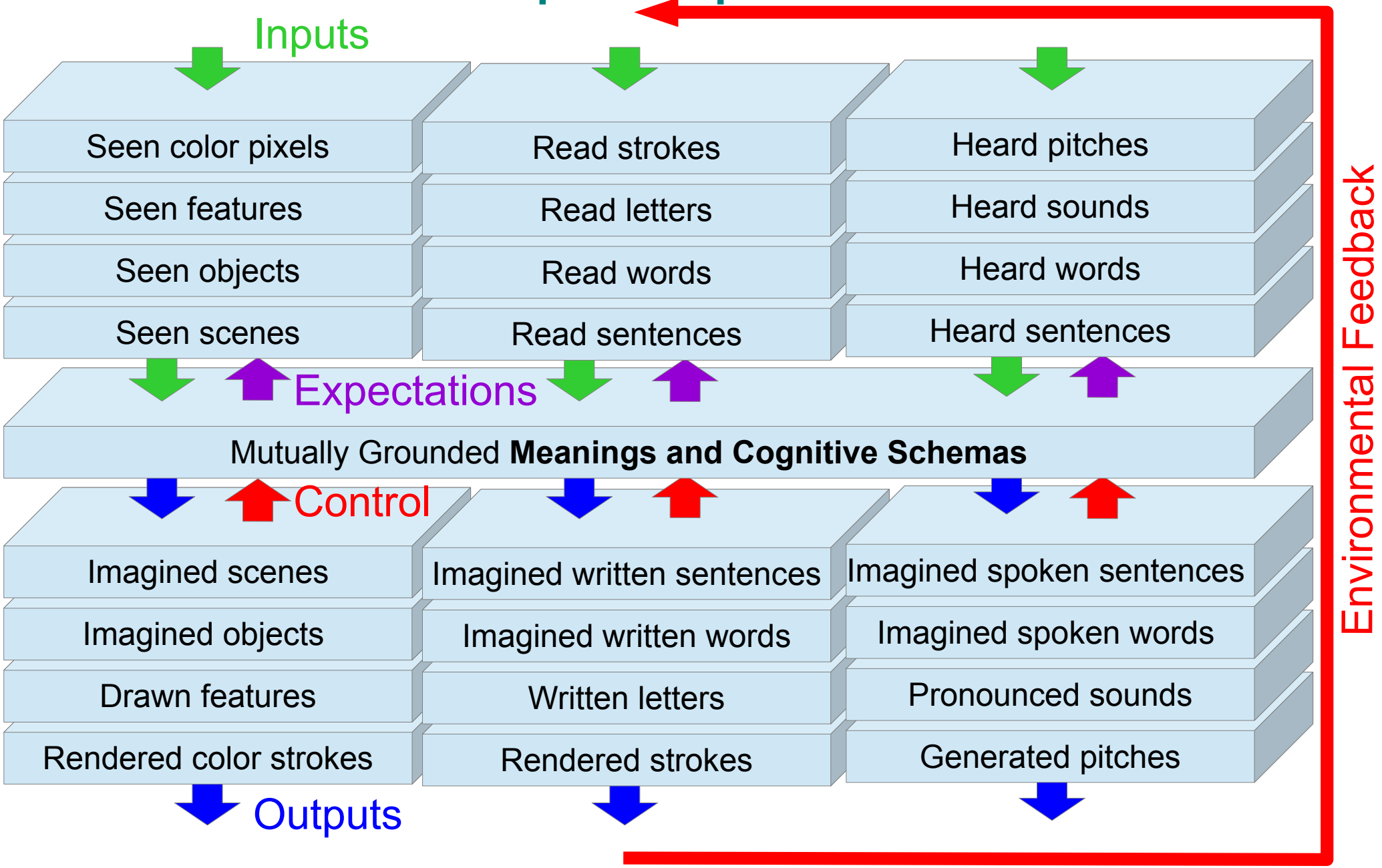


Local and Global Feedback



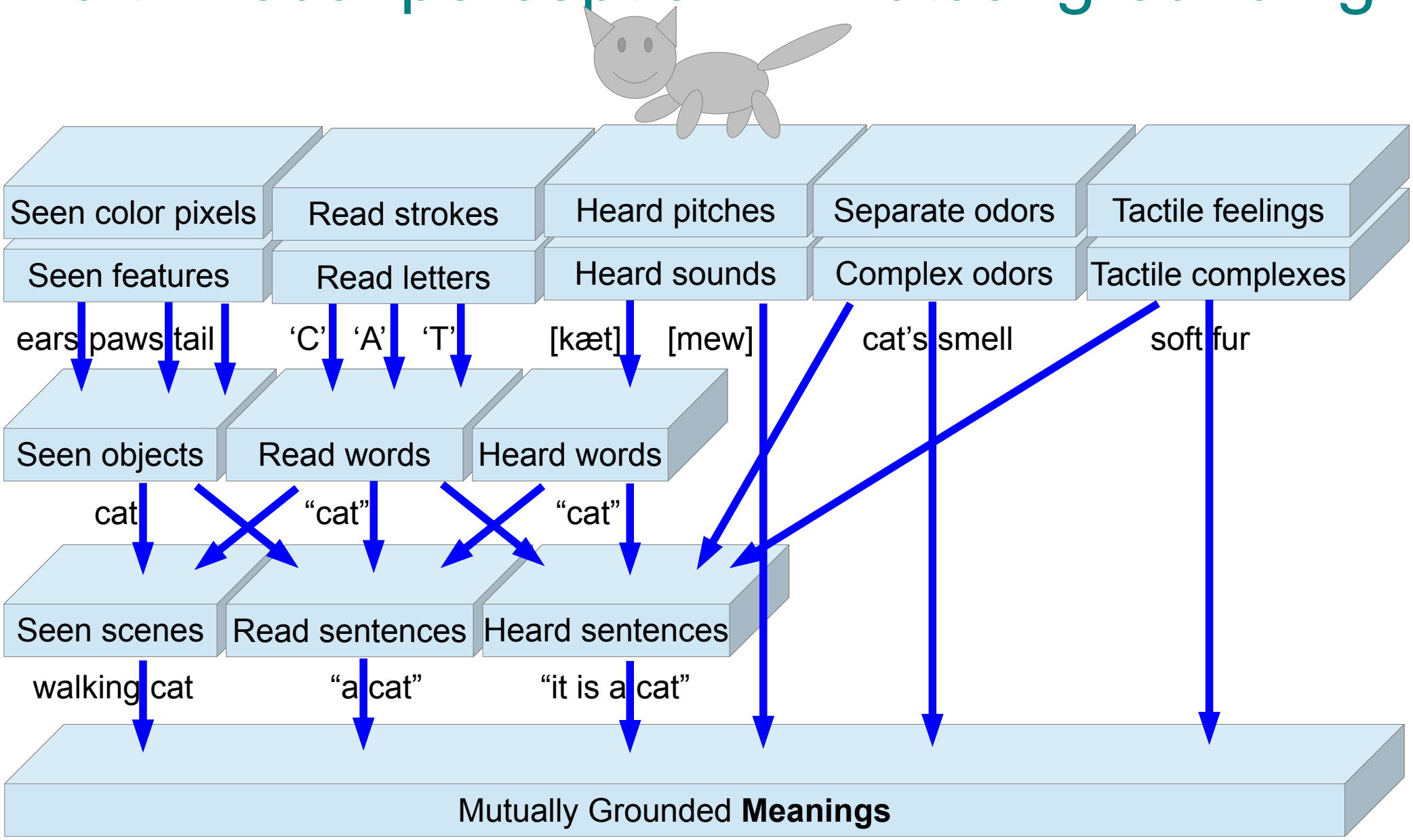
http://www.acad.bg/ebook/ml/Society%20of%20Mind.pdf (*"Global and Local Reward"*)

Multi-modal perception and action



https://www.abiresearch.com/blogs/2019/10/10/multimodal-learning-artificial-intelligence/

Multi-modal perception + mutual grounding

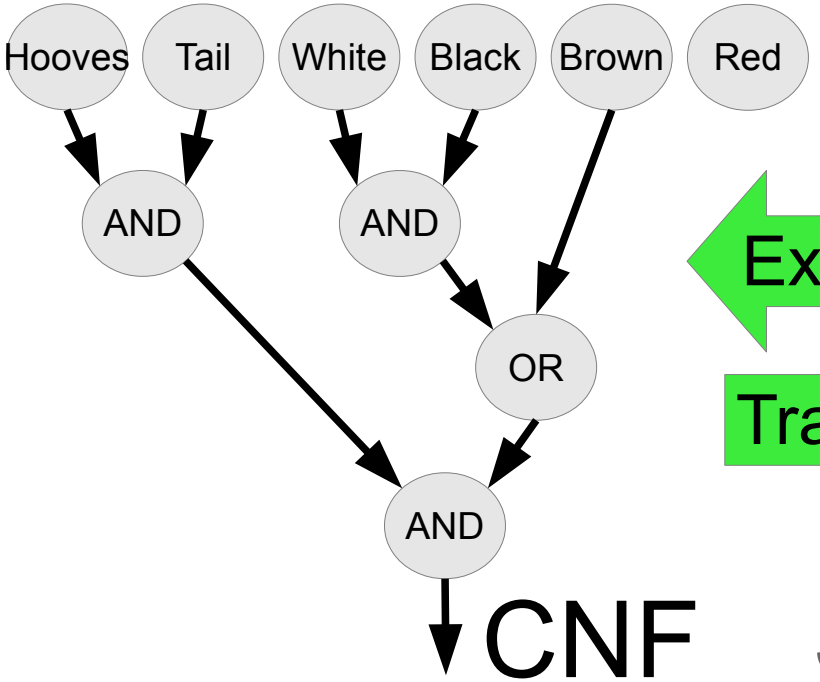


This walking cat is mewing!

https://www.abiresearch.com/blogs/2019/10/10/multimodal-learning-artificial-intelligence/

Bridging the Symbolic-Subsymbolic gap
for “explainable AI” and “transfer learning”

Labeled graph

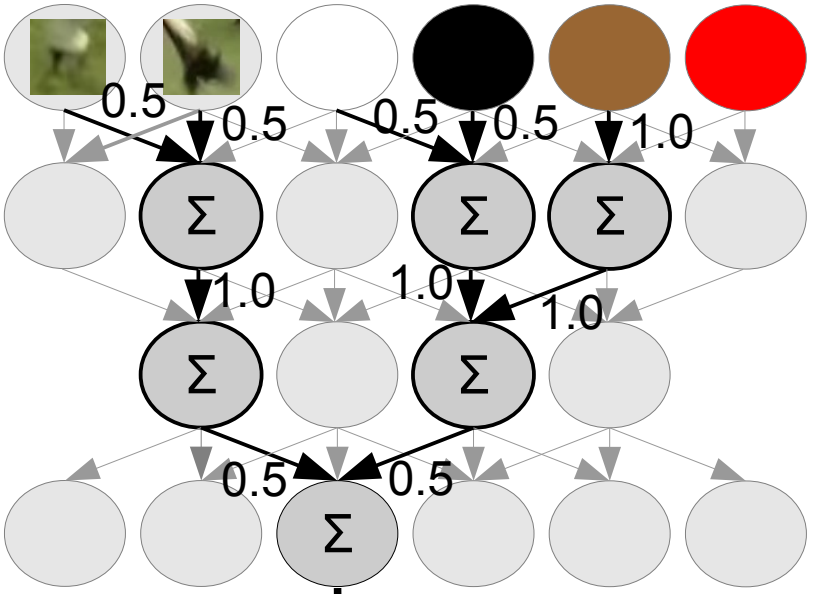


CNF

Explain

Transfer

Unlabeled graph



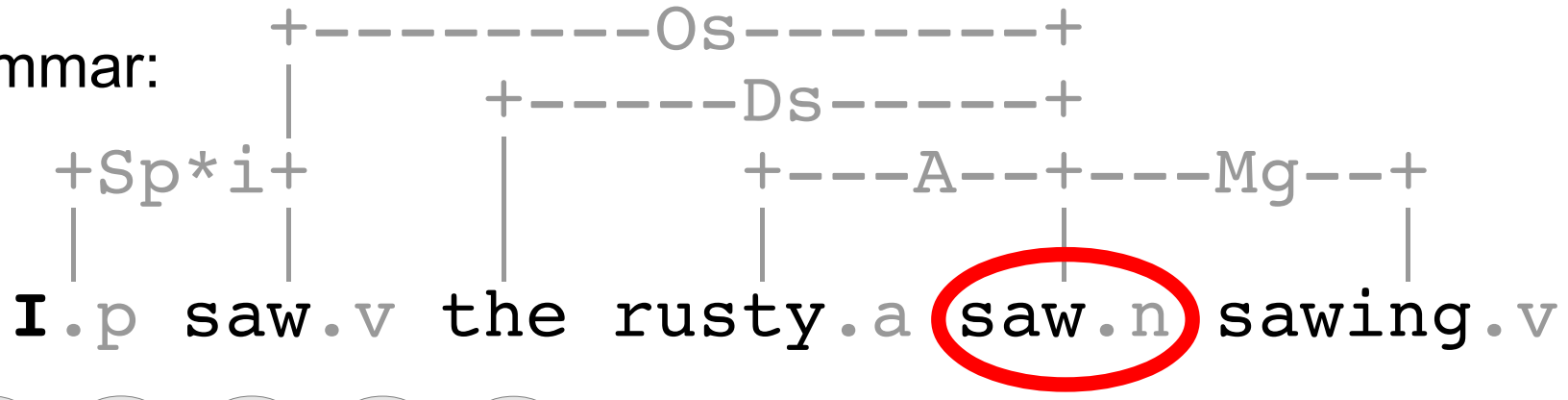
Soft-CNF

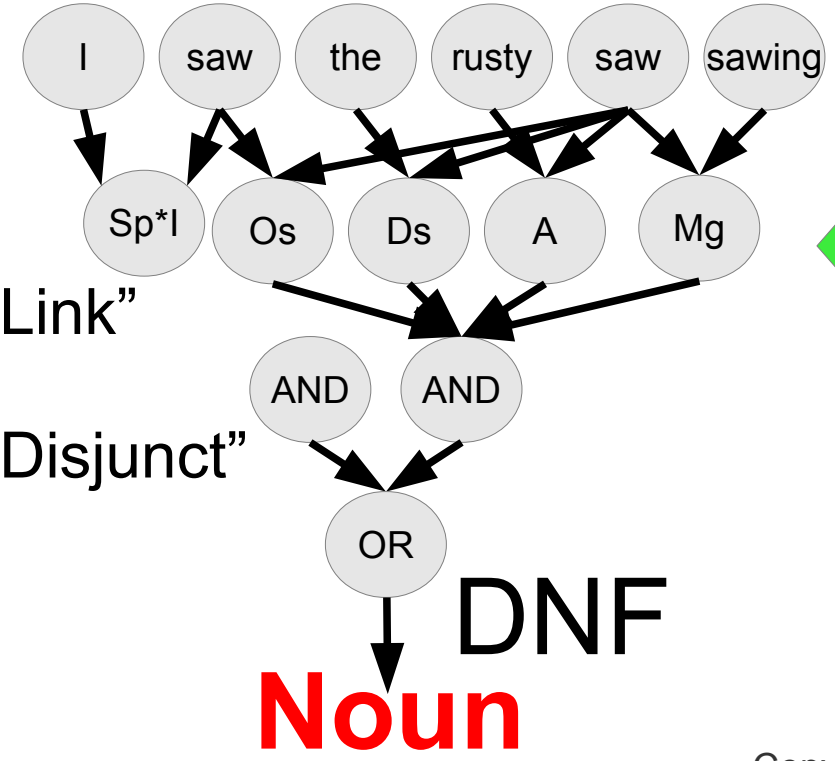
(Hooves AND Tail) AND
((White and Black) OR Brown)

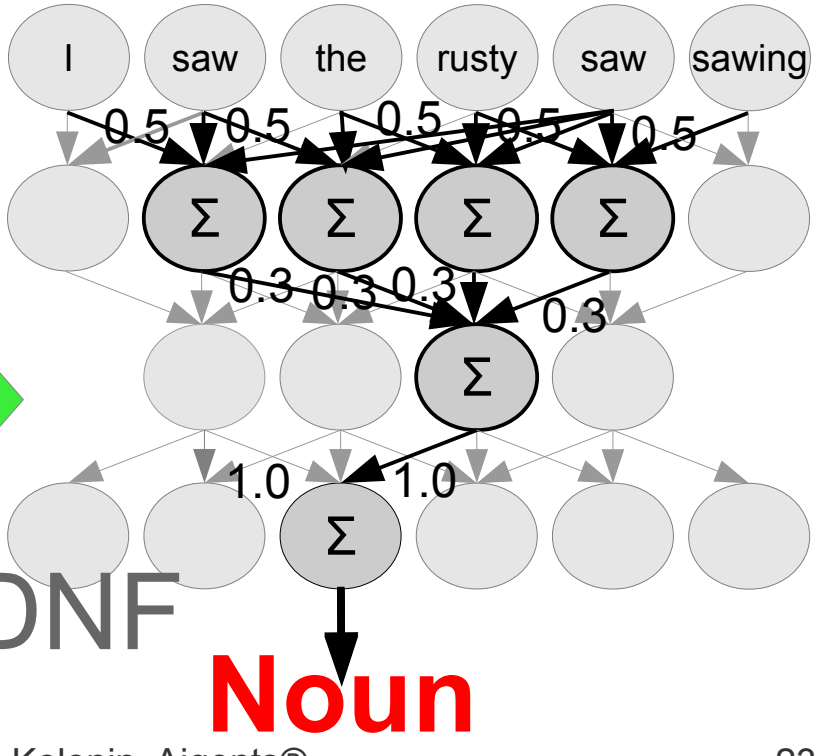
=> Horse

Bridging the Symbolic-Subsymbolic gap in NLP between distributed representations and formal grammars with ontologies

Link Grammar:







Explainable AI (XAI) vs. Interpretable AI

XAI model output explanation techniques

- "Dual" systems (like DNN + CBR)
- Transformation to "soft" CNF/DNF rules (softmax, arg max)
- XAI LIME
- XAI SHAP
- Life-long & Incremental Learning
  - (compression/forgetting–extraction–injection–extension)

Interpretable model representations

- Deep Bayes
- Deep Random Forest
- Soft Decision Trees
- Probabilistic/Fuzzy Logic
  - OpenCog/PLN
  - NARS
  - "Discovery"
- Formal Concept Analysis (FCA)

New (and forgotten old) approaches

Convergence

- Joining neural networks into cognitive architectures
  - Complex DNN Architectures
    - Attention-allocation networks
      - Contextual question answering and sentiment analysis
    - 3D CNN for spatiotemporal predictions
- Hybridization of neural networks with probabilistic programming
- Integration of neural networks with semantic networks (“knowledge graphs”)

AGI

- Narrow AGI (Ben Goertzel, SingularityNET)
- Probabilistic programming
  - It is in the state of “neural networks” 20 years ago and may turn revolutionary
    - in just few years (Alexey Potapov, SingularityNET)
- Agent Models
  - AIXI – universal compressor and predictor (Marcus Hutter, Arthur Franz)
  - Anokhin’s Theory of Functional Systems (TFS) extension (Evgenii Vityaev)
  - Reinforcement Learning based on "fluents" (state variables or atoms representing the states)
- Ecosystem approach
  - No one knows how to make AGI (so "let all colors bloom")
  - Brain on itself is not the same (different brain regions have totally different “architectures”)

New (and forgotten old) approaches

Deep Learning for AGI

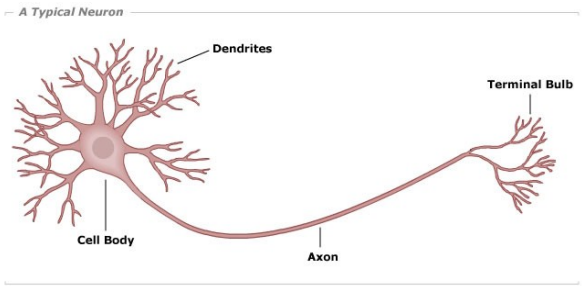
Demis Hassabis: "transfer learning is a key ... to general intelligence"
Yan Lecun: "learn models of the world"
Cognitive Pyramid:
- Casuality
  - Reinforcement-learning
    - Transfer-learning
  - Meta-learning (Learning to Learn)
    - DL
    - GANs
    - ML
    - ...

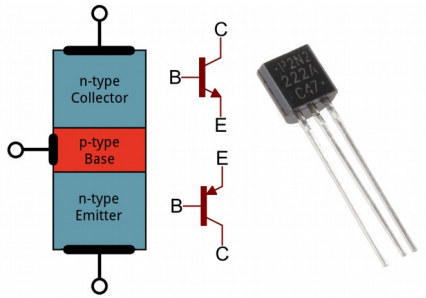
World<->ControlNetwork<->ModelNetwork *("Learn how to Learn")*

A lot to study, still

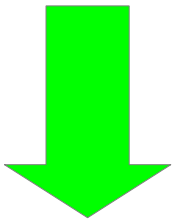
Dedrites/synapses branching
Holobionts - synergetically interacting societies
(matching studies by Jeff Hawkins,
who suggested the HTMs, preceding the DNNs)
Neuron Microtubules

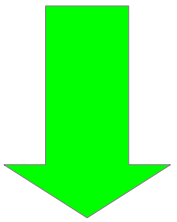
What is the Atom suitable to build AGI?

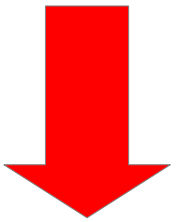


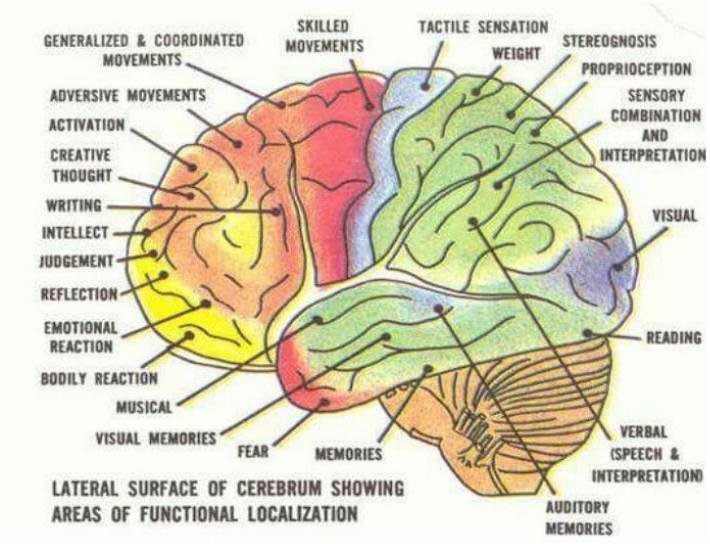


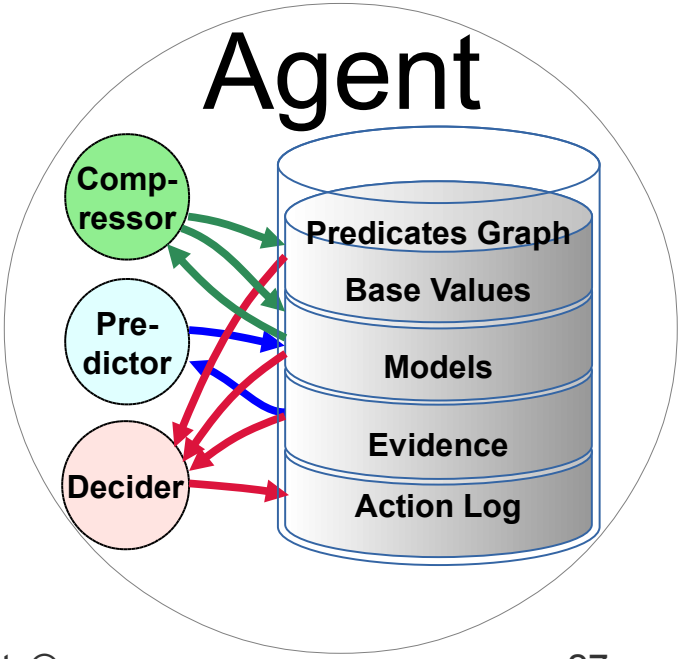
?



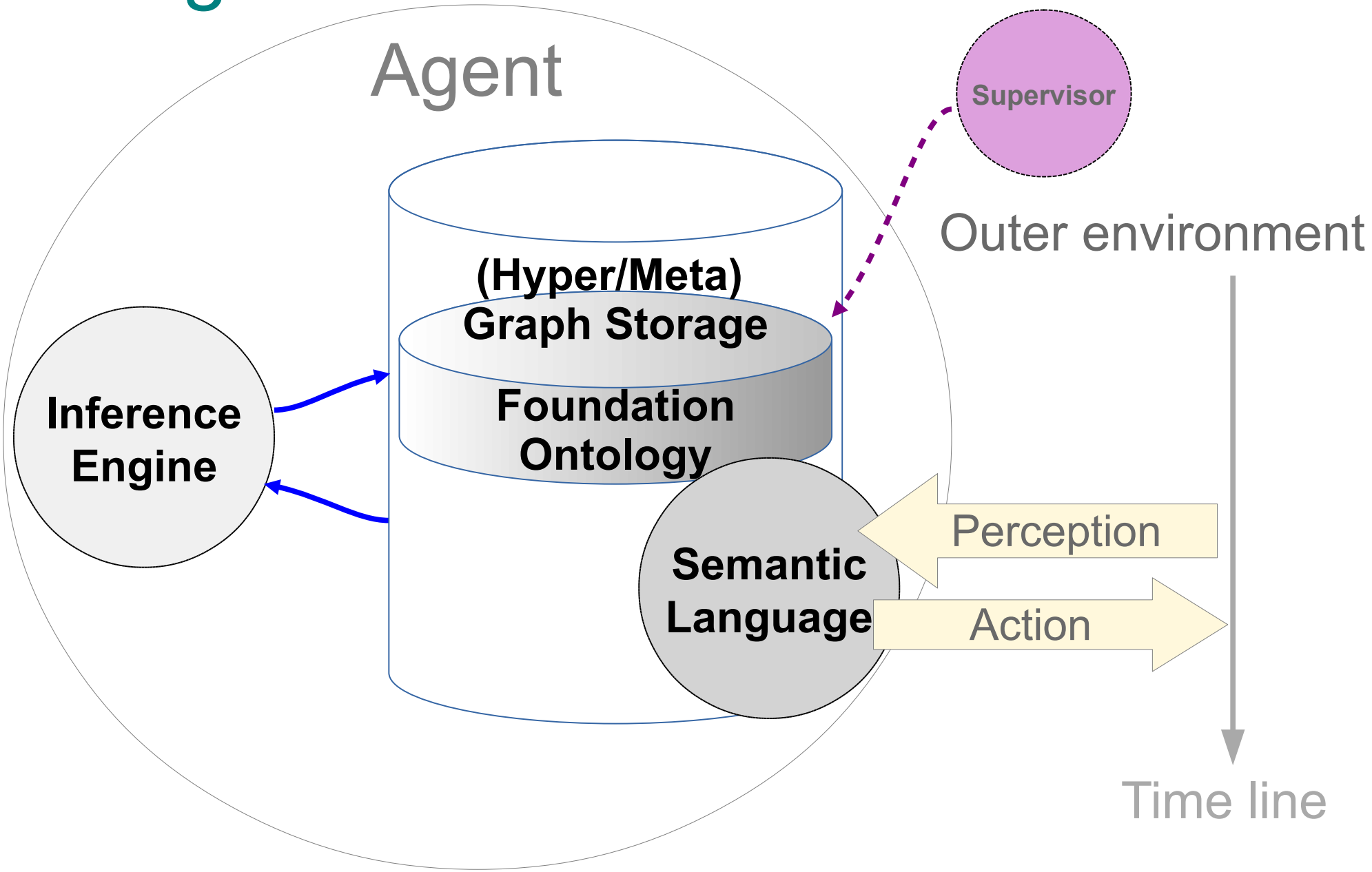








AGI Agent Cognitive Architecture
4 ingredients for “reasonable” AGI



Foundation Ontology

No-ontology
Cyc Knowledge Base
Freebase/Google Knowledge Graph
OpenCog Atomspace
Webstructor/Aigents®
Custom Domain

Graph Storage

Any RDBMS + time & triple indexing
Any "Triple Store" + time indexing
TimeScale DB + triple indexing
OpenCog AtomSpace
Aigents® Temporal Graphs

Semantic Language

RDF/OWL/Turtle/JSON-LD/SparQL
CycL – Cyc Language (D.Lenat)
ORL (A.Kolonin, L.Kuzin)
D0SL (V.Gumirov, D.Sviridenko)
OpenCog Atomese (B.Goertzel)
Narsese (P.Wang, P.Hammer)
AL – Aigents® Language (A.Kolonin)
Premise (M.Miller)

Inference Engine

Non-Axiomatic Reasoning System – NARS (P.Wang, et. al.)
Logical Prediction System – "Discovery" (E.Vityaev et.al.)
OpenCog Probabilistic Logic Network – PLN (B. Goertzel et. al.)

What is all you need?

	Inference Engine	**Semantic Language**	**Graph Database**	**Foundation Ontology**	**License**
OpenCog	Unified Rule Engine (URE)/ Probabilistic Logic Network (PLN)	Atomese	AtomSpace	AtomSpace Atom Types	GPL
Non-Axiomatic Reasoning System (NARS)	Non-Axiomatic Reasoning System (NARS)	Narsese	Internal In-memory	Ontology-agnostic?	MIT
Discovery + D0SL	Discovery	D0SL	Internal in-memory	D0SL	GPL + Commercial
Webstructor/ Aigents®	*TBD*	ORL, AL	Aigents Graphs	Aigents	MIT

Graph Storage for “Temporal Predicates”

Verb(Subject,Object)<Time>

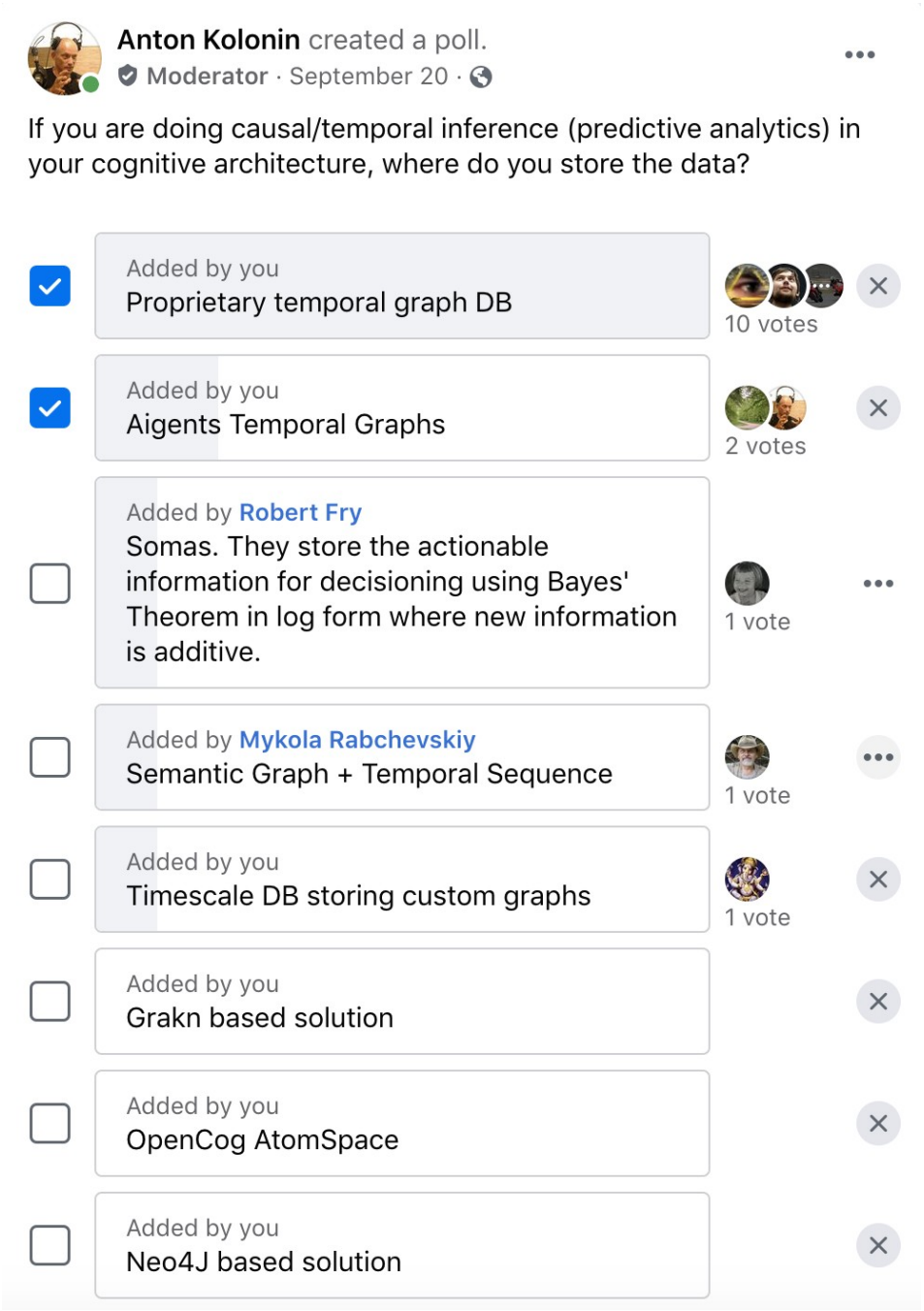
RDBMS:
[Verb,Subject,Object,Time]
+ custom indexes on triple & time

Triple Stores (Neo4J, Grakn, etc.):
[Verb,Subject,Object](Time)
+ custom indexes on time

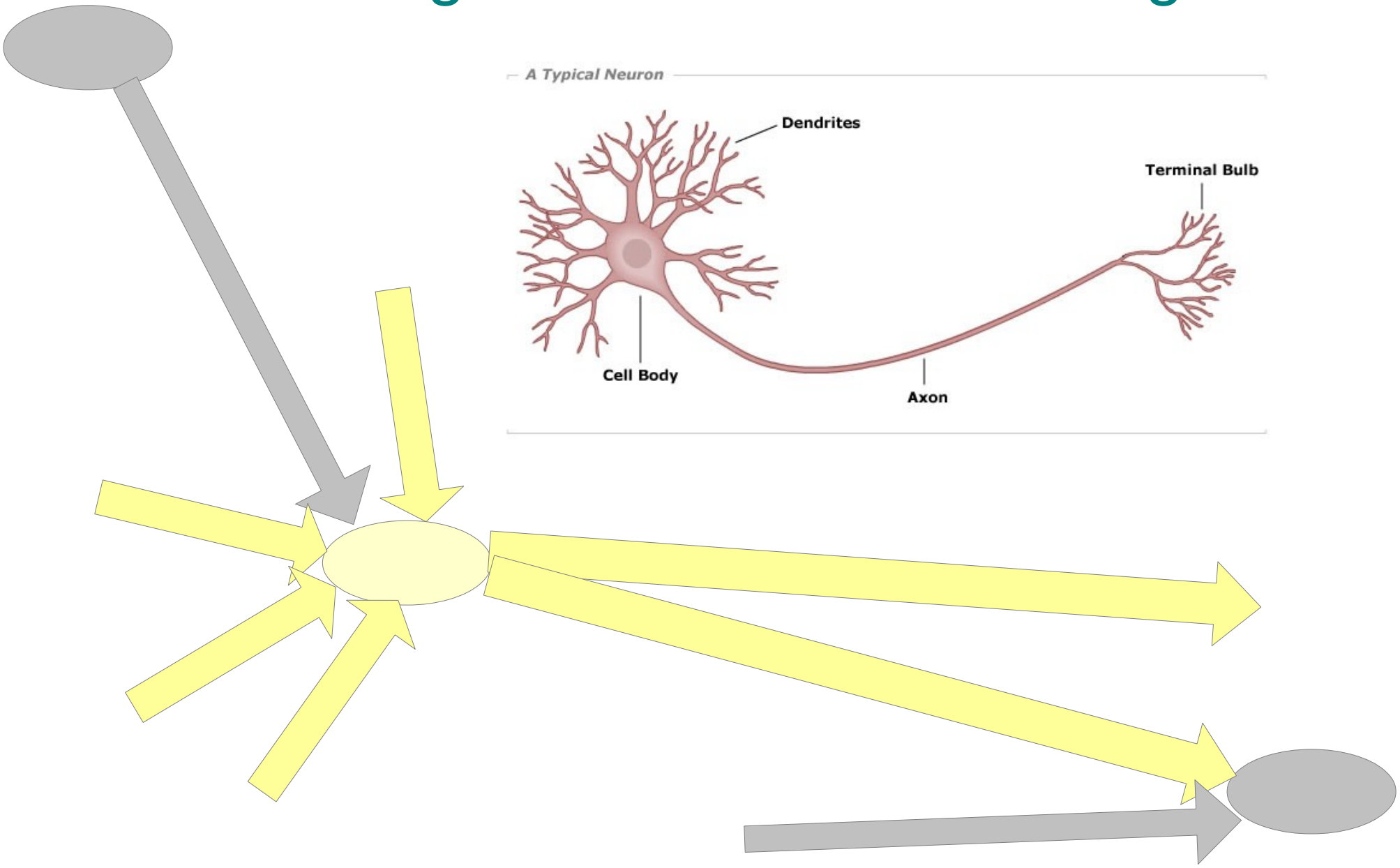
TimeScale DB:
[Verb,Subject,Object,Time]
+ custom indexes on triple

OpenCog AtomSpace:
custom representation & indexes

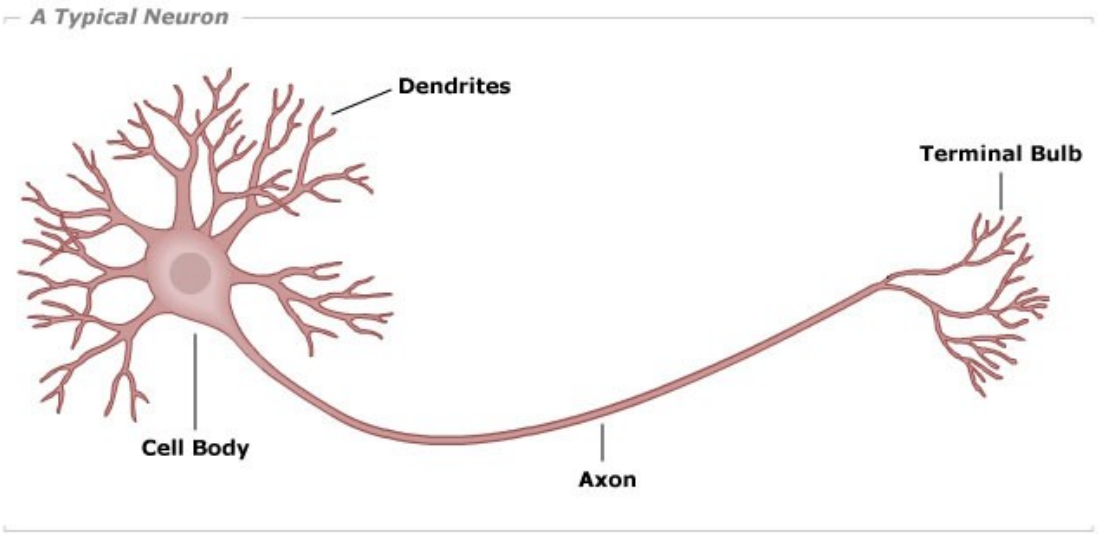
Aigents® Temporal Graphs:
Time([Verb,Subject,Object])



Does Artificial Neuron Looks Like the Real One? Processing on Dendrites Branchings?

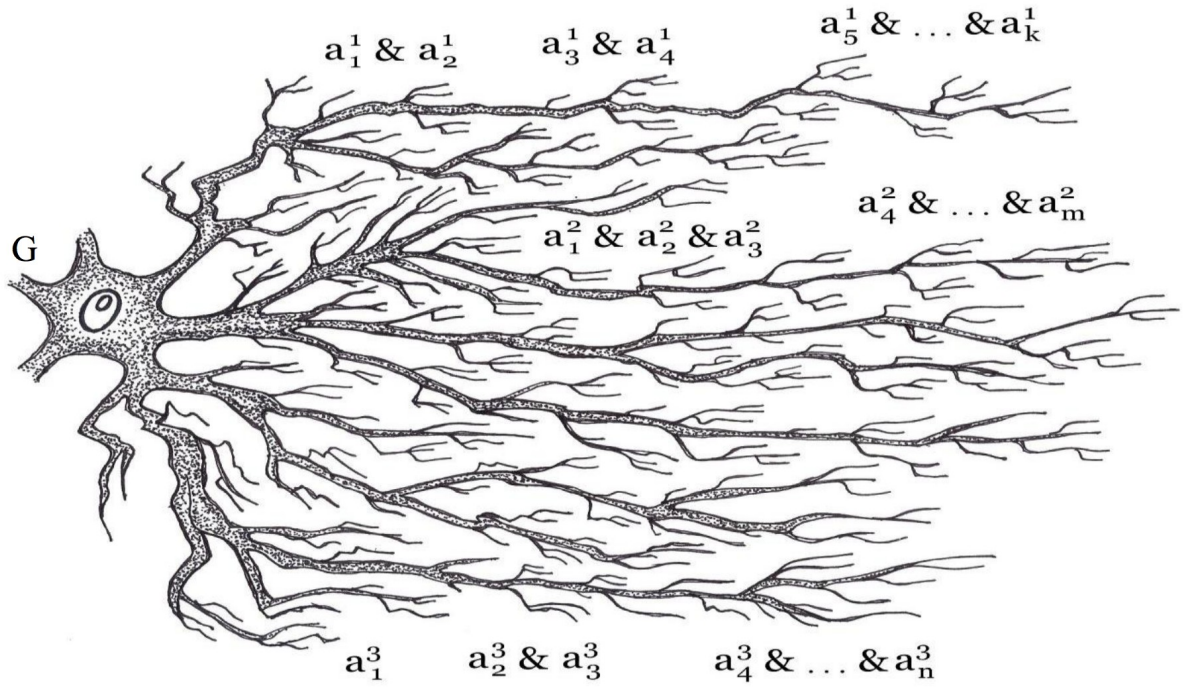


Does Artificial Neuron Looks Like the Real One? Processing on Dendrites Branchings?



Treating single neuron as hierarchy of predicates

Semantic probabilistic inference as a formal model of neuron



If G denotes a stimulus, on which the neuron responds unconditionally, then the stimuli arriving at the dendrites can establish conditional connections with excitation of the neuron G.

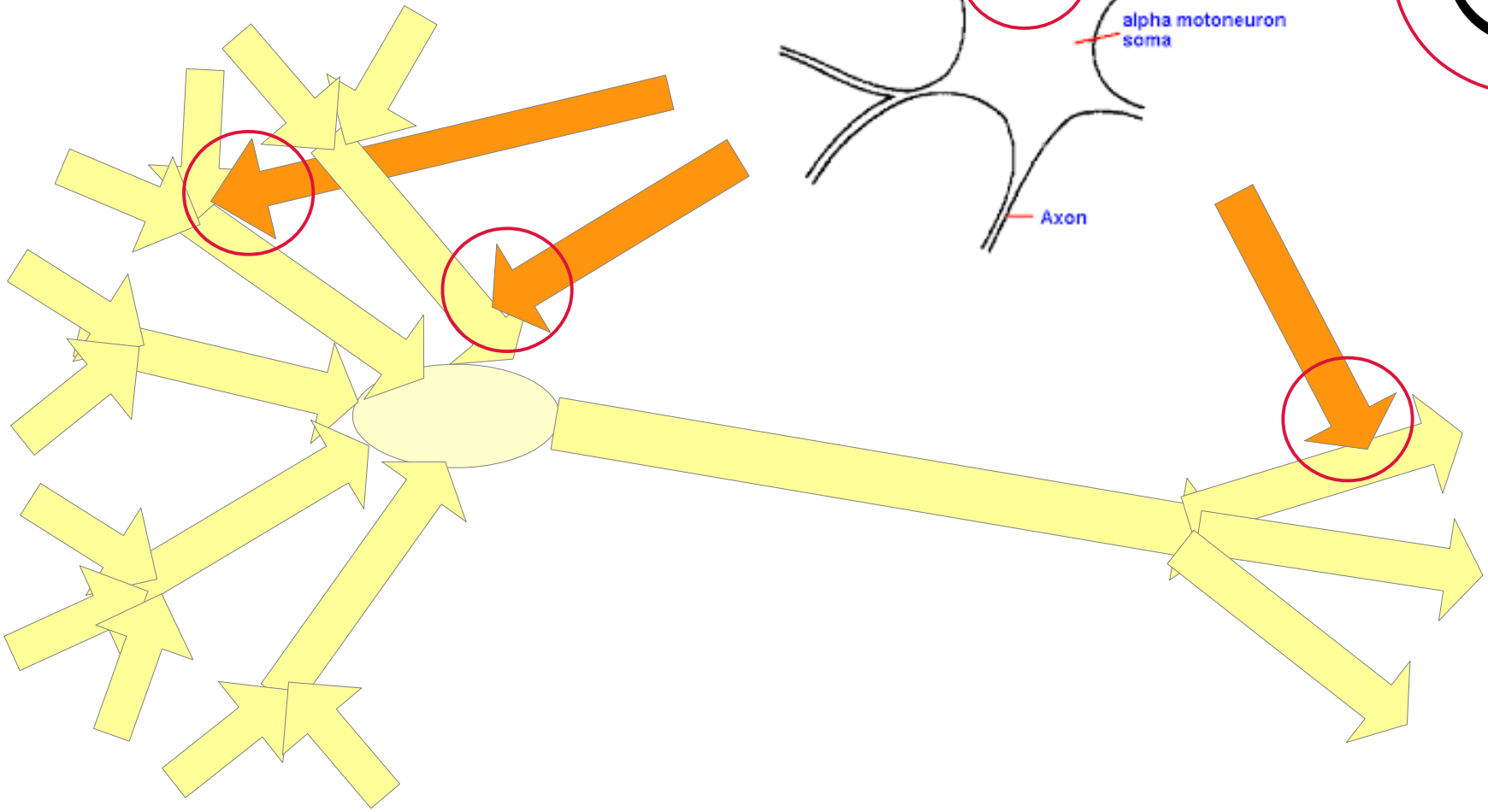
New stimulus are added to the rule only if they increase its conditional probability. This is a manifestation of the conditional connections closure at the level of a neuron, its plasticity.

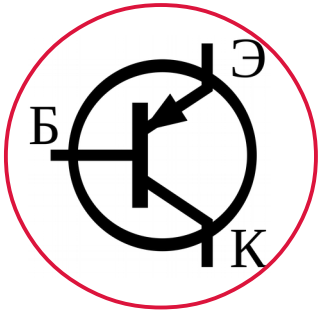
Neurons respond faster to the most probable conditional connections and hence first of all on Maximum Specific Rules.

Vityaev E.E. A formal model of neuron that provides consistent predictions // Biologically Inspired Cognitive Architectures 2012. Proceedings of the Third Annual Meeting of the BICA Society. In Advances in Intelligent Systems and Computing, v.196, Springer. 2013, pp. 339-344.

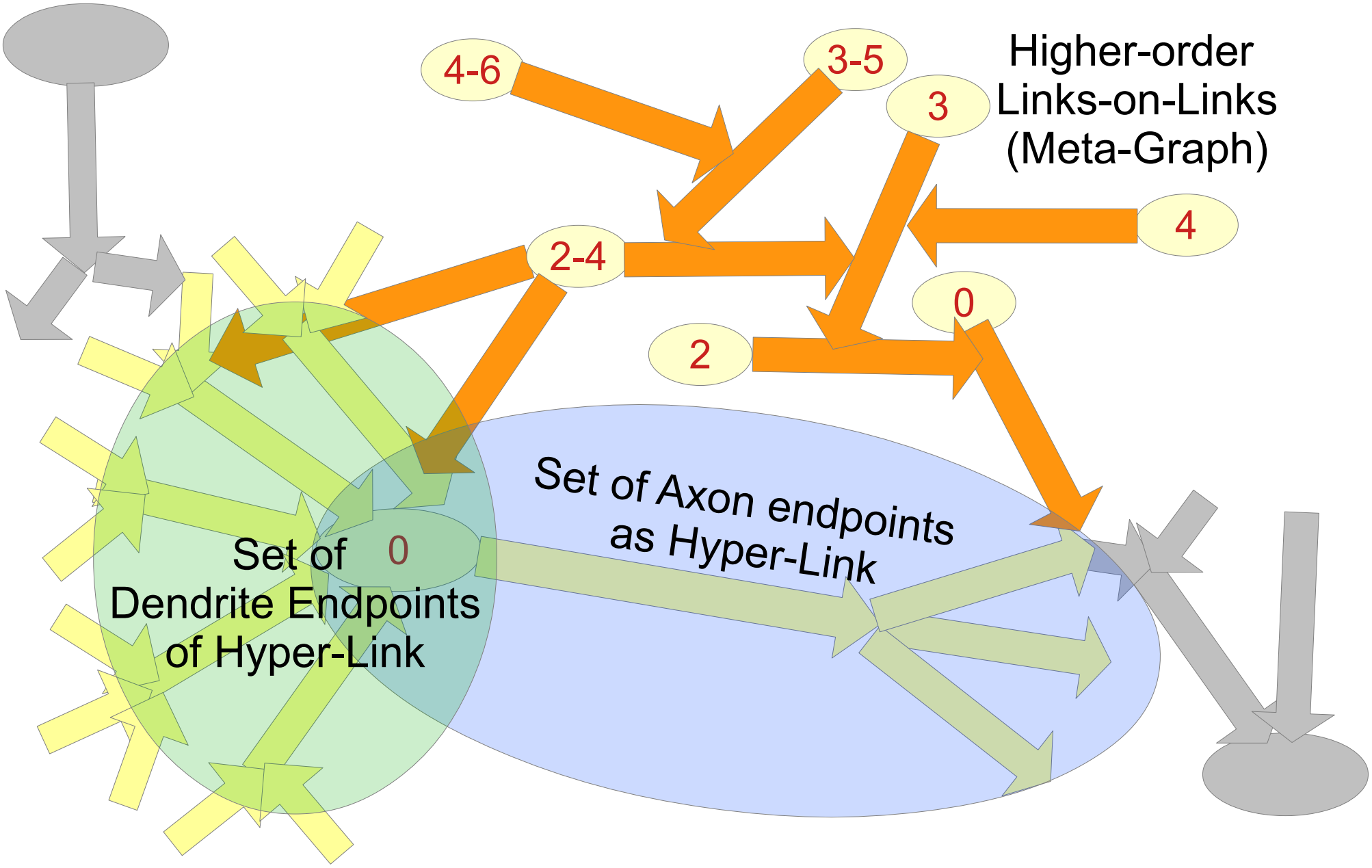
https://www.youtube.com/watch?v=fQ-bK17Y0Ak

Does Artificial Neuron Looks Like the Real One? Synapses-on-Synapes?

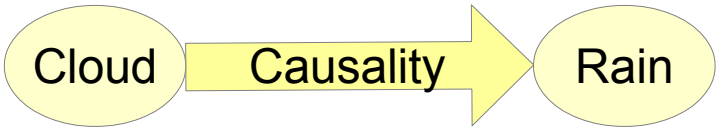




True Connectome with Hyper-Meta Graphs

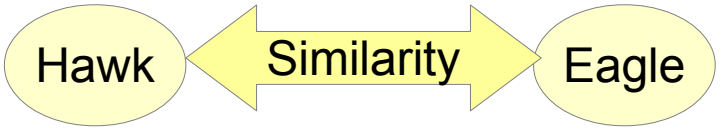


Ordered and Unordered Binary Links/Predicates and Hyper-Links

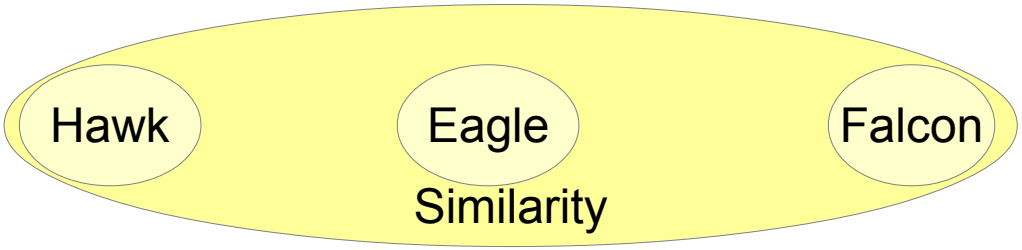


Ordered (directed) Link, arity = 2
Cloud is “source”
Rain is “target”
Real Link

Ordered (directed) Link, arity = 3
Cloud is “source”
Rain is “source” and “target”
Flood is “target”
Sequence or Series

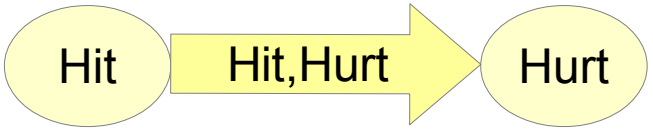


Unordered (undirected) Link, arity = 2
Just Pair

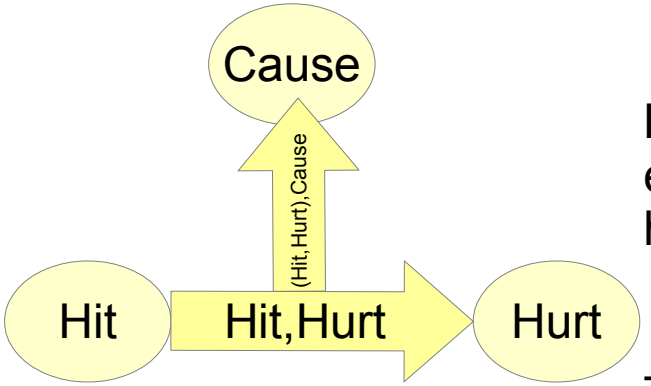


Unordered (undirected) Link, arity = 3
Set

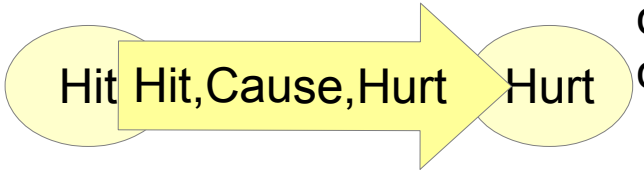
Typed/Labeled Links/Predicates vs. Arity



Binary typeless link (from, to) with arity = 2



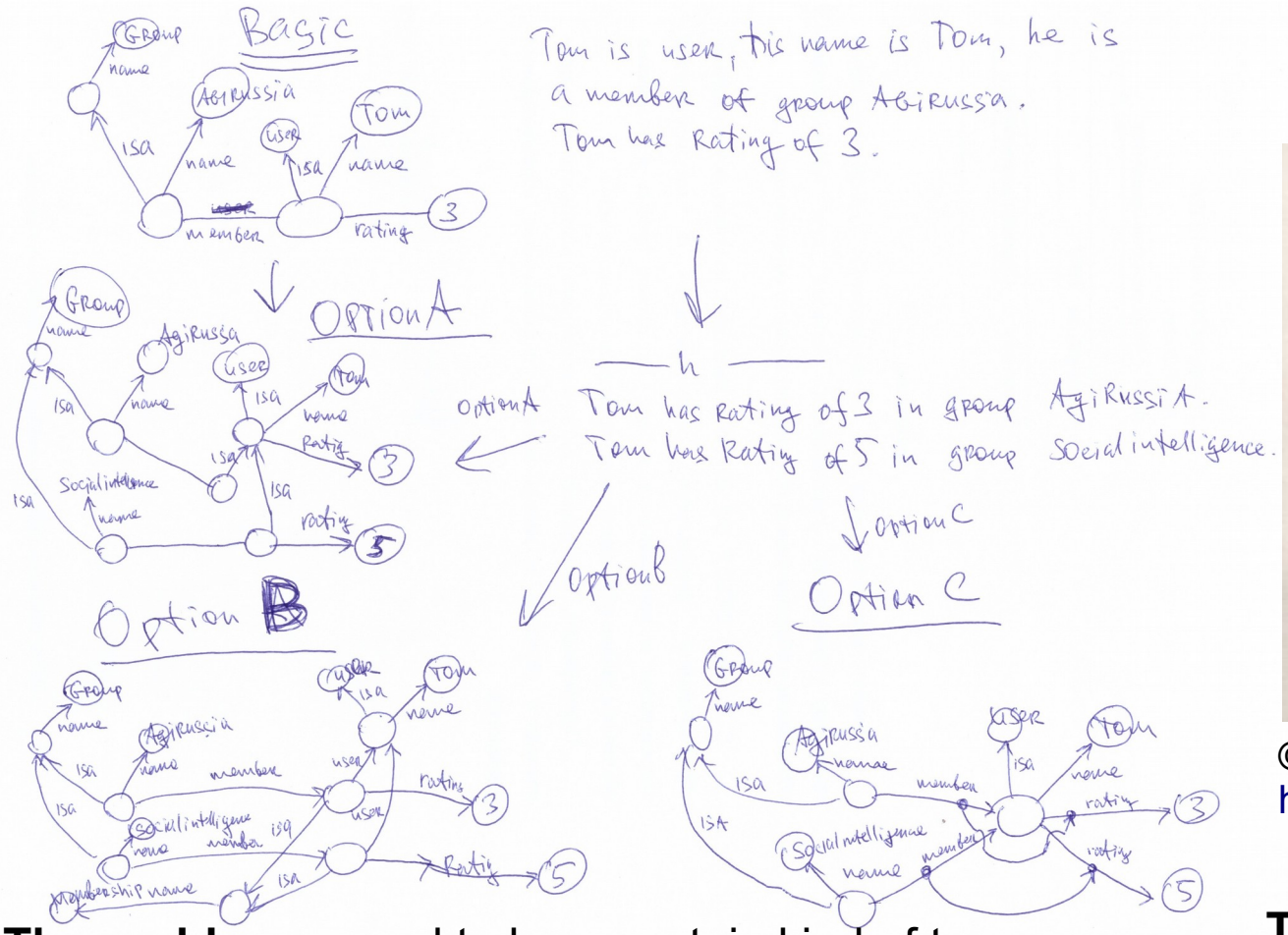
Binary link (from, to) with arity = 2 where type is expressed with extra binary link to link type node (needs higher-order links-on-links capability)

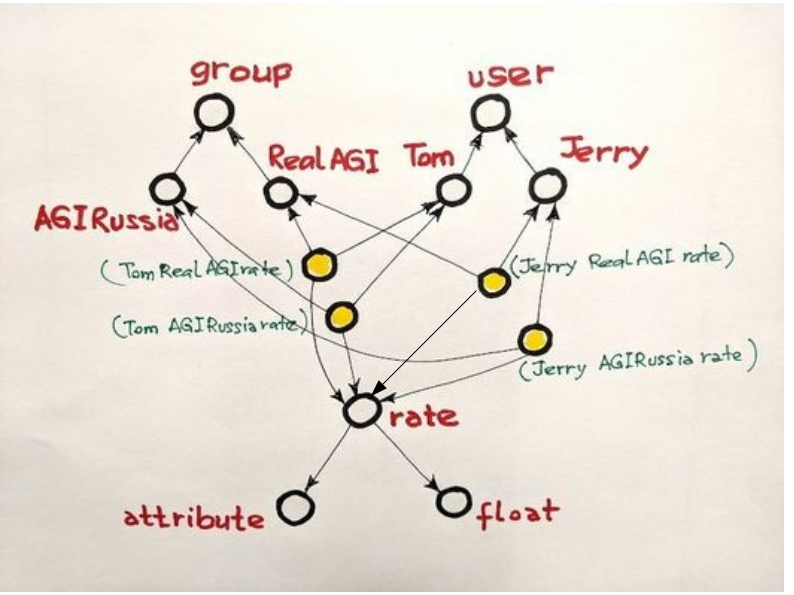


Ternary triple (from, type, to) with arity = 3 where type is part of relation, corresponds to “term” subject-predicate-object in natural language processing and “triple” in most of conventional graph databases (RDF triplet stores)

Separate predicate (type) and list of its arguments (from, to) with arity = 2 corresponds to “term” in predicate logic and RDF statement in modern semantic web programming and implemented in OpenCog and Webstructor, dealing with arbitrary arities, specific to those predicate types

Typed predicates and design ambiguity vs. No-ontology with binary predicates only



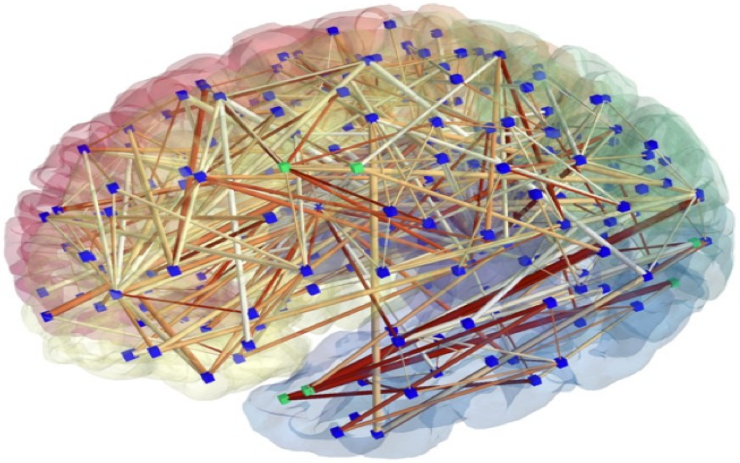


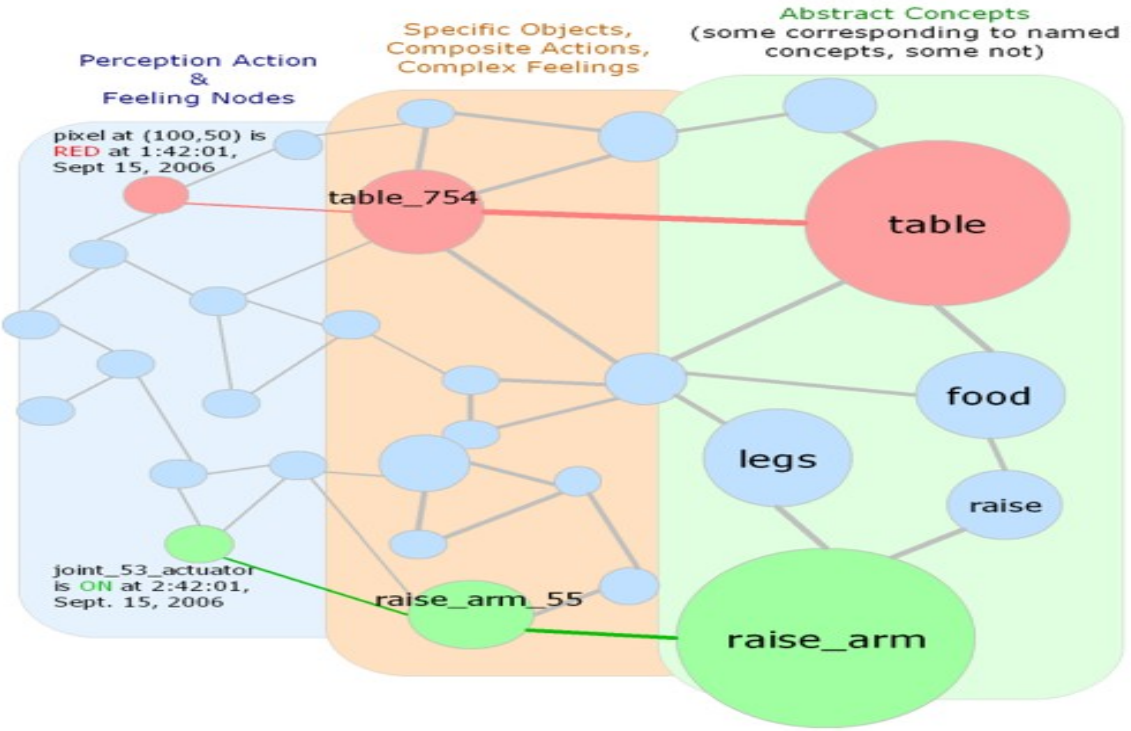
The problem – need to have certain kind of ternary relationships (typed links) to be specific to context, eg. "rating" property if a user in different groups may have different value. Redesign of the domain ontology to account for this has multiple ambiguous options involving need for meta-links (option C).

The solution – don't have a "domain ontology" at all, let all links "binary" with implicit "is-a" type. Treat every meaningful relationship as an "unlabeled" node (yellow nodes above).

OpenCog's "AtomSpace"

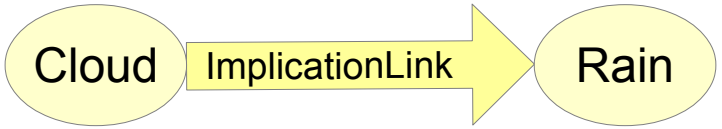
Implements Generalized Hyper-graph and Meta-graph, so each directed/undirected link may link together any number of atoms, where atom could be either node (arity = 0) or any other link with any arity, including unordered N-ary links representing subgraphs as their elements.



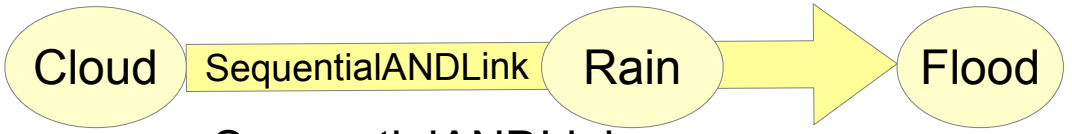


https://github.com/opencog/atomspace
https://github.com/opencog/atomspace/blob/master/opencog/atoms/atom_types/atom_types.script

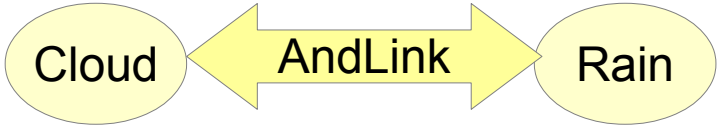
Ordered (directed) and Unordered (undirected) Links



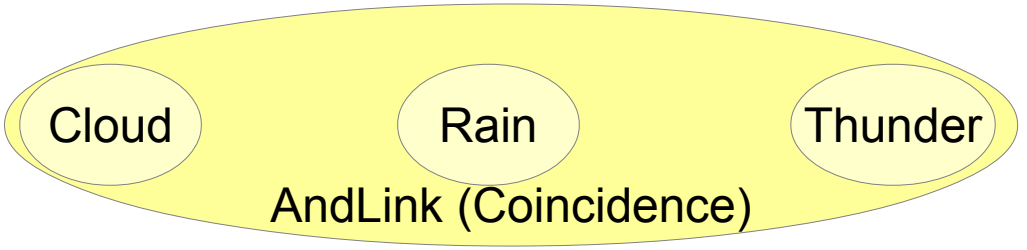
ImplicationLink
  ConceptNode ”Cloud”
  ConceptNode ”Rain”



SequentialANDLink
  ConceptNode ”Cloud”
  ConceptNode ”Rain”
  ConceptNode ”Flood”



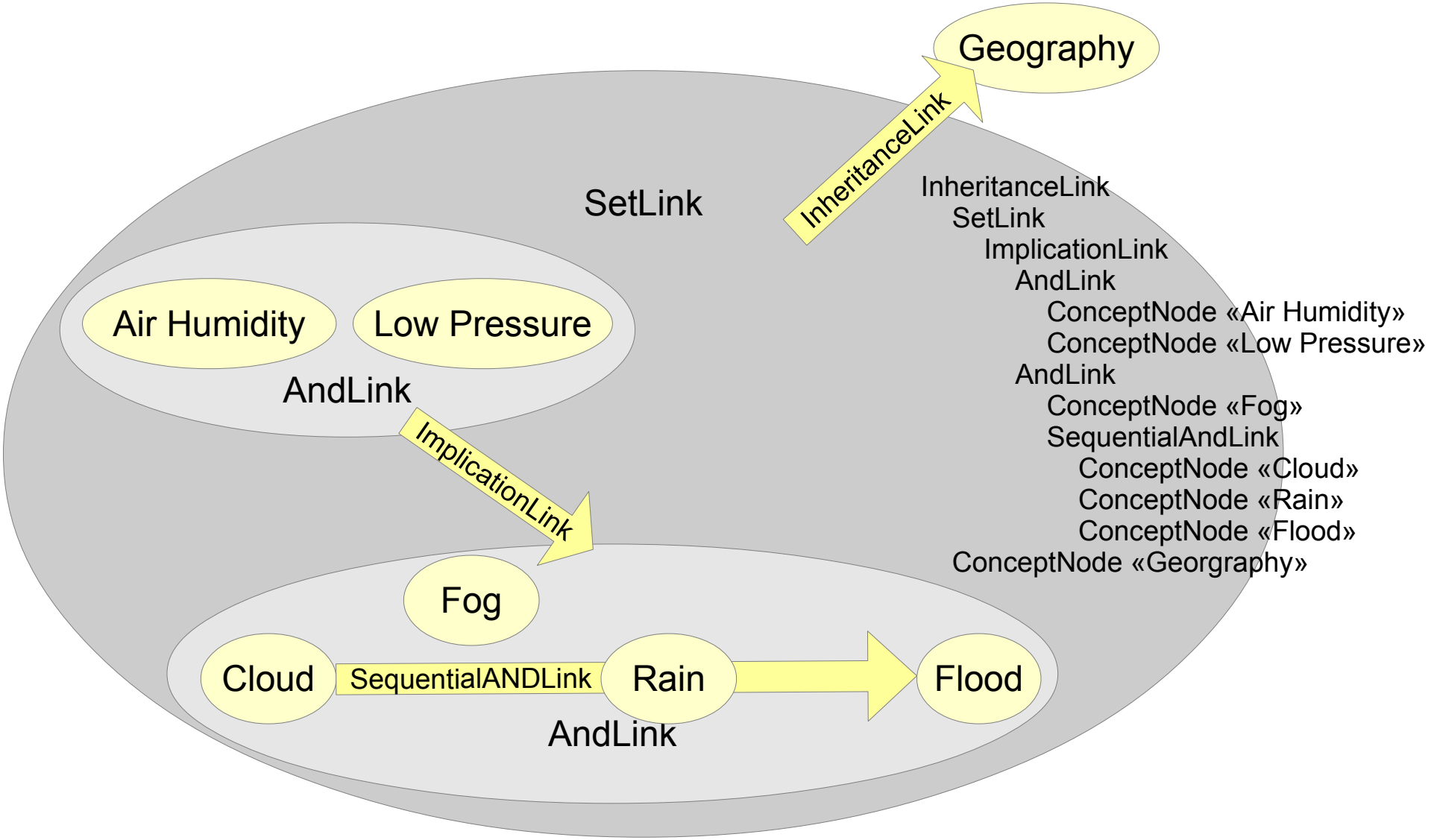
AndLink
  ConceptNode ”Cloud”
  ConceptNode ”Rain”



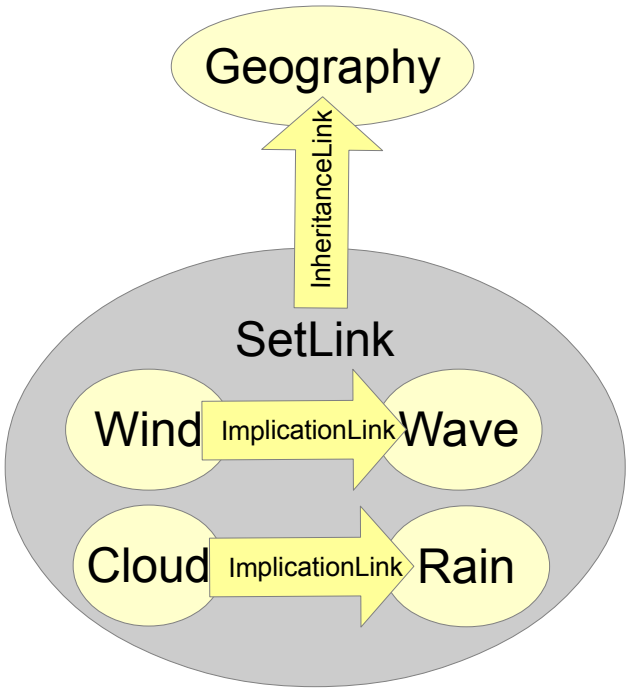
AndLink
  ConceptNode ”Cloud”
  ConceptNode ”Rain”
  ConceptNode ”Thunder”

expressed in Atomese language

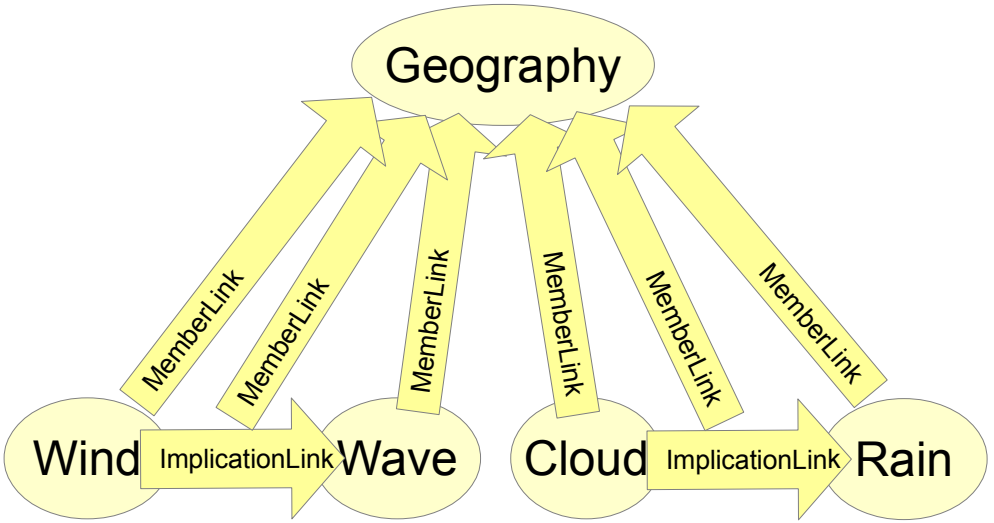
AtomSpace - Generalized (Link-as-Node) Hyper-Graph is a Meta-Graph (in Atomese)



AtomSpace - Meta-Graphs with Hyper-Graphs and Links-on-Links



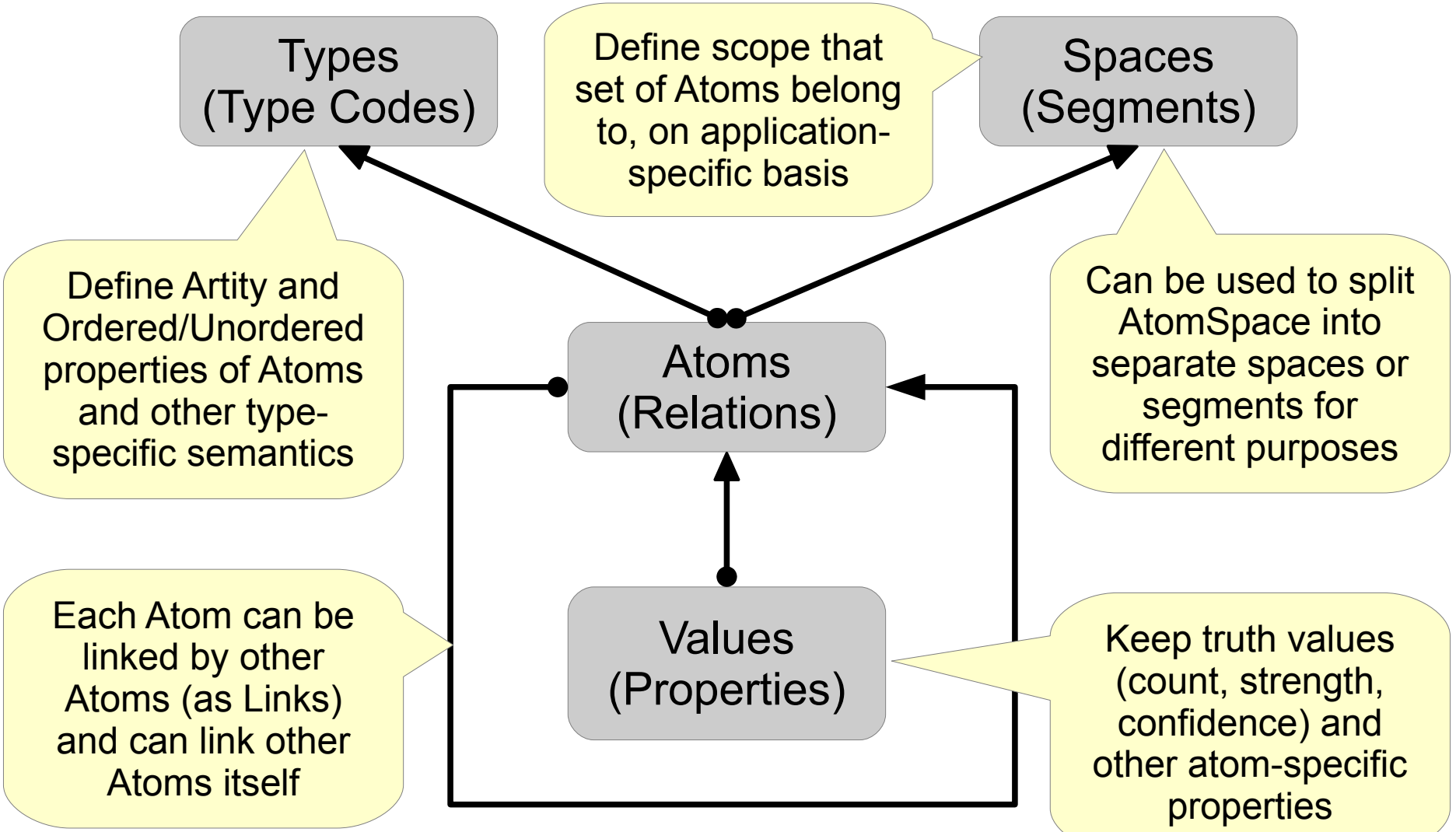

```
InheritanceLink
    SetLink
        ImplicationLink
            ConceptNode «Wind»
            ConceptNode «Wave»
        ImplicationLink
            ConceptNode «Cloud»
            ConceptNode «Rain»
     ConceptNode «Georgraphy»
```




```
MemberLink
    ConceptNode «Wind»
    ConceptNode
«Georgraphy»
MemberLink
    ConceptNode «Wave»
    ConceptNode
«Georgraphy»

MemberLink
    ImplicationLink
        ConceptNode «Wind»
        ConceptNode «Wave»
    ConceptNode
«Georgraphy»
```

```
MemberLink
    ConceptNode «Cloud»
    ConceptNode «Georgraphy»
MemberLink
    ConceptNode «Rain»
    ConceptNode «Georgraphy»

MemberLink
    ImplicationLink
        ConceptNode «Cloud»
        ConceptNode «Rain»
    ConceptNode «Georgraphy»
```

OpenCog AtomSpace, what's inside

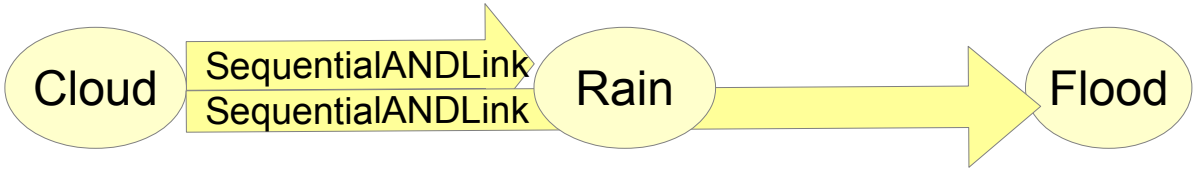


https://github.com/opencog/atomspace/blob/master/opencog/atoms/base/atom_types.script

OpenCog Atoms - example

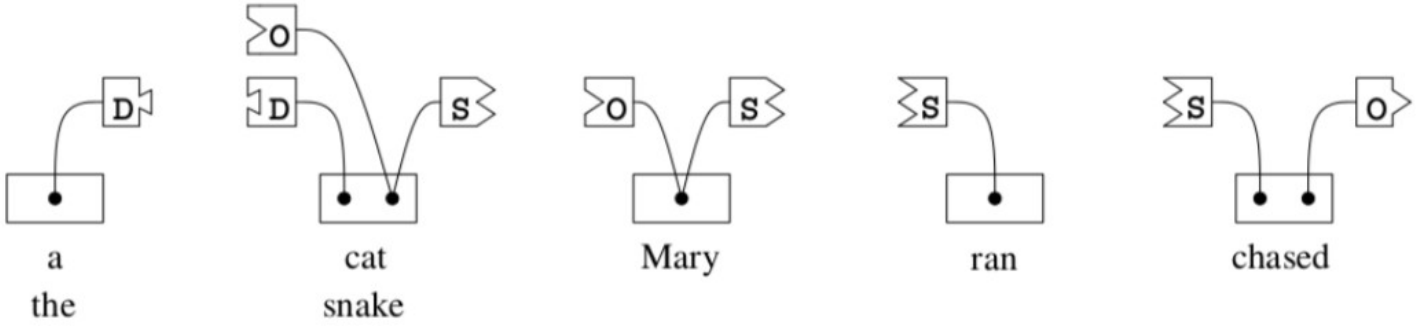
Id	Type	Space	Name (Label)	Level in Meta-Graph	Arity	Argu-ments	Truth Value: Count, Strength, Confidence
11	SequentialANDLink	5	-	2	2	13, 14	22 0.5 (22/44=50%) 1.0 («surely»)
12	SequentialANDLink	5	-	2	3	13, 14, 15	11 0.25 (11/44=25%) 0.5 («probably»)
13	ConceptNode	5	«Cloud»	0	0	-	44 1.0 1.0 («fact»)
14	ConceptNode	5	«Rain»	0	0	-	22 1.0 1.0 («fact»)
15	ConceptNode	5	«Flood»	0	0	-	11 1.0 0.5 («just news»)

Atom Type is used to infer if atom is link and if it is Ordered/directed or Unordered/undirected



https://github.com/opencog/atomspace/blob/master/opencog/atoms/base/atom_types.script

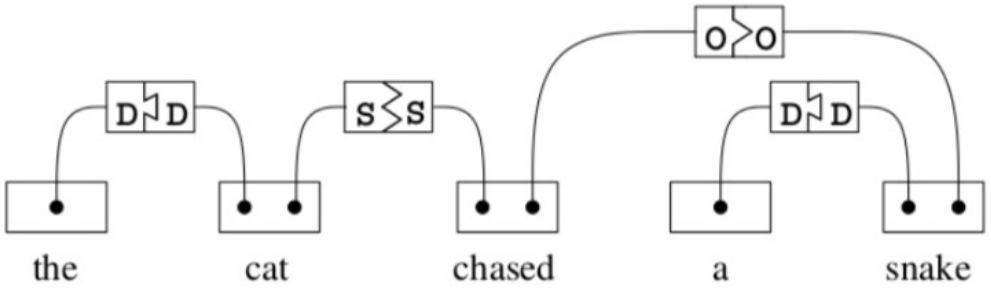
OpenCog Language Learning Domain (Link Grammar – Disjuncts & Connectors)



An illustration of Link Grammar connectors and disjuncts. The connectors are the jigsaw-puzzle-shaped pieces; connectors are allowed to connect only when the tabs fit together. A disjunct is the entire (ordered) set of connectors for a word. As lexical entries appearing in a dictionary, the above would be written as

```
a the: D+;
cat snake: D- & (S+ or O-);
Mary: O- or S+;
ran: S-;
chased: S- & O+;
```

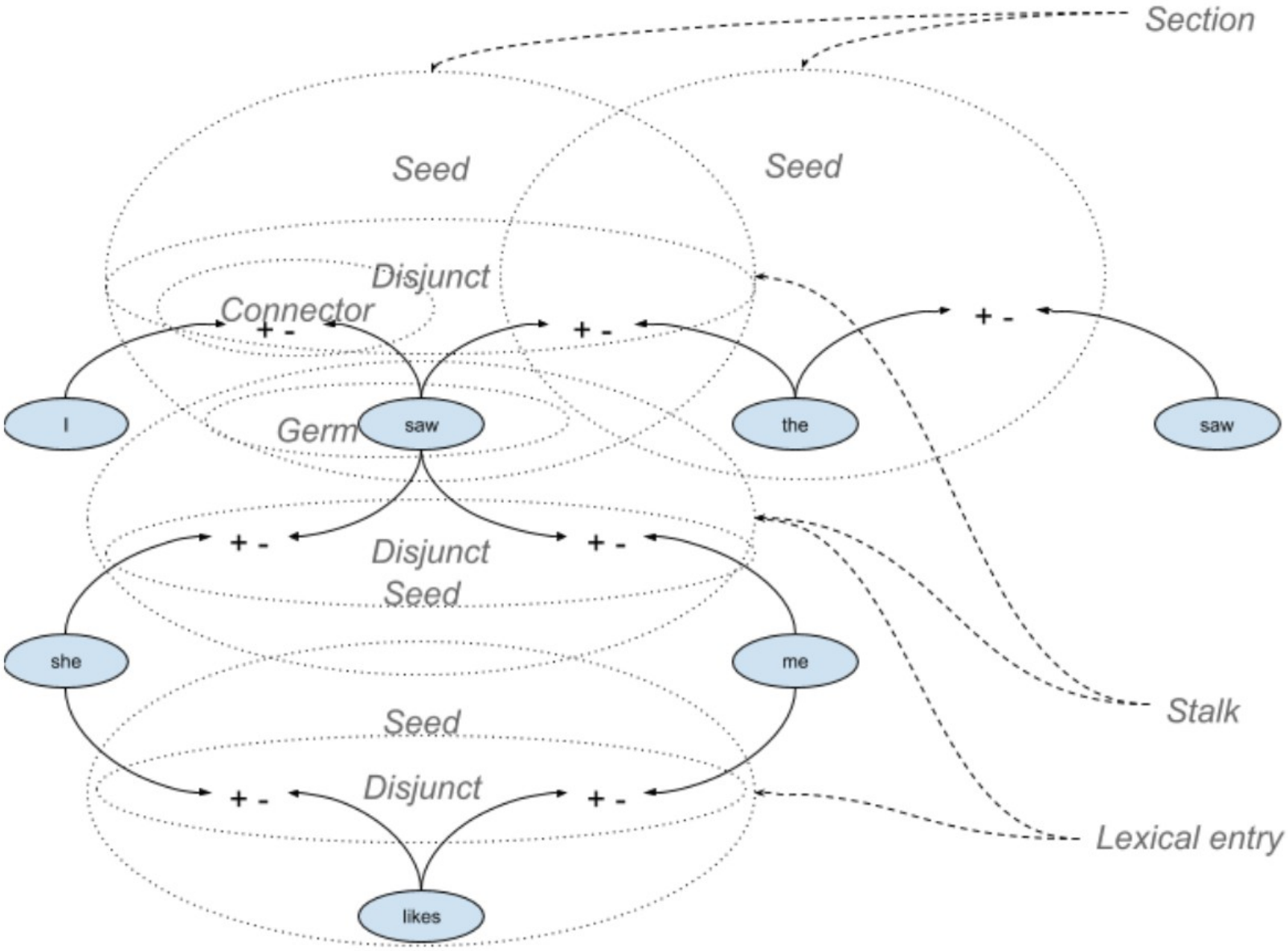
Note that although the symbols ‘‘&’’ and ‘‘or’’ are used to write down disjuncts, these are ***not*** Boolean operators, and do ***not*** form a Boolean algebra. They do form a non-symmetric compact closed monoidal algebra. The diagram below illustrates puzzle pieces, assembled to form a parse:



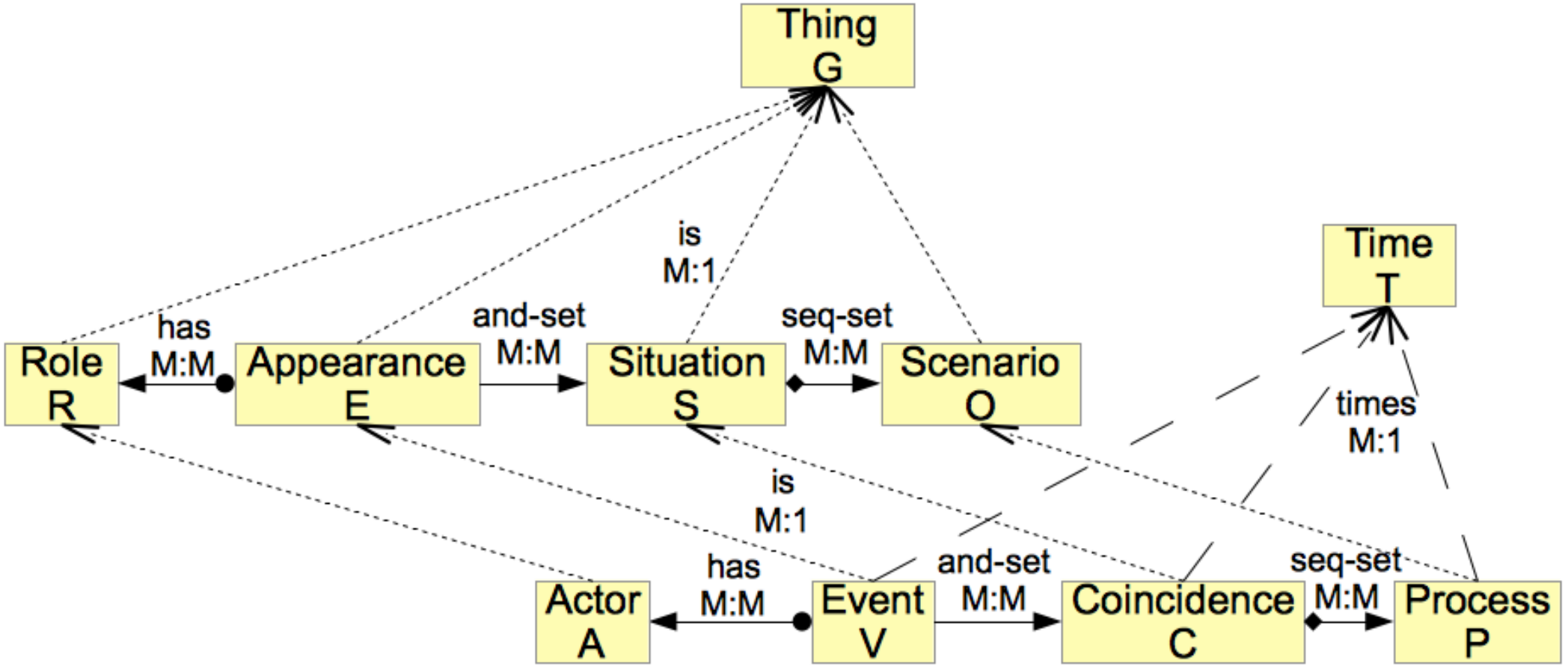
B. Goertzel, L. Vepstas, 2014

https://arxiv.org/abs/1401.3372

OpenCog Language Learning Domain (Lexical Entries, Stalks, Sections)



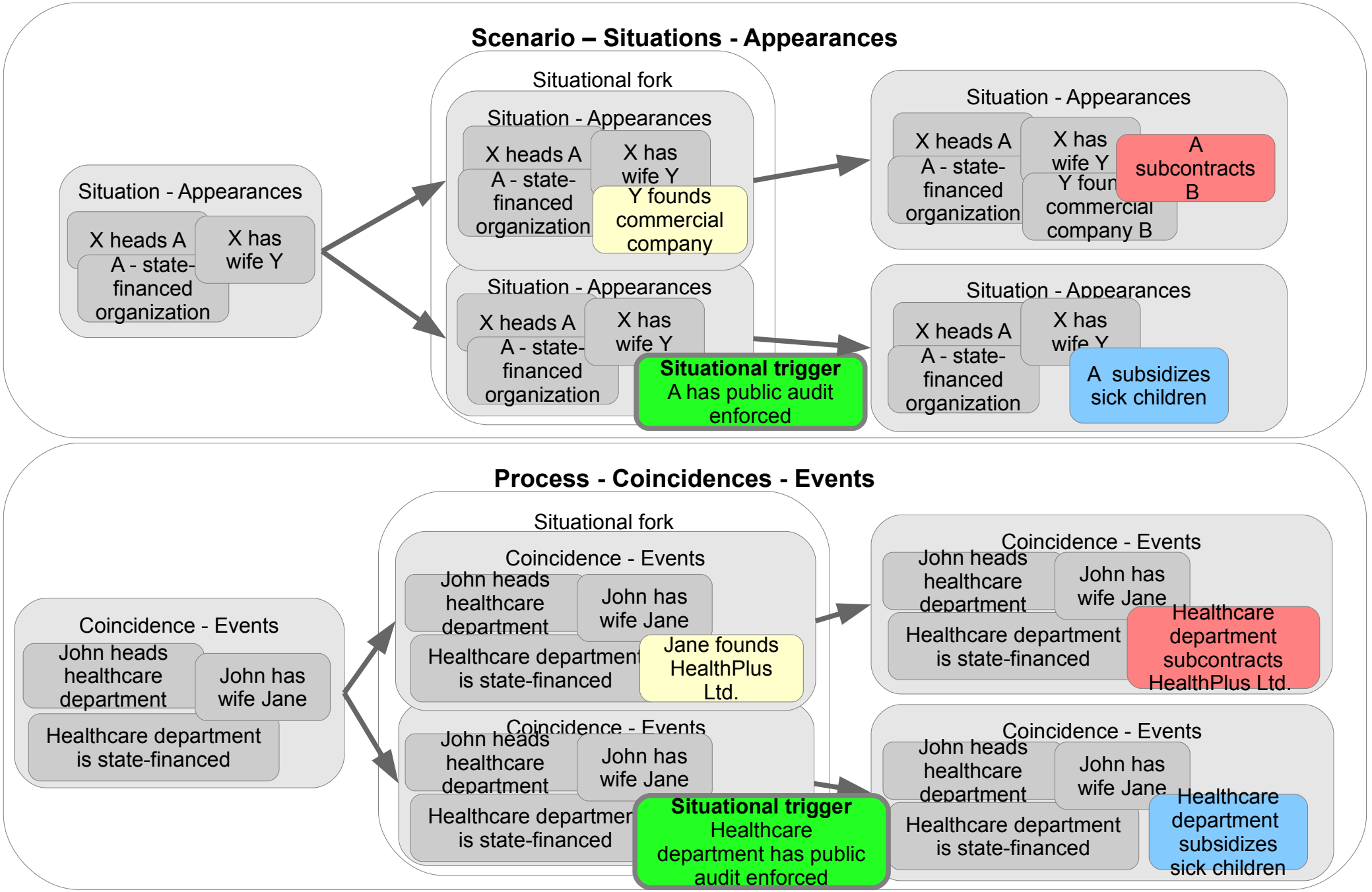
Scenarios and Processes - Unified Ontology



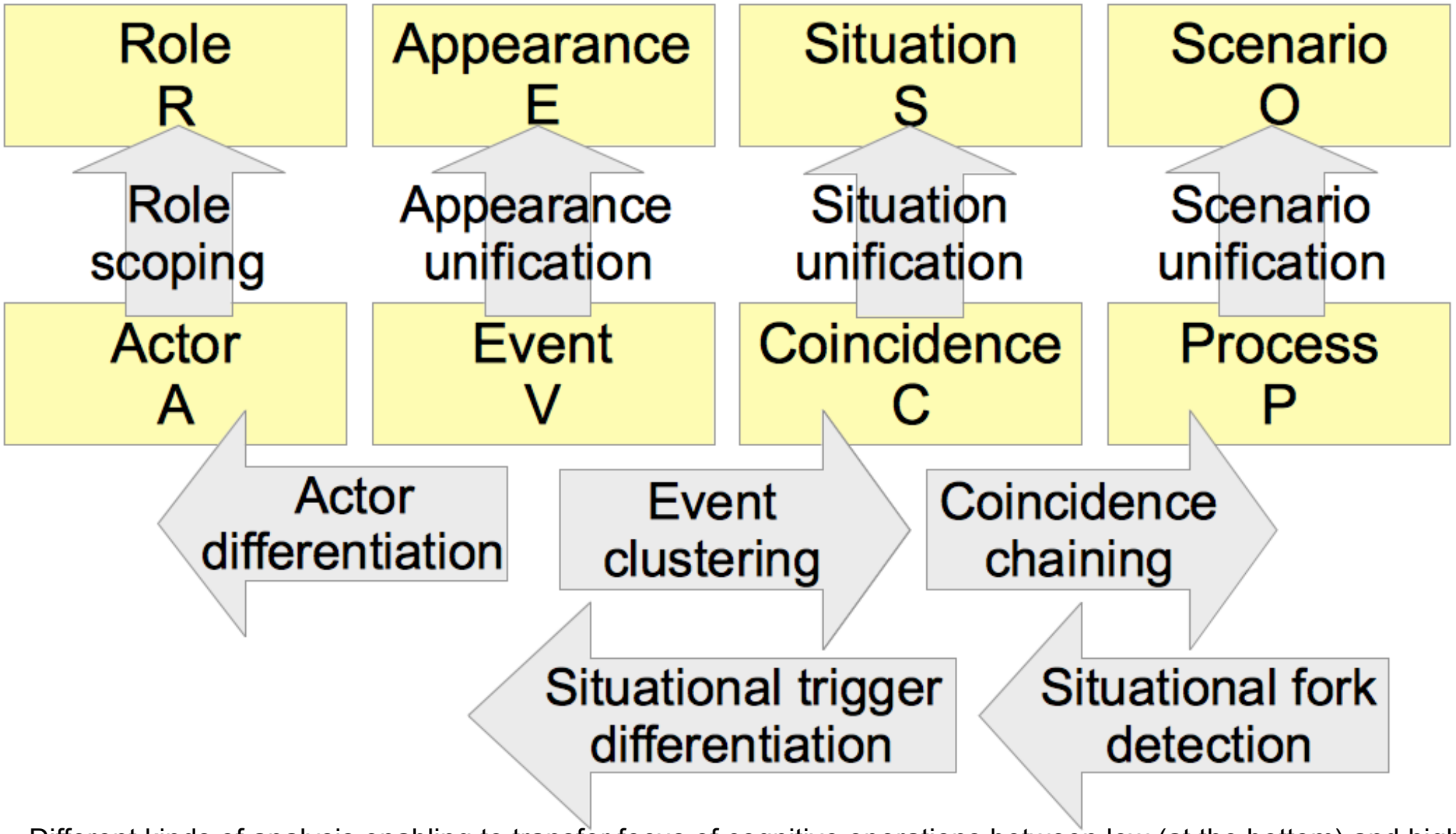
Data model in use with types of relationships indicated

https://arxiv.org/abs/1807.02072

Scenarios and Processes - Real Case



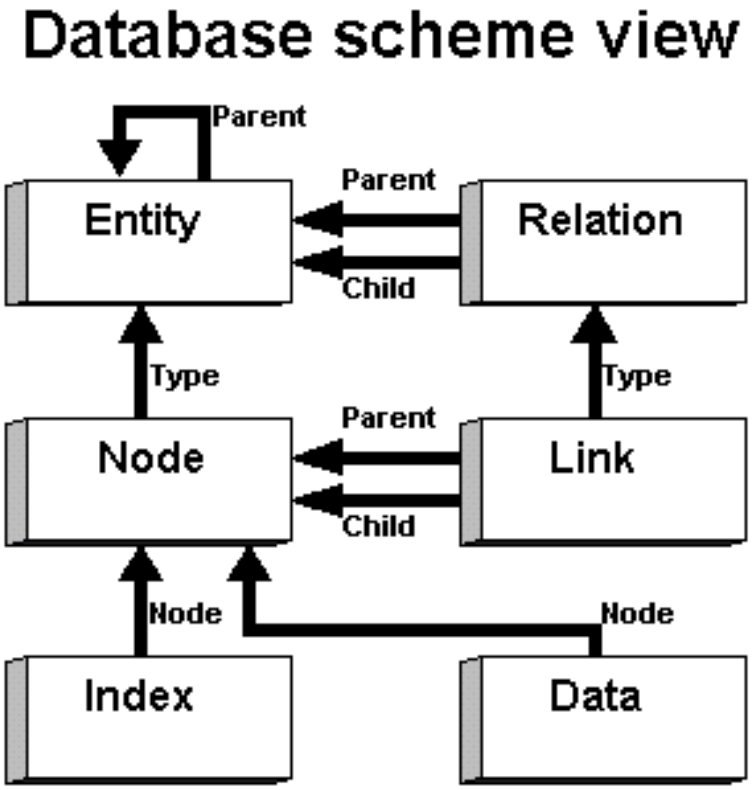
Scenarios and Processes - Cognitive Activities



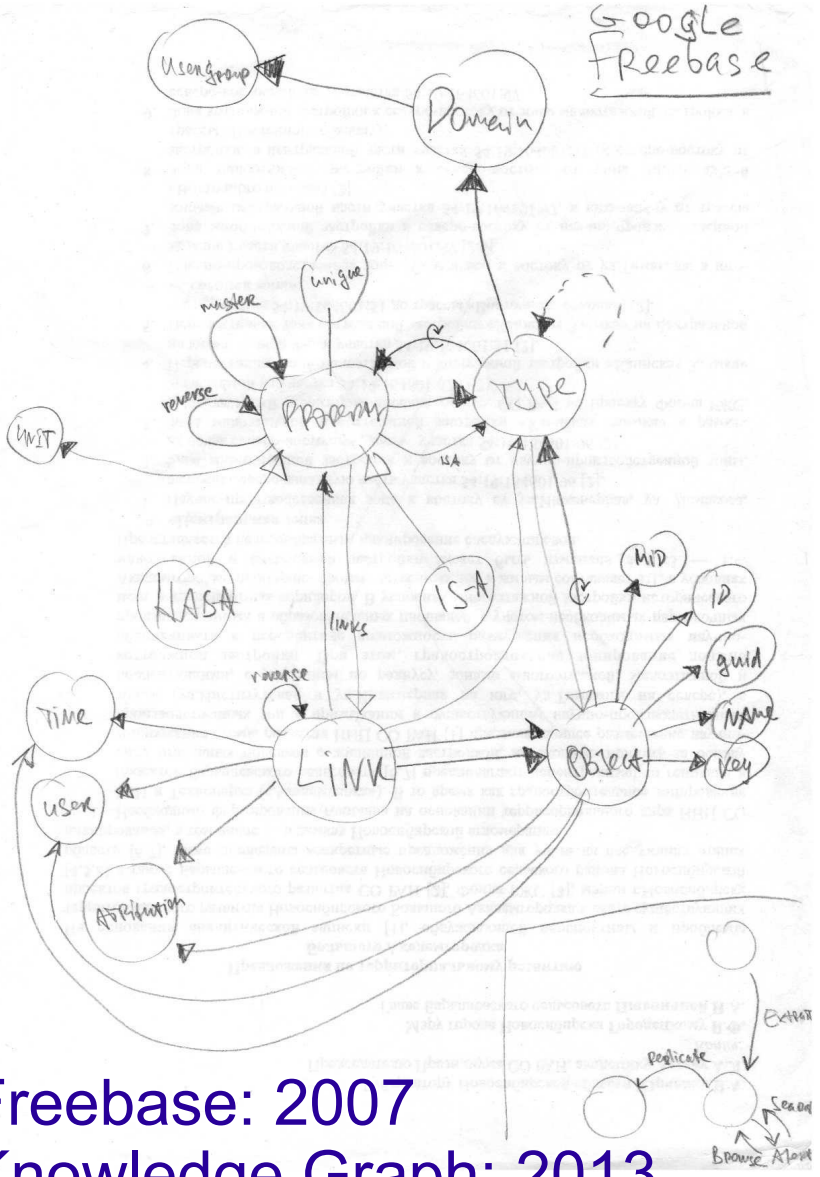
Different kinds of analysis enabling to transfer focus of cognitive operations between low (at the bottom) and high (at the top) abstraction levels and different degrees of aggregation – atomic (on the left) and ordered (on the right).

https://arxiv.org/abs/1807.02072

Foundation Ontologies - History



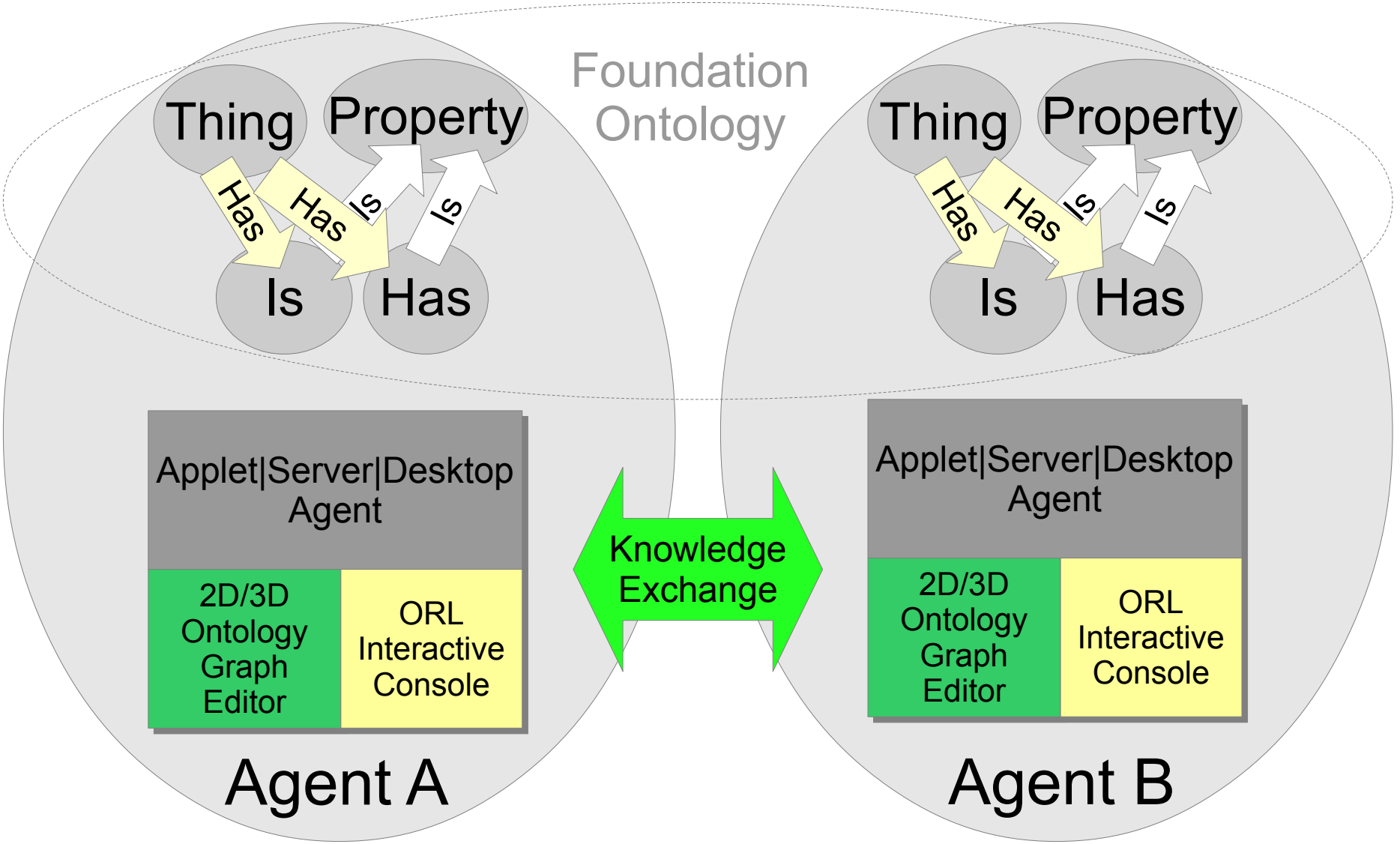
Data4: 1995



Freebase: 2007
Knowledge Graph: 2013

http://aigents.com/papers/2019/ontologies-language-aigents-2019.pdf

Distributed Knowledge Editor
Webstructor: 2001-2006



http://aigents.com/papers/2019/ontologies-language-aigents-2019.pdf

Semantic Modeling of Business Processes

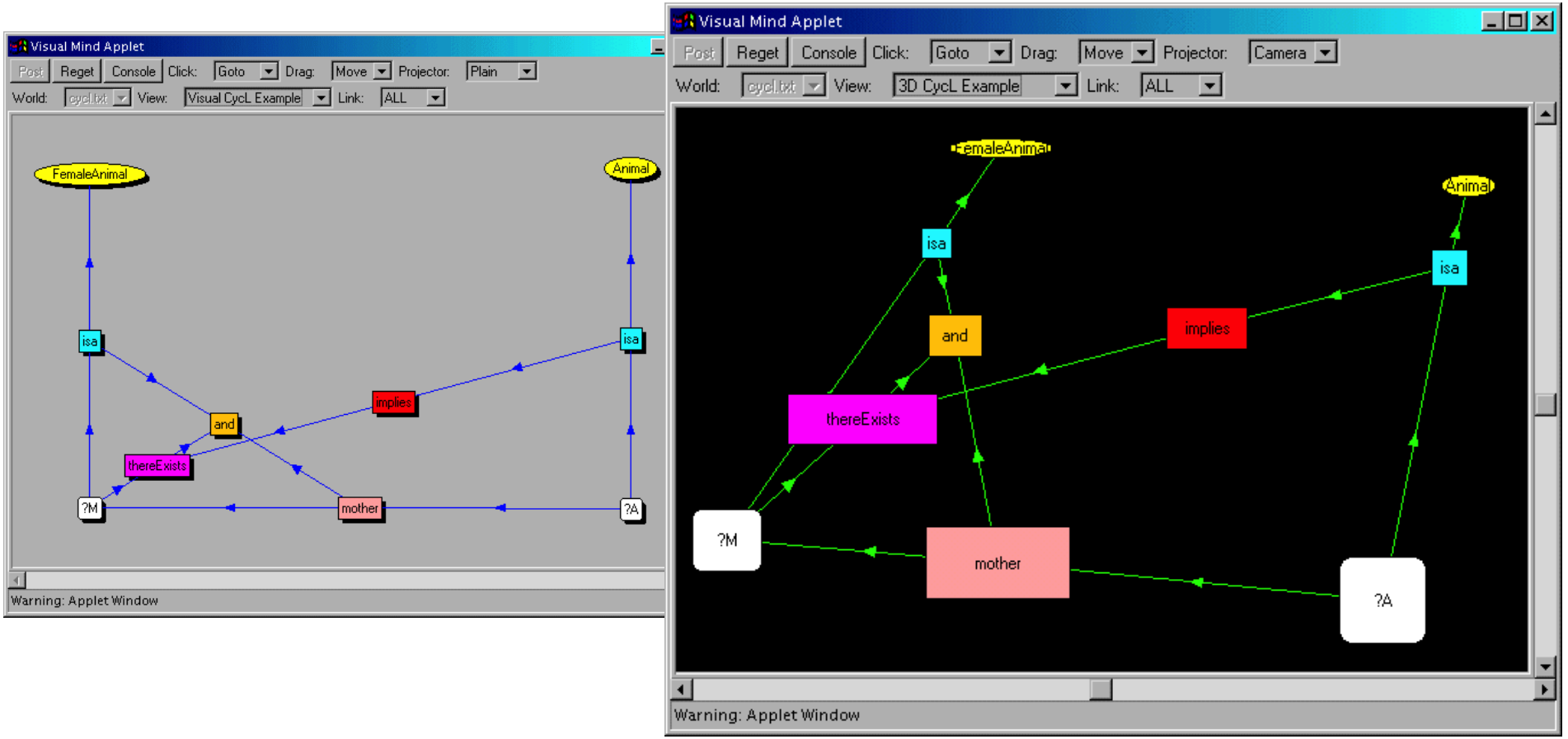
Object Relational Language (ORL)

English	ORL
Here are the items A, B and C where A has properties X and Y while B and C are in relationship Z.	ITEM A,B,C;; A HAS (X), (Y);; B Z(C);;
In order to reach goal 1 one needs condition 2 and 3 to be held true while 2 can be true only if condition 4 happens.	CONDITION C2,C3,C4;; GOAL G1 REQUIRES (C2),(C3);; CONDITION(C2) REQUIRES (C4);;
Each morning need to perform this and that in order, having such and such done at once next.	PROCESS TIME “8:00”; REPEAT (DAILY); ORDER DO THIS, DO THAT;, FORK DO SUCH, DO SUCH;;;
What is that my stuff you mentioned yesterday or the day before?	STUFF(OWNER (ME), UPDATE (AUTHOR (YOU), {TIME “2013-03-22”, TIME “2013-03-21”}).TELL;
What were the relationships between P and Q last year?	PROPERTY(OWNER (P), THING (Q), TIME “2012”).TELL;
Let me know once they roll out next version of the product.	DO EMAIL TO “me@at.org”;; WHEN PRODUCT(VENDOR (THEY)).VERSION CHANGE;;

http://aigents.com/papers/2019/ontologies-language-aigents-2019.pdf

Webstructor: Distributed Knowledge Editor

Visual 2D/3D editor of logical assertions (e.g. CycL)



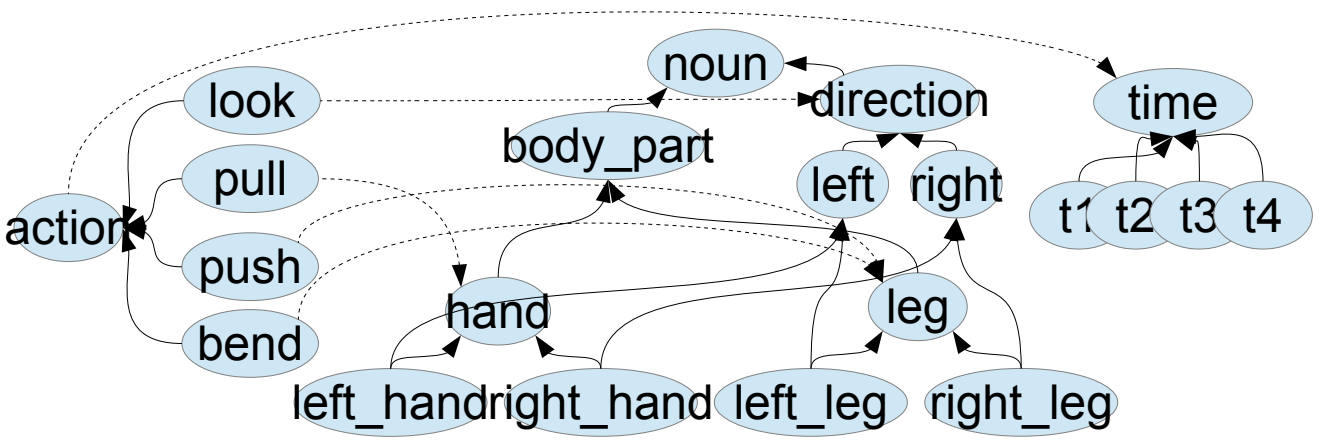
(implies (isa ?A Animal) (thereExists ?M (and (mother ?A ?M) (isa ?M FemaleAnimal))))

=

Mother of an animal is a female animal.

http://aigents.com/papers/2019/ontologies-language-aigents-2019.pdf

Semantic modeling using **Premise** language



push(right_leg, t1);
push(left_leg, t1);
pull(right_hand, t1).

push(left_leg, t2);
pull(right_hand, t2);
pull(left_hand, t2);
greater(t2,t1);

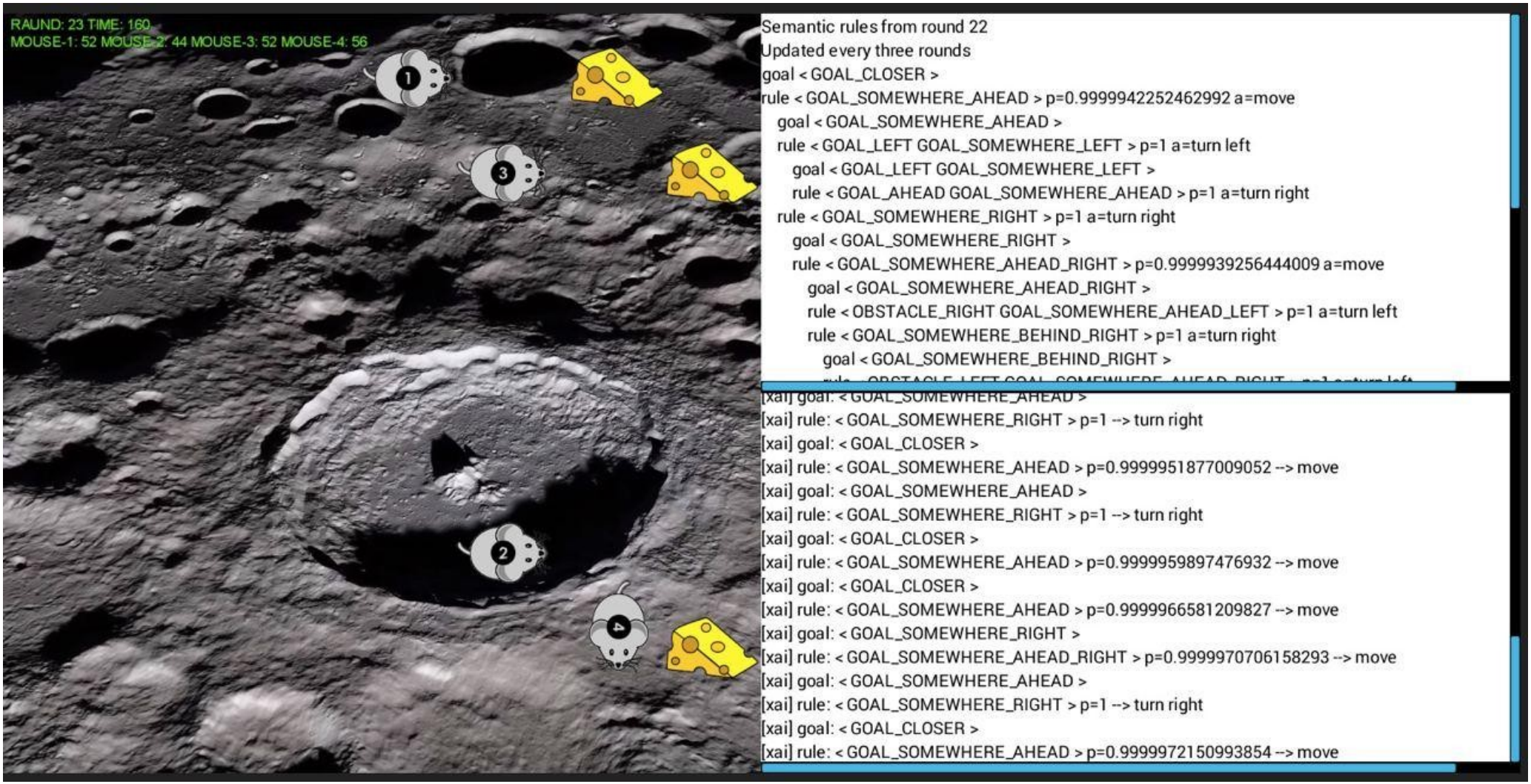
look(right,t3);
pull(left_hand,t3);
greater(t3,t2);

pull(right_hand,t4);
bend(right_leg,t4);
bend(left_leg,t4);
greater(t4,t3).

```
;;; Aigents.pr

(module Aigents

  (enumeration Action look pull push bend)

  (enumeration BodyPart hand leg)

  (enumeration Direction left right)

  (enumeraton Hand left_hand right_hand)

  (enumeration Leg left_leg right_leg)

  (enumeration Time t1 t2 t3 t4)

  (relation Instance
    :Category
    :Instance
  )

  (new Instance :Category left :Instance left_hand)

  (new Instance :Category left :Instance left_leg)

  (new Instance :Category right :Instance right_leg)

  (new Instance :Category right :Instance right_leg)

  (relation Statement
    :Verb
    :Object
    :Adverb
  )
  (relation Greater :More :Less)

  (new Statement :Verb push :Object right_leg :Adverb t1)
  (new Statement :Verb push :Object left_leg  :Adverb t1)
  (new Statement :Verb pull :Object right_hand :Adverb t1)

  (new Statement :Verb push :Object left_leg :Adverb t2)
  (new Statement :Verb pull :Object right_hand :Adverb t2)
  (new Statement :Verb pull :Object left_hand :Adverb t2)
  (new Greater :More t2 :Less t1)

  (new Statement :Verb look :Object right :Adverb t3)
  (new Statement :Verb pull :Object left_hand :Adverb t3)
  (new Greater :More t3 :Less t2)

  (new Statement :Verb pull :Object right_hand :Adverb t4)
  (new Statement :Verb bend :Object right_leg  :Adverb t4)
  (new Statement :Verb bend :Object left_leg   :Adverb t4)
  (new Greater :More t4 :Less t3)

)
```

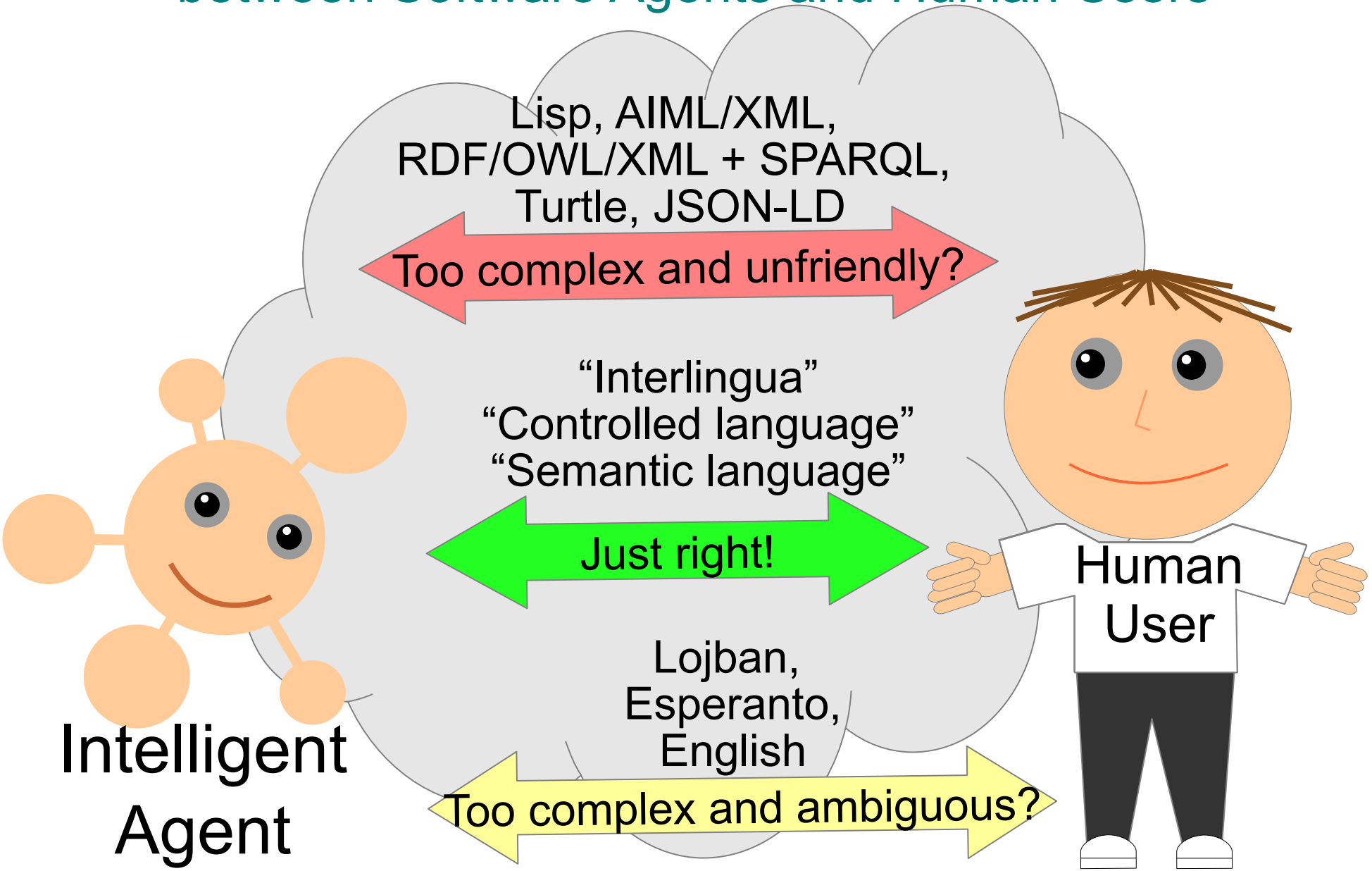
The Premise Reference by Michael S. P. Miller
(ISBN 978-1795680127)
https://www.amazon.com/Premise-Language-Building-Minds-Patterns/dp/1983178136

Reinforcement Learning for a Goal-Driven Behavior as a Probabilistic Semantic Program (D0SL)



http://www.math.nsc.ru/AP/ScientificDiscovery/PDF/muhortov_khlebnikov_vityaev_neuroinformatics_v6.pdf
http://www.math.nsc.ru/AP/ScientificDiscovery/soft/FS.html
https://www.youtube.com/watch?v=985Fw3yKeOM
https://d0sl.org/en/

Aigents® - "Semantic Language" for Knowledge Transfer between Software Agents and Human Users



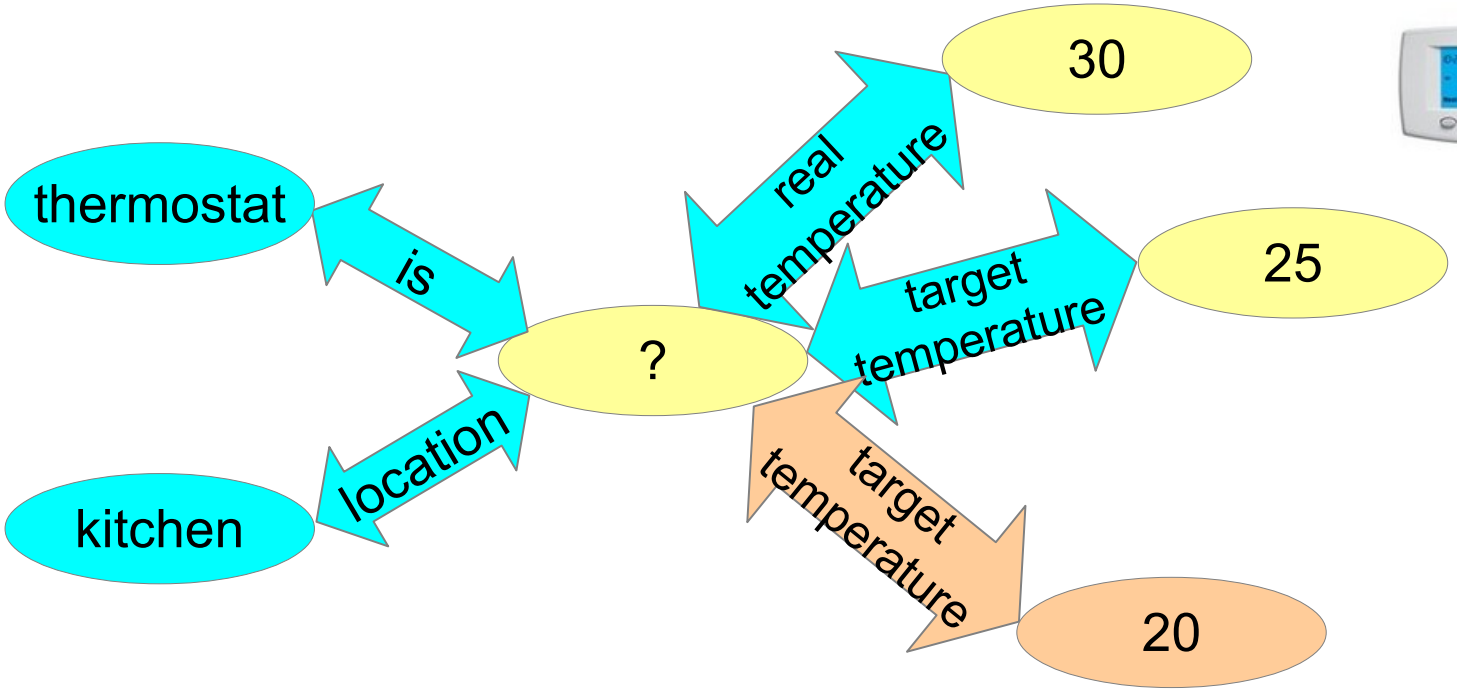
http://aigents.com/papers/2019/ontologies-language-aigents-2019.pdf

Aigents® Language (AL) as a Graph Manipulation

Interrogation:
What is thermostat, location kitchen real temperature, target temperature?

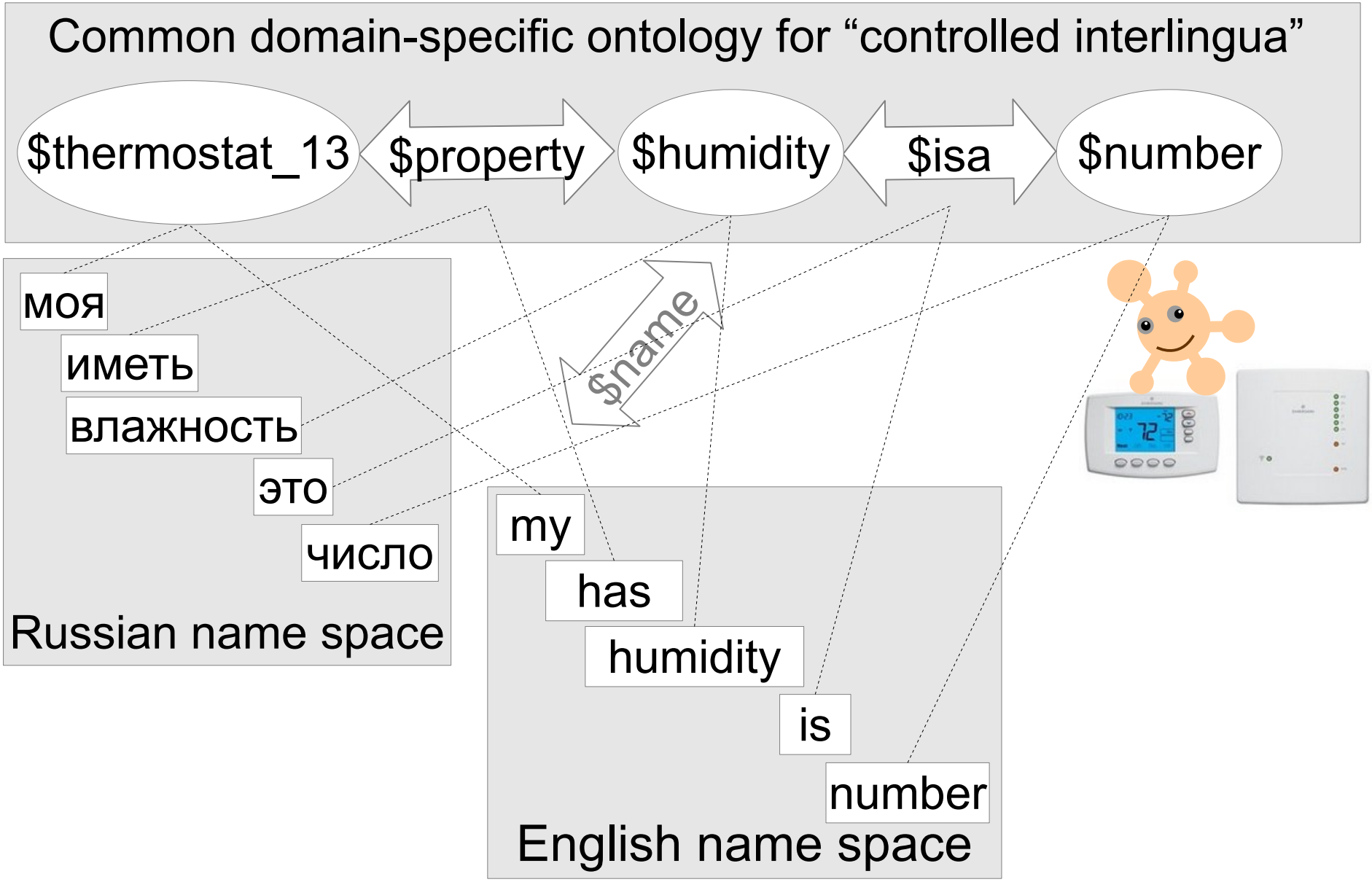
Declaration:
Is thermostat, location kitchen real temperature 30, target temperature 25.

Direction:
Is thermostat, location kitchen target temperature 20!



http://aigents.com/papers/2019/ontologies-language-aigents-2019.pdf
http://aigents.com/papers/2015/ZONT-2015-Agent-Language-Kolonin.pdf

Aigents® Language (AL) as a “Labeled Turtle”



http://aigents.com/papers/2019/ontologies-language-aigents-2019.pdf
http://aigents.com/papers/2015/ZONT-2015-Agent-Language-Kolonin.pdf

Aigents® Language (AL) - EBNF

```
<message> := ( <statement> | <acknowledgement> )*
<acknowledgement> := ( 'ok' | ('true' | 'yes' | <number>) | ('no' | 'false' | 0) ) '.'
<statement> := <interrogation> | <confirmation> | <declaration> | <direction>
<interrogation> := 'what' ? <expression> '?'        (* “open” incomplete graph *)
<confirmation> := 'if' ? <expression-set> '?'       (* “closed” complete graph  *)
<declaration> := ( <expression-set> ) '.'           (* “closed” complete graph *)
<direction> := 'do' ? <expression-set> '!'              (* “closed” complete graph *)
<expression> := <term> (' ' <term>)*                (* separated by spaces *)
<expression-set> := <all-set> | <any-set> | <seq-set>       (* different kinds of sets *)
<term> := <negation>? ( <anonymous>? | <self> | <peer> | <id> | <name> | <value> | <qualifier> )
<qualifier> := <expression> | <expression-set>
<any-set> := <or-list> | ( '{' <or-list> '}' )
<all-set> := <and-list> | ( '(' <and-list>  ')' )
<seq-set> := <next-list> | ( '[' <next-list> ']' )
<or-list> := <expression> ( (',' | 'or' ) <expression> )*
<and-list> := <expression> ( (',' | 'and' ) <expression> )*
<then-list> := <expression> ( (',' | 'next' ) <expression> )*
<negation> := 'not' | 'no' | '~'
<anonymous> := ('there' ('is'|'are')) | 'any' | 'anything' ?
<self> := 'my'|'i'|'we'|'our'
<peer> := 'your'|'you'
<value> := <number> | <date> | <time> | <string>
```

That is all!
The rest is done by means of domain-specific ontology and providing national-specific name space

http://aigents.com/papers/2019/ontologies-language-aigents-2019.pdf
http://aigents.com/papers/2015/ZONT-2015-Agent-Language-Kolonin.pdf

Thank you and welcome!

Anton Kolonin

akolonin@aigents.com

Facebook: akolonin

Telegram: akolonin

https://aigents.com

https://facebook.com/groups/agirussia

https://t.me/agirussia

Novosibirsk State University

*THE REAL SCIENCE

SingularityNET

https://singularitynet.io