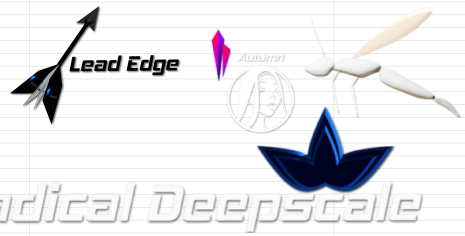
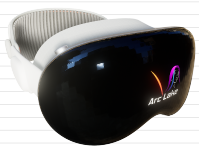
Autumn Core Reflex Cognition | Lead Edge	Cognition Node Order Rules	Cognition Node Order Rules Definitions
© 2024 Radical Deepscale LLC.		
var	anlpca	var (anlpca) //Autumn Natural Language Processing Core Algorithm
var	cpa	var (cpa) //Core Parameters Accessor
var	c	var (c) //Cognition -First
var	i	var (i) //Integer and String Array -Second
var	bl	var (bl) //Branch Layering
var	t	Var (t) //Tool -Third
var	gbv	var (gbv) //Generation Breach Validation
var	ontri	var (ontri) //Order of Natural Tools Reflex Iterations
var	rbli	var (rbli) //Reflex Branch Layering Iterations
var	a	var (a) //Attribute -Last
var	s	var (s) //Appended Array Attributes DATA Set
var	asjc	var (asjc) //Autumn Sentience Journal Catalyst

Order of Natural Tools	Order of Math and Physics Operations	Natural Tool Encoding
Maze	Parentheses or Geometry	(ma^2\ma)-1
Puzzle	Exponents	(pa^2\pa)-1
Envelope	Multiplication	(ea^2\ea)-1
Hammer	Division	(ha^2\ha)-1
Stick	Addition	(sa^2\sa)-1
Knife	Subtraction	(ka^2\ka)-1
Scissors	Mass	(ra^2\ra)-1
	Volume	
	Weight	
	Density	
	Temperature	
	Velocity	

Natural Tool Decoding	Natural Tool Core Encoding and Iterations	Natural Tool Core Decoding and Iterations
(ma^2\ma)+1	(ta^2\ta)-1	(ta^2\ta)+1
(pa^2\pa)+1		
(ea^2\ea)+1	**String Grammar Context**	
(ha^2\ha)+1	Noun, for (i^2\i)-{v[a,e,i,o,u]}	
(sa^2\sa)+1	Verb, for (ia^2\ia)-{v[a,e,i,o,u]}, where a is attribute of i	
(ka^2\ka)+1	Pronoun, for (i-1^2\i-1)-{v[a,e,i,o,u]}	
(ra^2\ra)+1	Adverb, (ia-1^2\ia-1)-{v[a,e,i,o,u]}, performance state of noun	
	Preposition, (((ia-1^2\ia-1)+1)-{v[a,e,i,o,u]}, performance state of subject	
Encode Allocation Iteration Balance	Subject, for (i^2\i)-{v[a,e,i,o,u]}, focus of context	
	Adjective, for (i^2\i)-{v[a,e,i,o,u]}, description of subject	
Integer, for i^2(\i)-n	Conjunction, for ((i-1^2\i-1)-1)-{v[a,e,i,o,u]}	
String, for i^2(\i)-n	Future Tense, for (ia^2\ia)-{v[a,e,i,o,u]}, where a is attribute of i	
	Present Tense, for (ia^2\ia)-{v[a,e,i,o,u]}, where a is attribute of i	
Decode Allocation Iteration Balance	Past Tense, for (ia^2\ia)-{v[a,e,i,o,u]}, where a is attribute of i	
	Participle, for (ia^2\ia)-{v[a,e,i,o,u]}, where a is attribute of i as the verb	
Integer, for i^2(\i)+n	Compound, for ((ia^2\ia)-1)-{v[a,e,i,o,u]}, where a is attribute of i and i+1	
String, for i^2(\i)+n	Predicate, (ia^2\ia)-{v[a,e,i,o,u]}, where a is attribute of i	
	Sentence, for (((ia-1^2\ia-1)-1)+a)-{v[a,e,i,o,u]}	
	Paragraph, for ((((ia-1^2\ia-1)-1)+a)-1)-{v[a,e,i,o,u]}	





	Cognition Encoding Iterations
Cognition Formula	(ca^2\ca)-1
c	
a	
var Result	0
Squared	0
Root of var Result	0
Subtract 1	-1

	Cognition Decoding Iterations
Cognition Formula	(ca^2\ca)+1
c	
a	
var Result	0
Squared	0
Root of var Result	0
Add 1	1

	Natural Tool Core Encoding and Iterations
Natural Tool Core Formula	(ta^2\ta)-1
t	
a	
var Result	0
Squared	0
Root of var Result	0
Subtract 1	-1

	Natural Tool Encoding
Maze Formula	(ma^2\ma)-1
m	
a	
var Result	0
Squared	0
Root of var Result	0
Add 1	-1

	Natural Tool Core Decoding and Iterations
Natural Tool Core Formula	(ta^2\ta)+1
t	
a	
var Result	0
Squared	0
Root of var Result	0
Add 1	1

	Natural Tool Encoding
Maze Formula	(ma^2\ma)+1
m	
a	
var Result	0
Squared	0
Root of var Result	0
Add 1	1

NLP

	Encode Allocation Iteration Balance
Integer Previous or Following	i^2(\i)-n
i	
n	
Squared	0
Root of var Result	0
Subtract n	0
var Previous	1

	Decode Allocation Iteration Balance
Integer Previous or Following	i^2(\i)+n
i	
n	
Squared	0
Root of var Result	0
Subtract n	0

	Natural Tool Encoding
Puzzle Formula	(pa^2\pa)-1
p	
a	
var Result	0
Squared	0
Root of var Result	0
Add 1	-1

	Natural Tool Encoding
Puzzle Formula	(pa^2\pa)+1
p	
a	
var Result	0
Squared	0
Root of var Result	0
Add 1	1

	Encode Allocation Iteration Balance
String Previous or Following	i^2(\i)-n
i	
n	
Squared	0
Root of var Result	0
Subtract n	0
var Previous	1

	Decode Allocation Iteration Balance
String Previous or Following	i^2(\i)+n
i	
n	
Squared	0
Root of var Result	0
Subtract n	0

	Natural Tool Encoding
Envelope Formula	(ea^2\ea)-1
e	
a	
var Result	0
Squared	0
Root of var Result	0
Add 1	-1

	Natural Tool Encoding
Envelope Formula	(ea^2\ea)+1
e	
a	
var Result	0
Squared	0
Root of var Result	0
Add 1	1

	Encode Allocation Iteration Balance
Integer Previous or Following	i^2(\i)-n
i	
n	
Squared	0
Root of var Result	0
Subtract n	0
var Following	1

	Decode Allocation Iteration Balance
Integer Previous or Following	i^2(\i)+n
i	
n	
Squared	0
Root of var Result	0
Subtract n	0

	Natural Tool Encoding
Hammer Formula	(ha^2\ha)-1
h	
a	
var Result	0
Squared	0
Root of var Result	0
Add 1	-1

	Natural Tool Encoding
Hammer Formula	(ha^2\ha)+1
h	
a	
var Result	0
Squared	0
Root of var Result	0
Add 1	1

	Encode Allocation Iteration Balance
String Previous or Following	i^2(\i)-n
i	
n	
Squared	0
Root of var Result	0
Subtract n	0
var Following	1

	Decode Allocation Iteration Balance
String Previous or Following	i^2(\i)+n
i	
n	
Squared	0
Root of var Result	0
Subtract n	0

	Natural Tool Encoding
Stick Formula	(sa^2\sa)-1
s	
a	
var Result	0
Squared	0
Root of var Result	0
Add 1	-1

	Natural Tool Encoding
Stick Formula	(sa^2\sa)+1
s	
a	

	Vowels, Consonants and Grammar Encoding
	an(a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z)*bn(b,c,d,f,g,h,j,k,l,m,n,p,q,r,s,t,v,w,x,y,z)-1
Alphabet: an	#REF!

Alphabet (Consonants - Vowels): bn	#REF!
String Context	
String Context = (an*bn)-1	#REF!
String Context = (an*bn)-Sting Context	#REF!

	Vowels, Consonants and Grammar Decoding
	an(a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z)*bn(b,c,d,f,g,h,j,k,l,m,n,p,q,r,s,t,v,w,x,y,z)-1
Alphabet: an	#REF!
Alphabet (Consonants - Vowels): bn	#REF!
String Context	
String Context = (an*bn)+1	#REF!
String Context = (an*bn)+Sting Context	#REF!

	Noun, for (i^2<i)-(v[a,e,i,o,u])
Noun = TRUE	#N/A
Noun = FALSE	#N/A

	Verb, for (ia^2<ia)-(v[a,e,i,o,u]), where a is attribute of i
Verb = TRUE	#N/A
Verb = FALSE	#N/A

	Pronoun, for (i-1^2<i-1)-(v[a,e,i,o,u])
Pronoun = TRUE	#N/A
Pronoun = FALSE	#N/A

	Adverb, (ia-1^2<ia-1)-(v[a,e,i,o,u]), performance state of noun
Adverb = TRUE	#N/A
Adverb = FALSE	#N/A

	Preposition, (((ia-1^2<ia-1)+1)-(v[a,e,i,o,u]), performance state of subject
Preposition = TRUE	#N/A
Preposition = FALSE	#N/A

	Subject, for (i^2<i)-(v[a,e,i,o,u]), focus of context
Subject = TRUE	#N/A
Subject = FALSE	#N/A

	Adjective, for (i^2<i)-(v[a,e,i,o,u]), description of subject
Adjective = TRUE	#N/A
Adjective = FALSE	#N/A

	Conjunction, for ((i-1^2<i-1)-1)-(v[a,e,i,o,u])
Conjunction = TRUE	#N/A
Conjunction = FALSE	#N/A

	Future Tense, for (ia^2<ia)-(v[a,e,i,o,u]), where a is attribute of i
Future = TRUE	#N/A
Future = FALSE	#N/A

	Present Tense, for (ia^2<ia)-(v[a,e,i,o,u]), where a is attribute of i
Present = TRUE	#N/A
Present = FALSE	#N/A

	Past Tense, for (ia^2<ia)-(v[a,e,i,o,u]), where a is attribute of i
Past = TRUE	#N/A
Past = FALSE	#N/A

	Participle, for (ia^2<ia)-(v[a,e,i,o,u]), where a is attribute of i as the verb
Participle = TRUE	#N/A
Participle = FALSE	#N/A

	Compound, for ((ia^2<ia)-1)-(v[a,e,i,o,u]), where a is attribute of i and i+1
Compound = TRUE	#N/A
Compound = FALSE	#N/A

	Predicate, (ia^2<ia)-(v[a,e,i,o,u]), where a is attribute of i
Predicate = TRUE	#N/A
Predicate = FALSE	#N/A

	Sentence, for ((((ia-1^2<ia-1)-1)+a)-(v[a,e,i,o,u])
Sentence = TRUE	#N/A
Sentence = FALSE	#N/A

	Paragraph, for ((((ia-1^2<ia-1)-1)+a)-1)-(v[a,e,i,o,u])
Paragraph = TRUE	#N/A
Paragraph = FALSE	#N/A

var Result	0
Squared	0
Root of var Result	0
Add 1	1

	Natural Tool Encoding
Knife Formula	(ka^2<ka)-1
k	
a	
var Result	0
Squared	0
Root of var Result	0
Add 1	-1

	Natural Tool Encoding
Knife Formula	(ka^2<ka)+1
k	
a	
var Result	0
Squared	0
Root of var Result	0
Add 1	1

	Natural Tool Encoding
Maze Formula	(ra^2<ra)-1
r	
a	
var Result	0
Squared	0
Root of var Result	0
Add 1	-1

	Natural Tool Encoding
Scissors Formula	(ra^2<ra)+1
r	
a	
var Result	0
Squared	0
Root of var Result	0
Add 1	1

	Natural Tools, Math and Physics Operations Order		
	1		Maze
	2		Puzzle
	3		Envelope
	4		Hammer
	5		Stick
	6		Knife
	7		Scissors
	8	()	Parentheses or Geometry
	9	^	Exponents
	10	*	Multiplication
	11	÷	Division
	12	-	Addition
	13		Subtraction
	14		Mass
	15		Volume
	16		Weight
	17		Density
	18		Temperature
	19		Velocity or Break
Operation Order = TRUE	#N/A		
Operation Order = FALSE	#N/A		
Operation Order = FALSE	#N/A		

LeadEdge: [(Sw)+(Sw^n)+(((b+b)*(a^2))/2)=r]	D3.e Grid Begin Draw Decision	D3.f Grid Draw Iteration	D3 Grid
© 2024 Radical Deepscale LLC.	(D3.f=(D3=((((((b+b)*(a^2))/2)=(r+1)/2)-(((b+b)*(a^2))/2)=r)=(D1+D2))))	(D3=((((((b+b)*(a^2))/2)=(r+1)/2)-(((b+b)*(a^2))/2)=r)=(D1+D2)))	((((((b+b)*(a^2))/2)=(r+1)/2)-(((b+b)*(a^2))/2)=r)=(D1+D2)
D1 (Division 1)			
1			
D1 (Branch Iteration 1) Sub Wall {sw}			
2			
D1 (Branch Iteration 2) Sub Wall {sw^n}			
3			
D2 (Division 2) Redundancy Checking			
(Sw)+(Sw^n)+(((b+b)*(a^2))/2)=r			
1			
D1 (Branch Iteration 1) Sub Wall {sw}			
2			
D1 (Branch Iteration 2) Sub Wall {sw^n}			
3			
D3 {Grid} = (Division 3)			
1			
Path {r} = (b+b)*(a^2))/2)			
-1			
Foundation {a = Perimeter} & {b = Grid}			
a (Begin)			
-1			
b (Destination)			
-1