Theorem prover Meth8 applies four valued Boolean logic for modal interpretation

Point of contact:

Colin James III, Director
Ersatz Systems Machine Cognition, LLC
Colorado Springs, CO 80904-5220 USA

Email: info@cec-services.com
Mobile: +011 (719) 210 – 9534

Websites: ersatz-systems.com; cec-services.com

Overview

- Theorem prover for modal logic : the *2-tuple*
- Jan Łukasiewicz : $Ł_4$
- Garry Goodwin : $Ł_4$ Variants
- Test results : “A as true does not imply necessarily A.”

Overview: modal logic theorem prover

- 2-tuple as 2-sides:
  - *left bit is sinister* **false 0**;
  - *right bit is dexter* **true 1**
- Bivalent and decidable:
  - not vector space;
  - not undecidable
- Scalable and expandable
  - 8-bit version has 256 connectives each as its own truth table for *non* quantified expressions.
  - 16-bit version has 65,536 connectives each as its own truth table for quantified expressions.

Overview: Jan Łukasiewicz

- Untenable $Ł_4$:
  - Béziau : ◊A & ◊B → ◊(A & B)

    If possibly Booth killed Lincoln and possibly Booth never killed anyone, then it is possible that Booth both killed Lincoln and never killed anyone.
  - Font, Hájek □A → (◊B → □B)

    Necessarily every coin has two sides implies that if possibly the next flip of the coin lands heads, then necessarily the coin lands heads.
- Prefix or Polish notation [+ 3 4] versus:
  - Infix notation [(3)+(4)]
  - Postfix or Reverse Polish notation (fewer parens) [3 4 +]

Overview: Garry Goodwin
(Website: semantic-qube.com.uk)

- Multi valued logic
  - Cartesian graphing in N-dimensions
  - Color mapping in RGB
  - Quantified axiom mappings in the tesseract
- Modal logic in plausable variants of $Ł_4$ as matrix $Ł_{4.M9}$, $Ł_{4.M13}$
  - 3 variants with 3 options in a system of 10 models
  - With and without *R* reference for worlds and frames
  - Multi valued Boolean logic leads to incompleteness
  - "Some arguments which are never false fail to be theorems."
- Postfix notation: no ambiguity of ◊□A in i order as $A◊^{ii}□^{i}$

Acronym

- Meth8: **Me**chanical **Th**eorem in **8**-bits
  - Based on the 256-connectives of **8**-bits:

    Connective number below is 0011 0011 [conditional] 0101 0101.

    ```
    & AND:  17 0001 0001: 00 00 00 00  00 01 00 01  00 00 10 10  00 01 10 11
    > IMP: 221 1101 1101: 11 11 11 11  10 11 10 11  01 01 11 11  00 01 10 11
    ```
  - Distinct from 65536-connectives of 16-bits:

    For proofs with existential ∃ and universal ∀ quantifiers

    ```
    & AND: 4369: 00010001 00010001: 00 00 00 00  00 00 00 00..11 00 11 01  11 10 11 11
    > IMP:56797: 11011101 11011101: 11 11 11 11  11 11 11 11..11 00 11 01  11 10 11 11
    ```
- *Demo* Meth8 is free with literals scaled down.
  - Two literals for propositions: p, q
  - Two literals for theorems: A, B

Meth8 logical expressions

- Antecedent and Consequent
  - Literal type:

    13 theorems: A, B, C, D, E, F, G, H, I, J, K, L, M

    13 propositions: n, o, p, q, r, s, t, u, v, w, x, y, z
  - Modifiers: # necessarily □ ; % possibly ◊ ; ~ not ¬

    3 affirmed: [none], #, %

    3 disaffirmed: ~ , ~#, ~%
- Conditional: AND & ; OR + ; IMP > ; EQV =
  - 4 affirmed: &, +, >, =
  - 4 disaffirmed: ~&, ~+, ~>, ~=

Example: Meth8 data structure

```
LET literal         =  2        ! theorem = 1; proposition = 2
LET antecedent  = 13            ! A, … , M, n , … , z
LET consequent = 13             ! A, … , M, n , … , z
LET modifier     =   6          ! none, # [], % <>, ~, ~#, ~%
LET conditional  =  8           ! null, &, +, >, ~, ~&, ~+, ~>
LET model        = 10           ! based on 3 variants, 10 models
DIM expr$( 0, 0, 0, 0, 0, 0)    ! = 4-byte string, 32-bits
MAT REDIM expr$( 2, 13, 6, 8, 10)      ! 5-dimensions
```

Example: Meth8 memory footprint

- Literal expressions: 13 theorems; 13 propositions
- Combination of antecedent, consequent, conditional

  Antecedent (6 modifiers * 13 literals)

  * Consequent (6 modifiers * 13 literals)

  * Conditional (8 connectives) = 46,208 expressions
- Combination of literal segments and number of modes

  * 2 segments * 10 models = 924,160 combinations
- Memory size (8-bits/row * 4 rows/ able = 32-bits = 4-bytes)

  * 4-bytes per expression = 3,696,640 bytes ≈ 3.6 MB

Meth8 paren parsing : Sliding Window

1. Map L/R parens
2. Slide window L
3. Tag first adjacent L/R pair [*04*, *09*]; then next [12,17]
4. Write stack list [*04*, *09*], [12, 17]
5. Match remaining parens [**02**, **18**]

			01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	
1.			□	(	□	(	◊	A	→	B	)	↔	□	(	A	→	□	B	)	)	S5:58.
				L		L					R			L					R	R	
				02		04					09			12					17	18	
2.	<<	01	02		04					09			12					17	18		<<
			L		L					R			L					R	R		
3.				**L**		*L*					R			L					R	R	
				02		*04*					09			12					17	18	
	<<	01	02		04					*09*			12					17	**18**		<<
			L		L					*R*			L					R	**R**		
4.	L			**02**		*04*								12							
	R					*09*								17					**18**		
5.	L			**02**		*04*								12							
	R			**18**		*09*								17							

Meth*N* scalability

- Meth8 [2^3] uses 8-bit connectives (256) in truth tables of 32-bits (of 8-bits / row).
- Meth16 [2^4] uses 16-bit connectives (65,536) in truth tables of 1024-bits (of 64-bits / row).

  But do we need a logic with more than 65,536 connectives?

  Maybe to map the human brain using the *Method and system for Kanban cell neuron network,* U.S. Patent (allowed 25 Sep 2015).
- Meth*N* [2^(log*N*/log2)] uses *N*-bit connectives (2^*N*) in truth tables of (2^*N)/N*-bits (of (2^*N)/N/4*-bits / row).

Variants of $Ł_4$: the Option Approach

- The 2-tuple follows $Ł_4$: { 00, 10, 01, 11 }, 11 as designated.
  - For { Contradiction, False, True, Tautology } : Boolean B_4
- Variant 1: { F, C, N, T } with designated value as T
  - For { False, Contingent, Noncontingent, True }
- Variant 2: { U, I, P, E } with designated value as E
  - For { Unevaluated, Improper, Proper, Evaluated }
- Option 1, 2, 3:
  - Strategies disambiguate the middle rows of truth tables where two unary propositions fall within the scope of the same modal operator.

Variants 1, 2a, 2b

B_4	**V1**	V1 ~	V1 []	V1 ~[]	V1 <>	V1 ~<>	**V2a**	V2a ~	V2a []	V2a ~[]	V2a <>	V2a ~<>	**V2b**	V2b ~	V2b []	V2b ~[]	V2b <>	V2b ~<>	**B_4**
11	T	F	N	C	T	F	E	U	P	I	E	U	E	U	I	P	P	I	11
01	N	C	N	C	T	F	P	I	P	I	E	U	P	I	**U**	**E**	**E**	**U**	01
10	C	N	F	T	C	N	I	P	U	E	I	P	I	P	**I**	**P**	**P**	**I**	10
00	F	T	F	T	C	N	U	E	U	E	I	P	U	E	U	E	E	U	00

The interpretation of truth tables is non standard: truth possibilities are (T, N, C) and (E, P, I); but proof is (T) and (E).

Truth possibilities are bivalent in V2a as 'is the case' or in V2b as 'is not the case'. Use of 'the case' disambiguates truth possibility from truth evaluations. Truth values are truth evaluations.

Middle rows in V2b 'not the case' contain a mix of the same values, making it unclear which matrix (V2a or V2b) to apply to the same modal operator. Hence these rows are not applicable.

Options 1, 2, 3 as strategies

	U	I	P	E
Option 1 □	U	I	P	E
Option 2 □	U	U	U	U
Option 3 □	U, U	U, I	P, U	P, I
Option x □	U	U, I	P, U	U, P, I, E
Option x′ □	U	U, I	P, U	U, E

Option 1 ◊	U	I	P	E
Option 2 ◊	E	E	E	E
Option 3 ◊	I, P	I, E	E, P	E, E
Option x ◊	U, I, P ,E	I, E	E, P	E
Option x′ ◊	U, E	I, E	E, P	E

Options 1, 2, and 3 cover permutations of modal operators.

Option x summarizes them; Option x' removes redundancy in x.

An argument is disproved or proved by testing only values closest to the minimal and maximal extrema.

Metadata of all permutations in ten models for (◊A & ◊B) → ◊(A & B)

(◊A	&	◊B)	→	◊	(A	&	B)
A_1–V1	(A_1–V1)*(B_1–V1)	B_1–V1		(A_1*B_1) –V1	A_1	A_1*B_1	B_1
A_2–V2a	(A_2–V2a)*(B_2–V2a)	B_2–V2a		(A_2*B_2)–Opx′	A_2	A_2*B_2	B_2
A_2–V2a	(A_2–V2a)*(B_2–V2b)	B_2–V2b		(A_2*B_2)–Opx′	A_2	A_2*B_2	B_2
A_2–V2b	(A_2–V2b)*(B_2–V2a)	B_2–V2a		(A_2*B_2)–Opx′	A_2	A_2*B_2	B_2
A_2–V2b	(A_2–V2b)*(B_2–V2b)	B_2–V2b		(A_2*B_2)–Opx′	A_2	A_2*B_2	B_2
A_2–V2a	(A_2–V2a)*(BB_2–Opx′)	BB_2–Opx′		(A_2*B_2)–Opx′	A_2	A_2*BB_2	BB_2
A_2–V2b	(A_2–V2b)*(BB_2–Opx′)	BB_2–Opx′		(A_2*B_2)–Opx′	A_2	A_2*BB_2	BB_2
AA_2–Opx′	(AA_2–Opx′)*(B_2–V2a)	B_2–V2a		(AA_2*B_2)–Opx′	AA_2	AA_2*B_2	B_2
AA_2–Opx′	(AA_2–Opx′)*(B_2–V2b)	B_2–V2b		(AA_2*B_2)–Opx′	AA_2	AA_2*B_2	B_2
AA_2–Opx′	(AA_2–Opx′)*(BB_2–Opx′)	BB_2–Opx′		(AA_2*BB_2)–Opx′	AA_2	AA_2*BB_2	BB_2

There is 1 instance of V1 and 9 instances of V2 for 10 models.

This becomes its own proof method, where each row is treated as its own model.

Worked example for □A ⊭(◊B → □B),

where 3a may be true when 3d is false

	a	b	c	d	e	
	□A	→	(◊B	→	□B)	Model
1	A*M1 A*N F, N	{F, N}→N Yes	B+M1 B+C	(B+C) → (B*N) ¬(B+C) + (B*N) ¬[(B+C)*(¬B+C)] ¬[(B*C)+(¬B*C)+C] ¬[C*(B+¬B)+C] ¬[(C*11)+C] ¬[C+C] ¬C N	B*M1 B*N	□A, M1 ◊B, M1 □B, M1
2	A*M2a A*01 00, 01	Yes	B+M2c B+00 B	B → B 11	B*M2c B*11 B	□A,M2a ◊B, M2c □B, M2c
3	A*M2a A*01 00, **01** ******	{00, **01**}→**00** No	B+M2d B+11 **11**	11 ⊂ 00 **00** ********	B*M2d B*00 00	□A, M2a ◊B, M2d □B, M2d
4	A*M2c A*11 A	A→01 No	B+M2a B+10	(B+10)→(B*01) ¬(B+10)+(B*01) ¬[(B+10)*(¬B+10)] ¬[(B*10)+(¬B*10)+10)] ¬[10*(B+¬B)+10)] ¬[(10*11)+10)] ¬[10+10] 01	B*M2a B*01	□A, M2c ◊B, M2a □B, M2a
5	A*M2c A*11 A	A → 11 yes	B+M2c B+00 B	B → B 11	B*M2c B*11 B	□A,M2c ◊B, M2c □B, M2c
6	A*M2d A*00 00	00 → 01 yes	B+M2a B+10	(B+10) → (B*01) ¬(B+10) + (B*01) ¬[(B*10)*(¬B+10)] ¬[(B*10)+(¬B*10)+10] ¬[10*(B+¬B)+10] ¬[10*(11)+10 ¬[10+10] ¬10 01	B*M2a B*01	□A, M2d ◊B, M2a □B, M2a
7	A*M2d A*00 00	00→00 yes	B+M2d B+11 11	11 → 00 00	B*M2d B*00 00	□A, M2d ◊B, M2d □B, M2d

Test results

- Not general theorems, and with semantic consequences.
- Not general theorems, but with*out* semantic consequences.

Not general theorems, and with semantic consequence

- K axiom $(A \to B) \to (\Box A \to \Box B)$

  08 $\neg\Diamond(A \,\&\, B) \leftrightarrow \Box(A \to \neg B)$

  15 $\Box(A \vee B) \to (\Box A \vee \Diamond B)$

- D axiom

  33 $(\Diamond\neg A \vee \Diamond\neg B) \vee \Diamond(A \,\&\, B)$

- B axiom

  40 $\neg\Diamond(\Diamond\Box\Diamond A \,\&\, \Box\neg A)$

- S5 axiom

  57 $\Box(\Box A \to \Box B) \vee \Box(\Box B \to \Box A)$

Not general theorems, but with*out* semantic consequence

- K axiom

  19 $(\Box A \,\&\, \Diamond B) \rightarrow \Diamond(A \,\&\, B)$

  21 $(\Box A \rightarrow \Diamond B) \rightarrow \Diamond(A \rightarrow B)$

- S4 axiom

  42 $(\Diamond A \,\&\, \Box B) \rightarrow \Diamond(A \,\&\, \Box B)$

- S5 axiom

  54 $(\Box A \vee \Diamond B) \rightarrow \Box(A \vee \Diamond B)$

  55 $(\Diamond A \,\&\, \Diamond B) \rightarrow \Diamond(A \,\&\, \Diamond B)$

  56 $(\Diamond A \,\&\, \Box B) \rightarrow \Diamond(A \,\&\, \Box B)$, also $(A \,\&\, \Box B) \rightarrow (\Diamond A \,\&\, \Box B)$

Point of contact:

Colin James III, Director
Ersatz Systems Machine Cognition, LLC
Colorado Springs, CO 80904-5220 USA

Email: info@cec-services.com
Mobile: +011 (719) 210 – 9534

Websites: ersatz-systems.com; cec-services.com