

Astrologic; Second-Order Logic and the k -order Ladder; Second-Order Axiomatized Arithmetic; Gödel's “God Theorem” & Speedup Theorem

Selmer Bringsjord

Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Lally School of Management
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

Intro to Logic
3/26/2020



Astrologic; Second-Order Logic and the k -order Ladder; Second-Order Axiomatized Arithmetic; Gödel's “God Theorem” & Speedup Theorem

Selmer Bringsjord

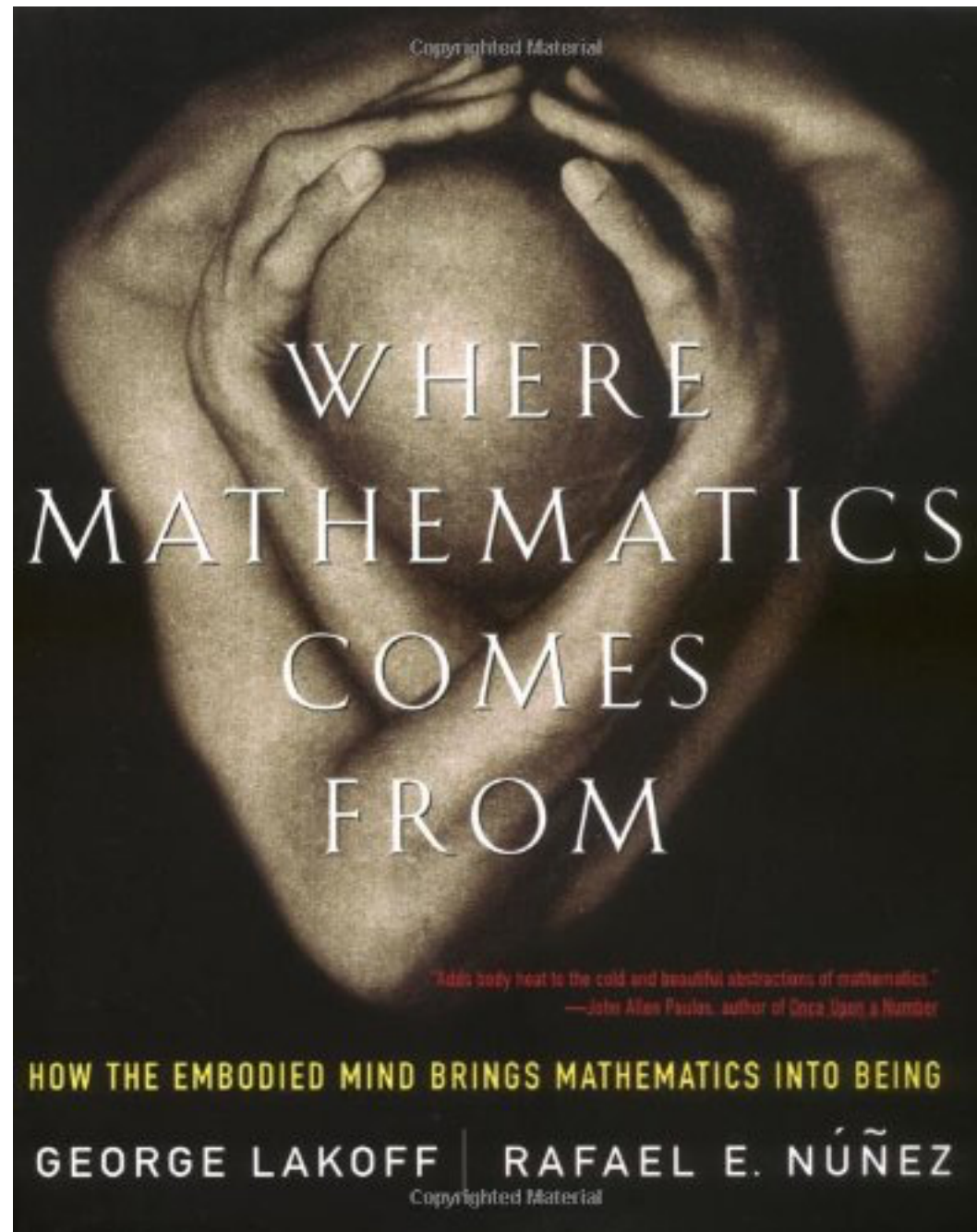
Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Lally School of Management
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

Intro to Logic
3/26/2020

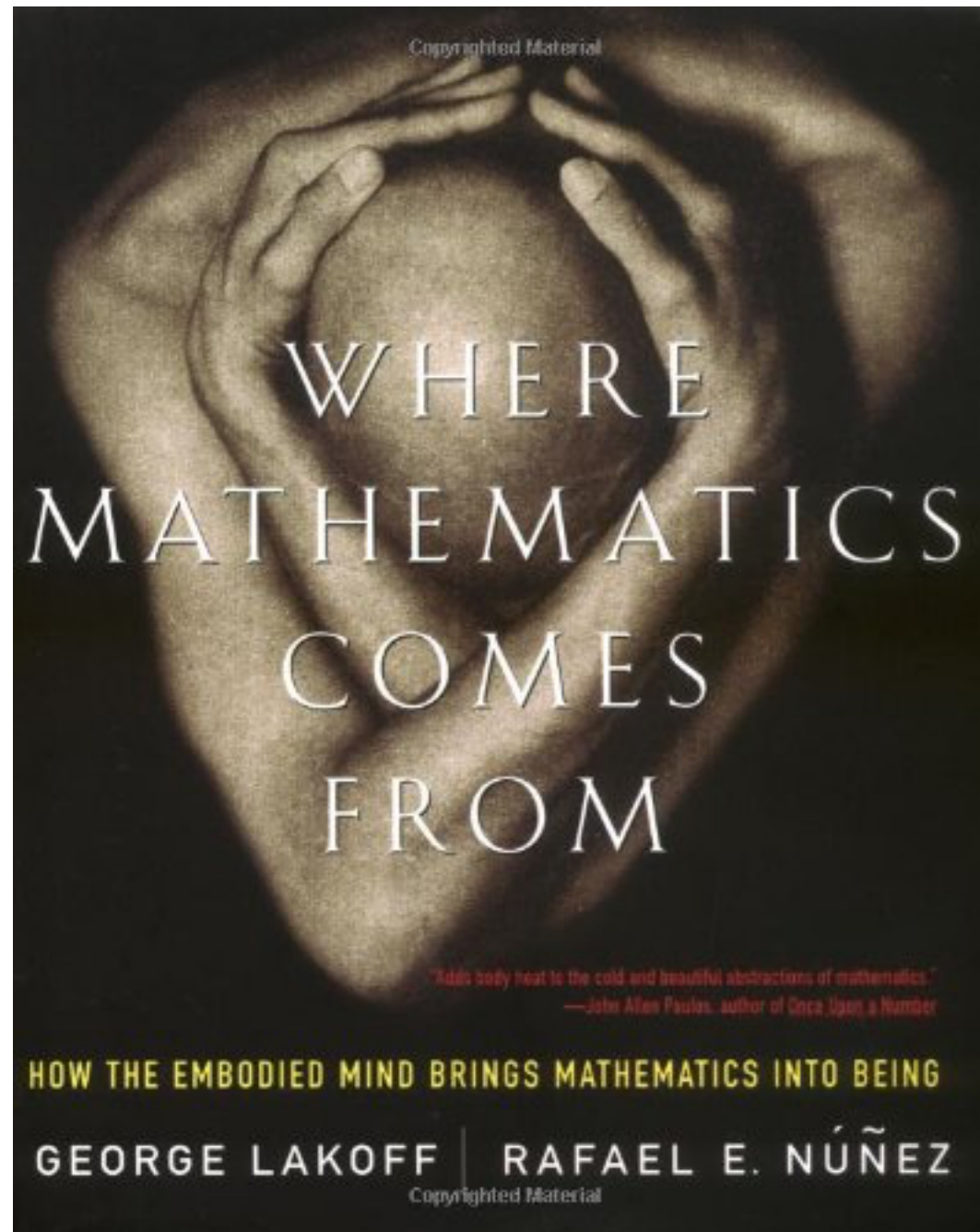


Do we just manufacture mathematics?

Do we just manufacture mathematics?



Do we just manufacture mathematics?



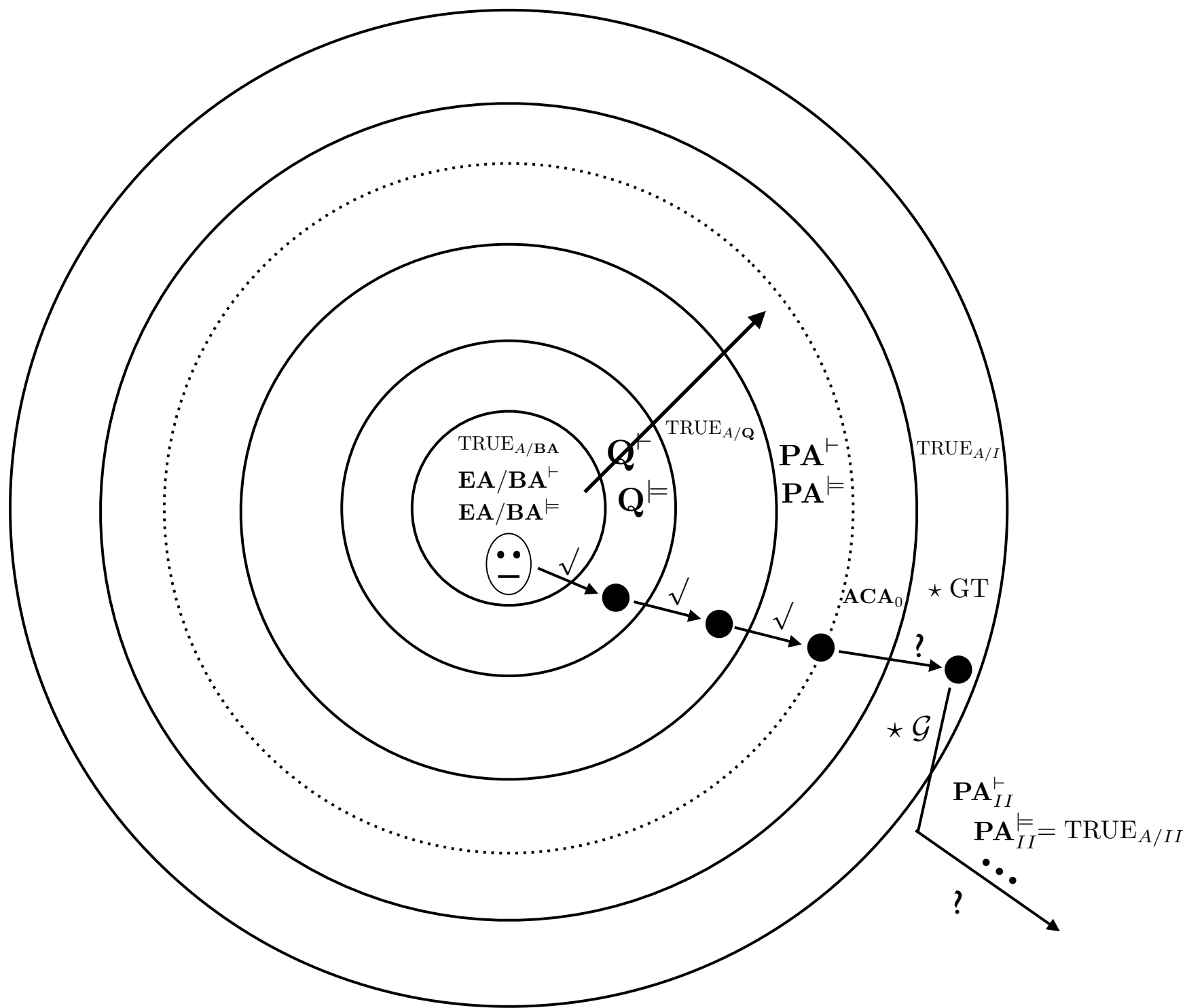
No! We tap into deep, underlying reality — and aliens do/would too ...

Do we just manufacture mathematics?

No! We tap into deep, underlying reality — and aliens do/would too ...

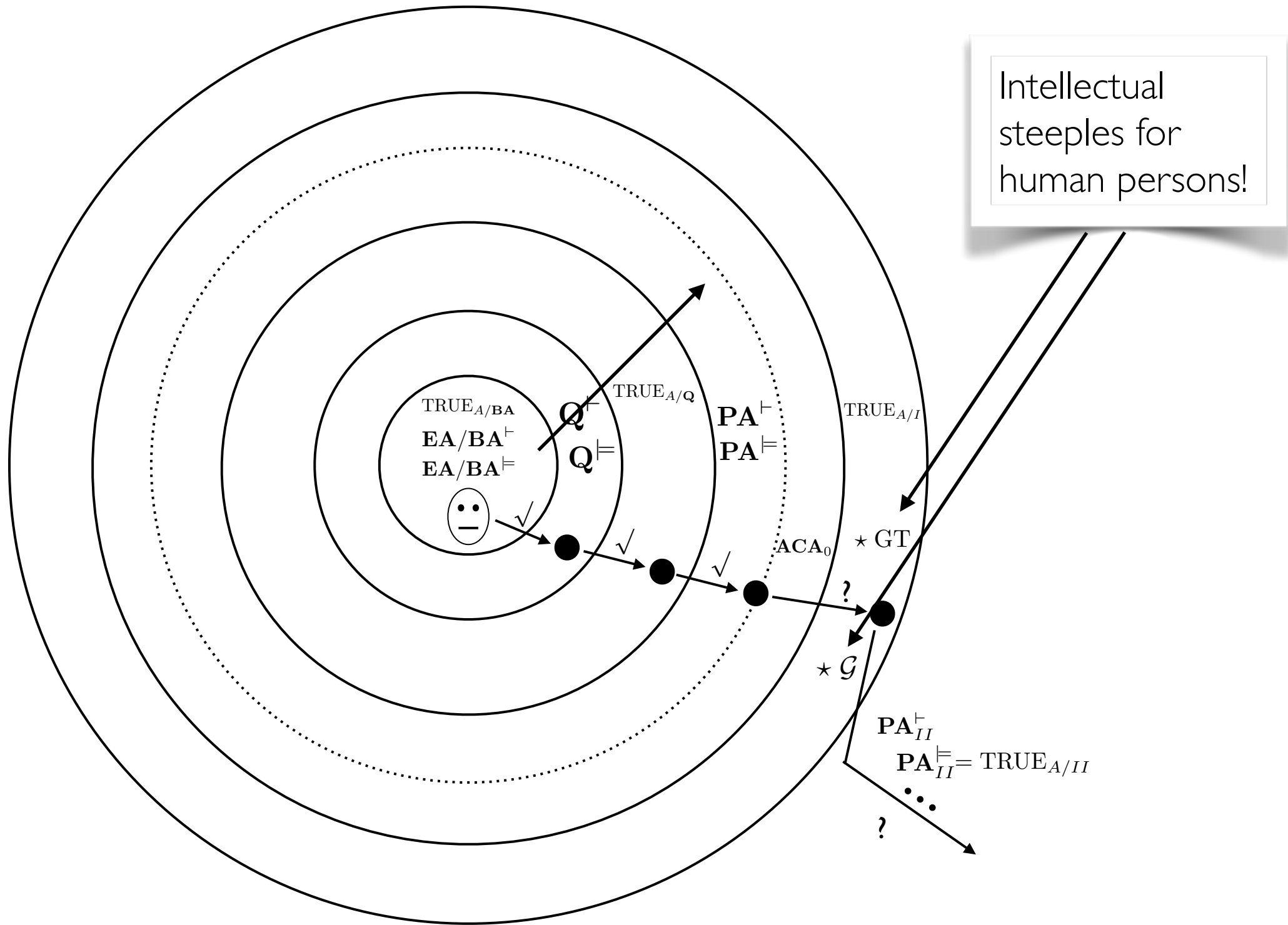
Astrologic

(Aliens & Angels on the Same “Race Track”)



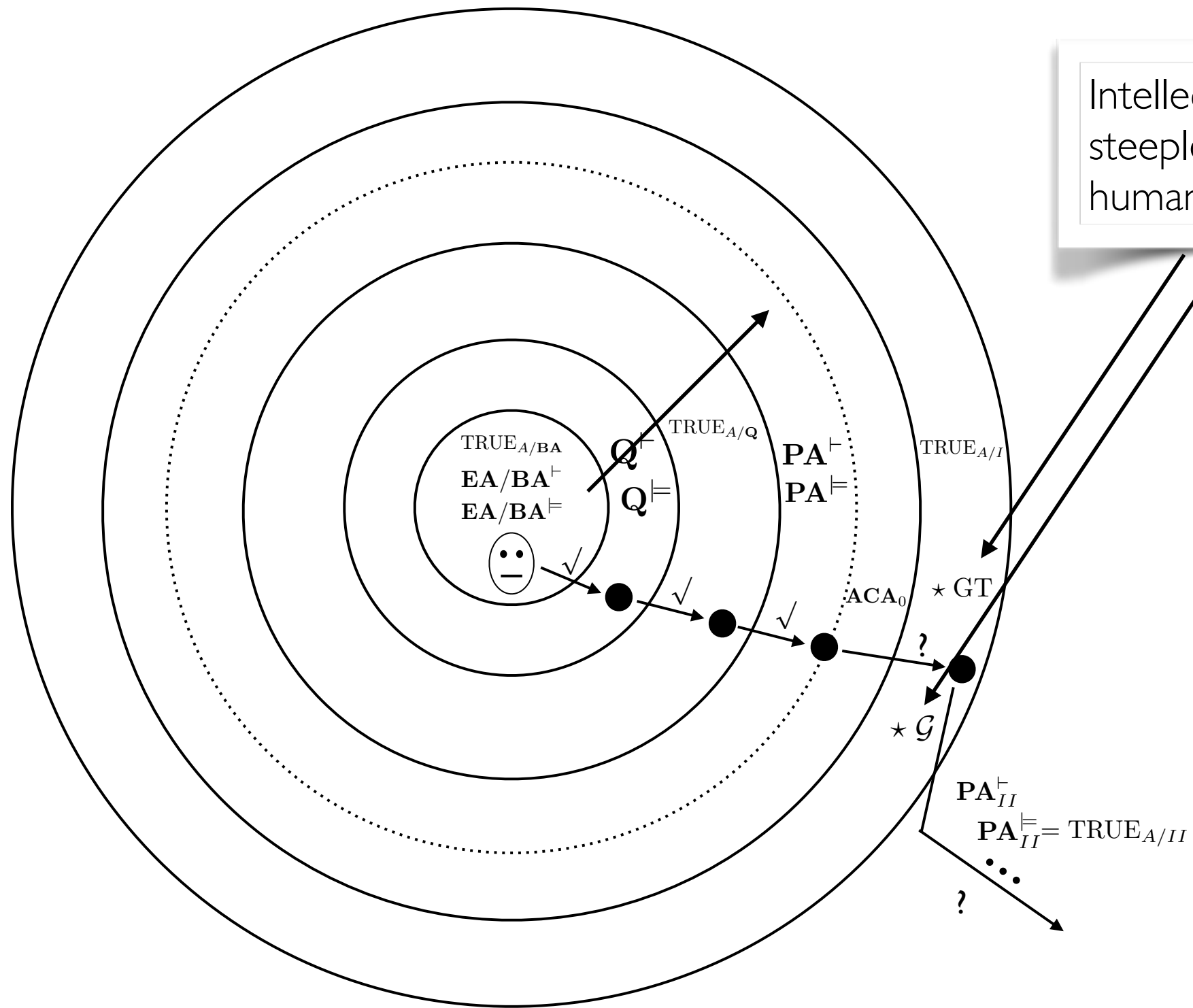
Astrologic

(Aliens & Angels on the Same “Race Track”)



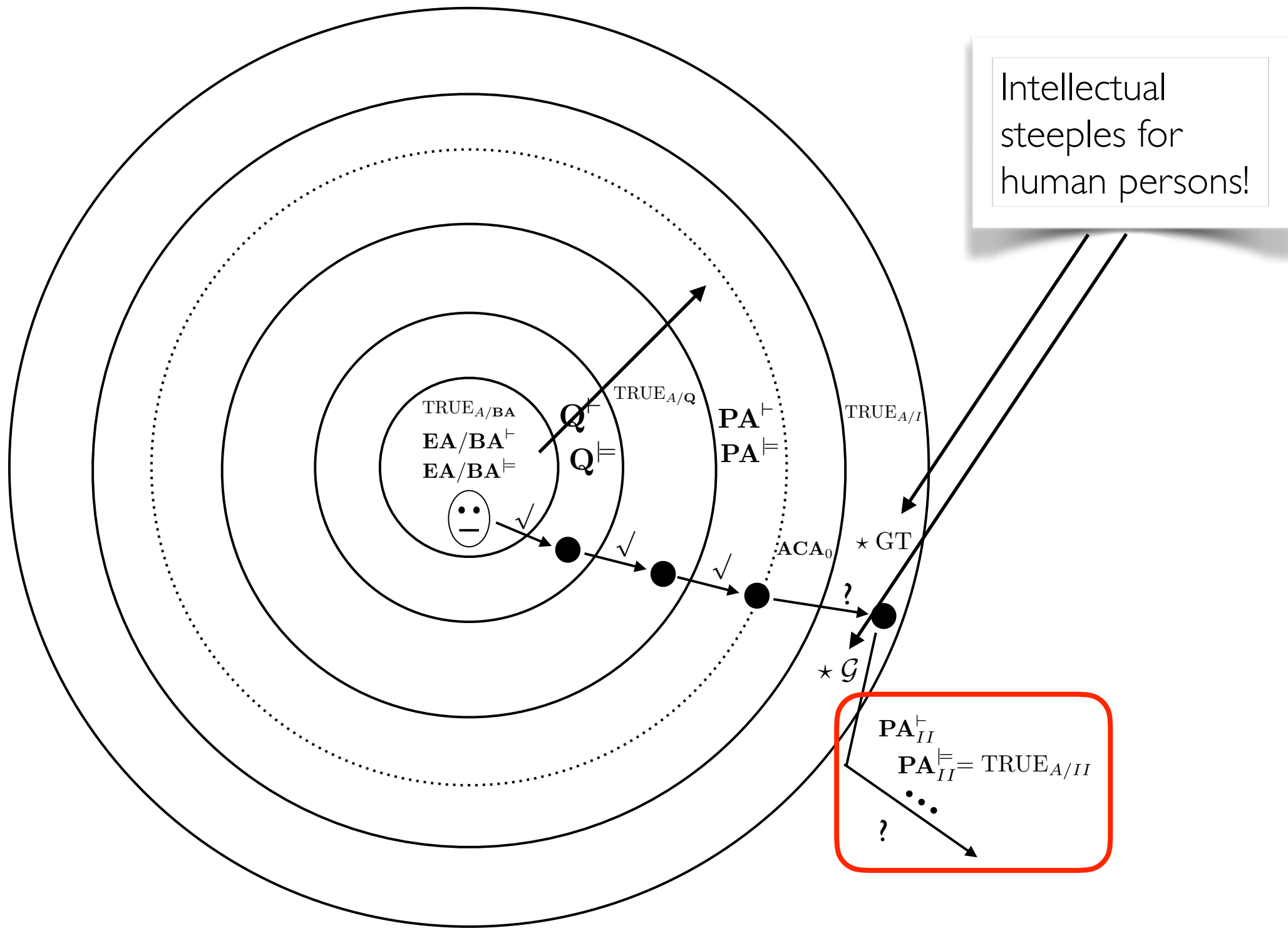
Astrologic

(Aliens & Angels on the Same “Race Track”)



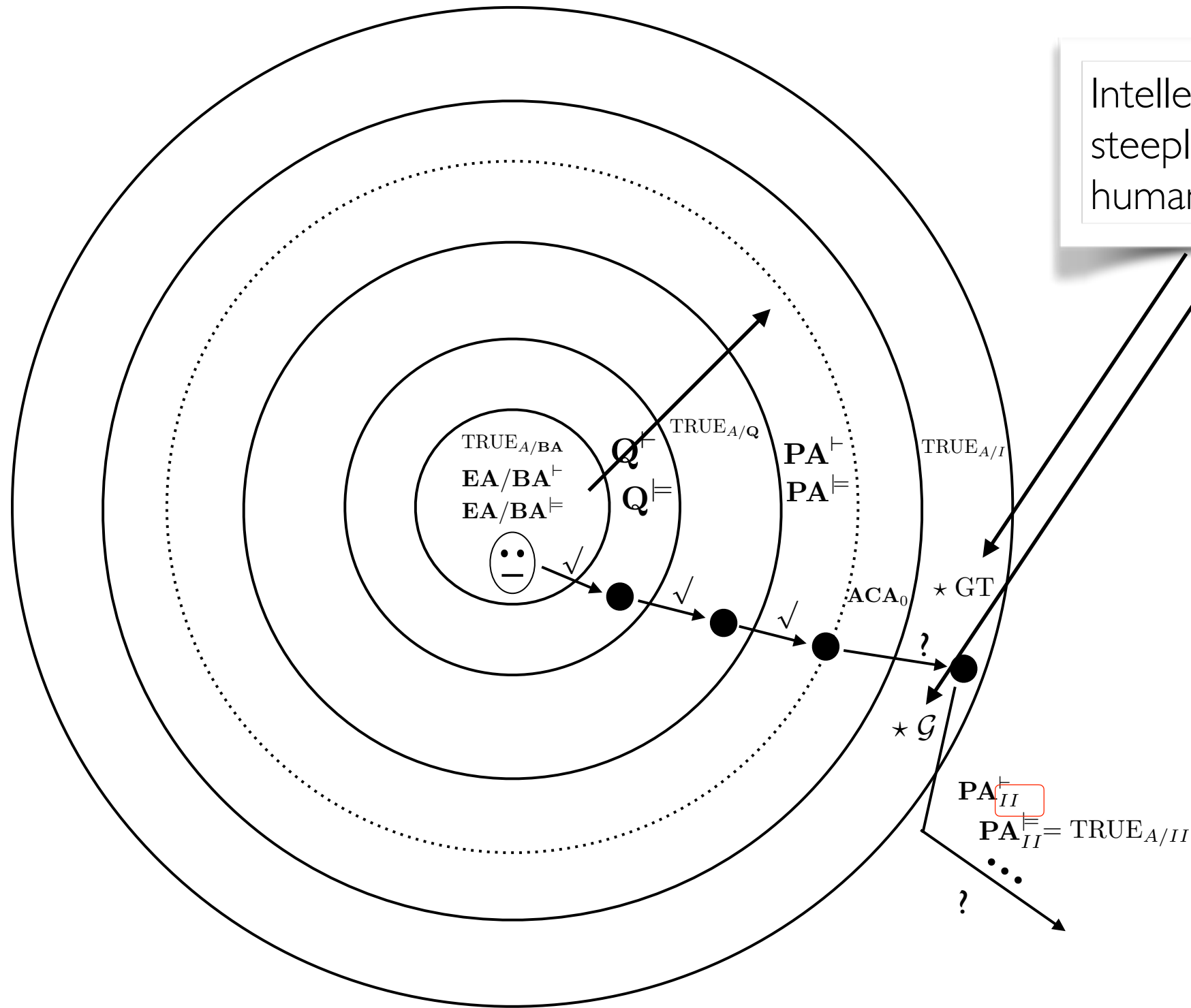
Astrologic

(Aliens & Angels on the Same “Race Track”)



Astrologic

(Aliens & Angels on the Same “Race Track”)



Intellectual
steeples for
human persons!





Q (= Robinson Arithmetic)

$$\text{A1} \quad \forall x(0 \neq s(x))$$

$$\text{A2} \quad \forall x \forall y (s(x) = s(y) \rightarrow x = y)$$

$$\text{A3} \quad \forall x (x \neq 0 \rightarrow \exists y (x = s(y)))$$

$$\text{A4} \quad \forall x (x + 0 = x)$$

$$\text{A5} \quad \forall x \forall y (x + s(y) = s(x + y))$$

$$\text{A6} \quad \forall x (x \times 0 = 0)$$

$$\text{A7} \quad \forall x \forall y (x \times s(y) = (x \times y) + x)$$



Q (= Robinson Arithmetic)

$$\text{A1} \quad \forall x(0 \neq s(x))$$

$$\text{A2} \quad \forall x \forall y (s(x) = s(y) \rightarrow x = y)$$

$$\text{A3} \quad \forall x (x \neq 0 \rightarrow \exists y (x = s(y)))$$

$$\text{A4} \quad \forall x (x + 0 = x)$$

$$\text{A5} \quad \forall x \forall y (x + s(y) = s(x + y))$$

$$\text{A6} \quad \forall x (x \times 0 = 0)$$

$$\text{A7} \quad \forall x \forall y (x \times s(y) = (x \times y) + x)$$

PA (Peano Arithmetic)

$$\text{A1} \quad \forall x(0 \neq s(x))$$

$$\text{A2} \quad \forall x \forall y (s(x) = s(y) \rightarrow x = y)$$

$$\text{A3} \quad \forall x (x \neq 0 \rightarrow \exists y (x = s(y)))$$

$$\text{A4} \quad \forall x (x + 0 = x)$$

$$\text{A5} \quad \forall x \forall y (x + s(y) = s(x + y))$$

$$\text{A6} \quad \forall x (x \times 0 = 0)$$

$$\text{A7} \quad \forall x \forall y (x \times s(y) = (x \times y) + x)$$

And, every sentence that is the universal closure of an instance of

$$([\phi(0) \wedge \forall x(\phi(x) \rightarrow \phi(s(x)))] \rightarrow \forall x \phi(x))$$

where $\phi(x)$ is open wff with variable x , and perhaps others, free.

PA (Peano Arithmetic)

$$A1 \quad \forall x(0 \neq s(x))$$

$$A2 \quad \forall x \forall y (s(x) = s(y) \rightarrow x = y)$$

$$A3 \quad \forall x (x \neq 0 \rightarrow \exists y (x = s(y)))$$

$$A4 \quad \forall x (x + 0 = x)$$

$$A5 \quad \forall x \forall y (x + s(y) = s(x + y))$$

$$A6 \quad \forall x (x \times 0 = 0)$$

$$A7 \quad \forall x \forall y (x \times s(y) = (x \times y) + x)$$

And, every sentence that is the universal closure of an instance of

$$([\phi(0) \wedge \forall x(\phi(x) \rightarrow \phi(s(x)))] \rightarrow \forall x \phi(x)) \quad \infty$$

where $\phi(x)$ is open wff with variable x , and perhaps others, free.

PA (Peano Arithmetic)

$$A1 \quad \forall x(0 \neq s(x))$$

$$A2 \quad \forall x \forall y (s(x) = s(y) \rightarrow x = y)$$

$$A3 \quad \forall x (x \neq 0 \rightarrow \exists y (x = s(y)))$$

$$A4 \quad \forall x (x + 0 = x)$$

$$A5 \quad \forall x \forall y (x + s(y) = s(x + y))$$

$$A6 \quad \forall x (x \times 0 = 0)$$

$$A7 \quad \forall x \forall y (x \times s(y) = (x \times y) + x)$$

And, every sentence that is the universal closure of an instance of

$$([\phi(0) \wedge \forall x(\phi(x) \rightarrow \phi(s(x)))] \rightarrow \forall x \phi(x)) \quad \infty$$

where $\phi(x)$ is open wff with variable x , and perhaps others, free.


Astrologic; Second-Order Logic and the k -order Ladder; Second-Order Axiomatized Arithmetic; Gödel's “God Theorem” & Speedup Theorem

Selmer Bringsjord

Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Lally School of Management
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

Intro to Logic
3/26/2020





Astrologic; Second-Order Logic and the k -order Ladder; Second-Order Axiomatized Arithmetic; Gödel's “God Theorem” & Speedup Theorem

Selmer Bringsjord

Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Lally School of Management
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

Intro to Logic
3/26/2020



FOL

✓ FOL

✓ FOL

Epistemic + FOL

$B_d B_v B_d V v$

✓ FOL

Epistemic + FOL (for coverage of “killer” robots, later)
 $B_d B_v B_d V v$

✓ FOL

✓ Epistemic + FOL (for coverage of “killer” robots, later)
 $B_d B_v B_d V v$

✓ FOL

✓ Epistemic + FOL (for coverage of “killer” robots, later)
 $B_d B_v B_d V v$

TOL

$\exists X[X(j) \wedge \neg X(m) \wedge S(X)]$

✓ FOL

✓ Epistemic + FOL (for coverage of “killer” robots, later)
 $B_d B_v B_d V v$

TOL

$\exists X[X(j) \wedge \neg X(m) \wedge S(X)]$



✓ FOL

✓ Epistemic + FOL (for coverage of “killer” robots, later)
 $B_d B_v B_d V v$

TOL

? $\exists X[X(j) \wedge \neg X(m) \wedge S(X)]$



Double-Minded Man

The Contemporary Craft of Creating Characters Meets Today's Cognitive Architectures: A Case Study in Expressivity*

Selmer Bringsjord • John Licato • Alexander Bringsjord

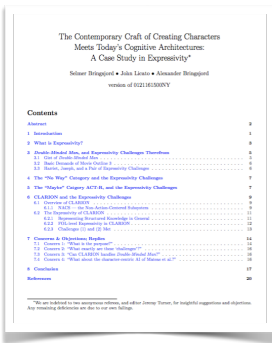
version of 0121161500NY

Contents

Abstract	2
1 Introduction	1
2 What is Expressivity?	3
3 <i>Double-Minded Man</i> , and Expressivity Challenges Therefrom	5
3.1 Gist of <i>Double-Minded Man</i>	5
3.2 Basic Demands of Movie Outline 3	6
3.3 Harriet, Joseph, and a Pair of Expressivity Challenges	6
4 The “No Way” Category and the Expressivity Challenges	7
5 The “Maybe” Category ACT-R, and the Expressivity Challenges	7
6 CLARION and the Expressivity Challenges	9
6.1 Overview of CLARION	9
6.1.1 NACS — the Non-Action-Centered Subsystem	9
6.2 The Expressivity of CLARION	11
6.2.1 Representing Structured Knowledge in General	11
6.2.2 FOL-level Expressivity in CLARION	12
6.2.3 Challenges (1) and (2) Met	13
7 Concerns & Objections; Replies	14
7.1 Concern 1: “What is the purpose?”	14
7.2 Concern 2: “What exactly are these ‘challenges’?”	16
7.3 Concern 3: “Can CLARION handles <i>Double-Minded Man</i> ?”	16
7.4 Concern 4: “What about the character-centric AI of Mateas et al.?”	16
8 Conclusion	17
References	20

*We are indebted to two anonymous referees, and editor Jeremy Turner, for insightful suggestions and objections. Any remaining deficiencies are due to our own failings.

Double-Minded Man



Movie Outline - Double-Minded_Man_010316.mvo

Arial 12 Scene Heading 100%

Outline Script Notes Characters FeelFactor Reference Library PowerView Step Cards Story Tasks

1. TWIRL - DAY

68-year-old Harriet Smith sits with two wrinkled hands firmly on the wheel of her rust-eaten Subaru wagon, staring straight ahead through the top level of bifocals as she waits serenely at a red light.

Harriet is alone in the car. To her right is another vehicle, also waiting, in this case to make a right turn; it's a sleek, low-slung, black Camaro.

We are inside the cabin with Harriet. The Subaru's sound system softly plays choral music. Harriet's lips move slightly as she internally sings along, mouthing a slow aria. Her head weaves slightly side to side, in the rhythm with the music.

Things are calm as can be here inside the car with Harriet. There are a pair of well-worn Bibles on the empty passenger seat beside her, one with a gold-lettered 'Harriet' on its leather front cover, the other with a matching 'Joseph' on its front cover.

Harriet's eyes swivel up to the light: still red. We wait with her.

Suddenly there is a piercing SCREECH outside. Harriet jerks her head to the right and we follow her line of sight.

A sleek motorcycle has swerved out of its lane and is now streaking straight for the right side of the Camaro beside Harriet's car.

The bike slams with CLANG into the side of the Camaro. Its rider is flung up and forward into the air, twirling passed Harriet's windshield.

We now watch from Harriet's POV, in slow motion. The black-leather-clad motorcyclist sails by Harriet's windshield, airborne. We see a man's face, clearly: His elephant-hide skin tells us that he is well beyond middle-age. Yet thick, black curls of youthful hair emerge from under his helmet. The rider has only one half of a black, bushy, swept-out, waxed mustache. His eyes are weary and grey, and appear to lock with Harriet's for an instant.

We return to normal speed. The body is now lying on the incoming lane to the left of Harriet's Subaru, perfectly still on the blacktop, the head twisted into an impossible angle. Blood seeps from a nostril. Beside the lifeless head, a BMW medallion lies on the pavement, glinting in the sunlight.

1. TWIRL - DAY

Step 1 of 3

1. TWIRL - DAY
2. YES, THAT'S HIM - LATER
3. SECOND HOME - LATER

The Contemporary Craft of Creating Characters
Meets Today's Cognitive Architecture:
A Case Study in Expressivity*

Selmer Bringsjord • John Lewis • Alexander Bringsjord
version of 022161000V

Contents

1 Introduction	1
2 What is Expressivity?	2
3 Double-Minded Men, and Expressivity Challenges: Overview	3
4 The "What-If" Scenario: A Case Study in Expressivity Challenges	4
5 The "What-If" Scenario: A Case Study in Expressivity Challenges	5
6 The "What-If" Scenario: A Case Study in Expressivity Challenges	6
7 The "What-If" Scenario: A Case Study in Expressivity Challenges	7
8 The "What-If" Scenario: A Case Study in Expressivity Challenges	8
9 The "What-If" Scenario: A Case Study in Expressivity Challenges	9
10 The "What-If" Scenario: A Case Study in Expressivity Challenges	10
11 The "What-If" Scenario: A Case Study in Expressivity Challenges	11
12 The "What-If" Scenario: A Case Study in Expressivity Challenges	12
13 The "What-If" Scenario: A Case Study in Expressivity Challenges	13
14 The "What-If" Scenario: A Case Study in Expressivity Challenges	14
15 The "What-If" Scenario: A Case Study in Expressivity Challenges	15
16 The "What-If" Scenario: A Case Study in Expressivity Challenges	16
17 The "What-If" Scenario: A Case Study in Expressivity Challenges	17
18 The "What-If" Scenario: A Case Study in Expressivity Challenges	18
19 The "What-If" Scenario: A Case Study in Expressivity Challenges	19
20 The "What-If" Scenario: A Case Study in Expressivity Challenges	20
21 The "What-If" Scenario: A Case Study in Expressivity Challenges	21
22 The "What-If" Scenario: A Case Study in Expressivity Challenges	22
23 The "What-If" Scenario: A Case Study in Expressivity Challenges	23
24 The "What-If" Scenario: A Case Study in Expressivity Challenges	24
25 The "What-If" Scenario: A Case Study in Expressivity Challenges	25
26 The "What-If" Scenario: A Case Study in Expressivity Challenges	26
27 The "What-If" Scenario: A Case Study in Expressivity Challenges	27
28 The "What-If" Scenario: A Case Study in Expressivity Challenges	28
29 The "What-If" Scenario: A Case Study in Expressivity Challenges	29
30 The "What-If" Scenario: A Case Study in Expressivity Challenges	30
31 The "What-If" Scenario: A Case Study in Expressivity Challenges	31
32 The "What-If" Scenario: A Case Study in Expressivity Challenges	32
33 The "What-If" Scenario: A Case Study in Expressivity Challenges	33
34 The "What-If" Scenario: A Case Study in Expressivity Challenges	34
35 The "What-If" Scenario: A Case Study in Expressivity Challenges	35
36 The "What-If" Scenario: A Case Study in Expressivity Challenges	36
37 The "What-If" Scenario: A Case Study in Expressivity Challenges	37
38 The "What-If" Scenario: A Case Study in Expressivity Challenges	38
39 The "What-If" Scenario: A Case Study in Expressivity Challenges	39
40 The "What-If" Scenario: A Case Study in Expressivity Challenges	40
41 The "What-If" Scenario: A Case Study in Expressivity Challenges	41
42 The "What-If" Scenario: A Case Study in Expressivity Challenges	42
43 The "What-If" Scenario: A Case Study in Expressivity Challenges	43
44 The "What-If" Scenario: A Case Study in Expressivity Challenges	44
45 The "What-If" Scenario: A Case Study in Expressivity Challenges	45
46 The "What-If" Scenario: A Case Study in Expressivity Challenges	46
47 The "What-If" Scenario: A Case Study in Expressivity Challenges	47
48 The "What-If" Scenario: A Case Study in Expressivity Challenges	48
49 The "What-If" Scenario: A Case Study in Expressivity Challenges	49
50 The "What-If" Scenario: A Case Study in Expressivity Challenges	50

*We are indebted to the anonymous referees, and other Jorgis Bringsjord, for insightful suggestions and comments.
Any remaining shortcomings are due to our own failure.

Double-Minded Man

Double-Minded Man
by
S Bringsjord & A Bringsjord

DRAFT #5
© June 30 2016

Selmer.Bringsjord@gmail.com

Contents

1. Introduction	1
2. What is Expressivity?	2
3. Double-Minded Man, and Expressivity Challenges: Overview	3
4. The "What?" Chapter (02-04), and the Expressivity Challenges	4
5. The "How?" Chapter (02-05), and the Expressivity Challenges	5
6. The "Why?" Chapter (02-06), and the Expressivity Challenges	6
7. The "What?" Chapter (02-07), and the Expressivity Challenges	7
8. The "How?" Chapter (02-08), and the Expressivity Challenges	8
9. The "Why?" Chapter (02-09), and the Expressivity Challenges	9
10. The "What?" Chapter (02-10), and the Expressivity Challenges	10
11. The "How?" Chapter (02-11), and the Expressivity Challenges	11
12. The "Why?" Chapter (02-12), and the Expressivity Challenges	12
13. The "What?" Chapter (02-13), and the Expressivity Challenges	13
14. The "How?" Chapter (02-14), and the Expressivity Challenges	14
15. The "Why?" Chapter (02-15), and the Expressivity Challenges	15
16. The "What?" Chapter (02-16), and the Expressivity Challenges	16
17. The "How?" Chapter (02-17), and the Expressivity Challenges	17
18. The "Why?" Chapter (02-18), and the Expressivity Challenges	18
19. The "What?" Chapter (02-19), and the Expressivity Challenges	19
20. The "How?" Chapter (02-20), and the Expressivity Challenges	20
21. The "Why?" Chapter (02-21), and the Expressivity Challenges	21
22. The "What?" Chapter (02-22), and the Expressivity Challenges	22
23. The "How?" Chapter (02-23), and the Expressivity Challenges	23
24. The "Why?" Chapter (02-24), and the Expressivity Challenges	24
25. The "What?" Chapter (02-25), and the Expressivity Challenges	25
26. The "How?" Chapter (02-26), and the Expressivity Challenges	26
27. The "Why?" Chapter (02-27), and the Expressivity Challenges	27
28. The "What?" Chapter (02-28), and the Expressivity Challenges	28
29. The "How?" Chapter (02-29), and the Expressivity Challenges	29
30. The "Why?" Chapter (02-30), and the Expressivity Challenges	30
31. The "What?" Chapter (02-31), and the Expressivity Challenges	31
32. The "How?" Chapter (02-32), and the Expressivity Challenges	32
33. The "Why?" Chapter (02-33), and the Expressivity Challenges	33
34. The "What?" Chapter (02-34), and the Expressivity Challenges	34
35. The "How?" Chapter (02-35), and the Expressivity Challenges	35
36. The "Why?" Chapter (02-36), and the Expressivity Challenges	36
37. The "What?" Chapter (02-37), and the Expressivity Challenges	37
38. The "How?" Chapter (02-38), and the Expressivity Challenges	38
39. The "Why?" Chapter (02-39), and the Expressivity Challenges	39
40. The "What?" Chapter (02-40), and the Expressivity Challenges	40
41. The "How?" Chapter (02-41), and the Expressivity Challenges	41
42. The "Why?" Chapter (02-42), and the Expressivity Challenges	42
43. The "What?" Chapter (02-43), and the Expressivity Challenges	43
44. The "How?" Chapter (02-44), and the Expressivity Challenges	44
45. The "Why?" Chapter (02-45), and the Expressivity Challenges	45
46. The "What?" Chapter (02-46), and the Expressivity Challenges	46
47. The "How?" Chapter (02-47), and the Expressivity Challenges	47
48. The "Why?" Chapter (02-48), and the Expressivity Challenges	48
49. The "What?" Chapter (02-49), and the Expressivity Challenges	49
50. The "How?" Chapter (02-50), and the Expressivity Challenges	50
51. The "Why?" Chapter (02-51), and the Expressivity Challenges	51
52. The "What?" Chapter (02-52), and the Expressivity Challenges	52
53. The "How?" Chapter (02-53), and the Expressivity Challenges	53
54. The "Why?" Chapter (02-54), and the Expressivity Challenges	54
55. The "What?" Chapter (02-55), and the Expressivity Challenges	55
56. The "How?" Chapter (02-56), and the Expressivity Challenges	56
57. The "Why?" Chapter (02-57), and the Expressivity Challenges	57
58. The "What?" Chapter (02-58), and the Expressivity Challenges	58
59. The "How?" Chapter (02-59), and the Expressivity Challenges	59
60. The "Why?" Chapter (02-60), and the Expressivity Challenges	60
61. The "What?" Chapter (02-61), and the Expressivity Challenges	61
62. The "How?" Chapter (02-62), and the Expressivity Challenges	62
63. The "Why?" Chapter (02-63), and the Expressivity Challenges	63
64. The "What?" Chapter (02-64), and the Expressivity Challenges	64
65. The "How?" Chapter (02-65), and the Expressivity Challenges	65
66. The "Why?" Chapter (02-66), and the Expressivity Challenges	66
67. The "What?" Chapter (02-67), and the Expressivity Challenges	67
68. The "How?" Chapter (02-68), and the Expressivity Challenges	68
69. The "Why?" Chapter (02-69), and the Expressivity Challenges	69
70. The "What?" Chapter (02-70), and the Expressivity Challenges	70
71. The "How?" Chapter (02-71), and the Expressivity Challenges	71
72. The "Why?" Chapter (02-72), and the Expressivity Challenges	72
73. The "What?" Chapter (02-73), and the Expressivity Challenges	73
74. The "How?" Chapter (02-74), and the Expressivity Challenges	74
75. The "Why?" Chapter (02-75), and the Expressivity Challenges	75
76. The "What?" Chapter (02-76), and the Expressivity Challenges	76
77. The "How?" Chapter (02-77), and the Expressivity Challenges	77
78. The "Why?" Chapter (02-78), and the Expressivity Challenges	78
79. The "What?" Chapter (02-79), and the Expressivity Challenges	79
80. The "How?" Chapter (02-80), and the Expressivity Challenges	80
81. The "Why?" Chapter (02-81), and the Expressivity Challenges	81
82. The "What?" Chapter (02-82), and the Expressivity Challenges	82
83. The "How?" Chapter (02-83), and the Expressivity Challenges	83
84. The "Why?" Chapter (02-84), and the Expressivity Challenges	84
85. The "What?" Chapter (02-85), and the Expressivity Challenges	85
86. The "How?" Chapter (02-86), and the Expressivity Challenges	86
87. The "Why?" Chapter (02-87), and the Expressivity Challenges	87
88. The "What?" Chapter (02-88), and the Expressivity Challenges	88
89. The "How?" Chapter (02-89), and the Expressivity Challenges	89
90. The "Why?" Chapter (02-90), and the Expressivity Challenges	90
91. The "What?" Chapter (02-91), and the Expressivity Challenges	91
92. The "How?" Chapter (02-92), and the Expressivity Challenges	92
93. The "Why?" Chapter (02-93), and the Expressivity Challenges	93
94. The "What?" Chapter (02-94), and the Expressivity Challenges	94
95. The "How?" Chapter (02-95), and the Expressivity Challenges	95
96. The "Why?" Chapter (02-96), and the Expressivity Challenges	96
97. The "What?" Chapter (02-97), and the Expressivity Challenges	97
98. The "How?" Chapter (02-98), and the Expressivity Challenges	98
99. The "Why?" Chapter (02-99), and the Expressivity Challenges	99
100. The "What?" Chapter (02-100), and the Expressivity Challenges	100

Double-Minded Man

1. TWIRL - DAY

68-year-old Harriet Smith sits with two wrinkled hands firmly on the wheel of her rust-eaten Subaru wagon, staring straight ahead through the top level of bifocals as she waits serenely at a red light.

Harriet is alone in the car. To her right is another vehicle, also waiting, in this case to make a right turn; it's a sleek, low-slung, black Camaro.

We are inside the cabin with Harriet. The Subaru's sound system softly plays choral music. Harriet's lips move slightly as she internally sings along, mouthing a slow aria. Her head weaves slightly side to side, in the rhythm with the music.

Things are calm as can be here inside the car with Harriet. There are a pair of well-worn Bibles on the empty passenger seat beside her, one with a gold-lettered 'Harriet' on its leather front cover, the other with a matching 'Joseph' on its front cover.

Harriet's eyes swivel up to the light: still red. We wait with her.

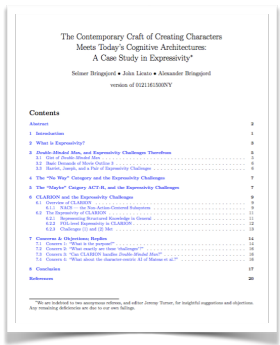
Suddenly there is a piercing SCREECH outside. Harriet jerks her head to the right and we follow her line of sight.

A sleek motorcycle has swerved out of its lane and is now streaking straight for the right side of the Camaro beside Harriet's car.

The bike slams with CLANG into the side of the Camaro. Its rider is flung up and forward into the air, twirling passed Harriet's windshield.

We now watch from Harriet's POV, in slow motion. The black-leather-clad motorcyclist sails by Harriet's windshield, airborne. We see a man's face, clearly: His elephant-hide skin tells us that he is well beyond middle-age. Yet thick, black curls of youthful hair emerge from under his helmet. The rider has only one half of a black, bushy, swept-out, waxed mustache. His eyes are weary and grey, and appear to lock with Harriet's for an instant.

We return to normal speed. The body is now lying on the incoming lane to the left of Harriet's Subaru, perfectly still on the blacktop, the head twisted into an impossible angle. Blood seeps from a nostril. Beside the lifeless head, a BMW medallion lies on the pavement, glinting in the sunlight.



Double-Minded Man

$$\exists X[X(joseph) \wedge \neg X(m(harriet, joseph)) \wedge Sleazy(X)]$$

1. TWIRL - DAY

68-year-old Harriet Smith sits with two wrinkled hands firmly on the wheel of her rust-eaten Subaru wagon, staring straight ahead through the top level of bifocals as she waits serenely at a red light.

Harriet is alone in the car. To her right is another vehicle, also waiting, in this case to make a right turn; it's a sleek, low-slung, black Camaro.

We are inside the cabin with Harriet. The Subaru's sound system softly plays choral music. Harriet's lips move slightly as she internally sings along, mouthing a slow aria. Her head weaves slightly side to side, in the rhythm with the music.

Things are calm as can be here inside the car with Harriet. There are a pair of well-worn Bibles on the empty passenger seat beside her, one with a gold-lettered 'Harriet' on its leather front cover, the other with a matching 'Joseph' on its front cover.

Harriet's eyes swivel up to the light: still red. We wait with her.

Suddenly there is a piercing SCREECH outside. Harriet jerks her head to the right and we follow her line of sight.

A sleek motorcycle has swerved out of its lane and is now streaking straight for the right side of the Camaro beside Harriet's car.

The bike slams with CLANG into the side of the Camaro. Its rider is flung up and forward into the air, twirling passed Harriet's windshield.

We now watch from Harriet's POV, in slow motion. The black-leather-clad motorcyclist sails by Harriet's windshield, airborne. We see a man's face, clearly: His elephant-hide skin tells us that he is well beyond middle-age. Yet thick, black curls of youthful hair emerge from under his helmet. The rider has only one half of a black, bushy, swept-out, waxed mustache. His eyes are weary and grey, and appear to lock with Harriet's for an instant.

We return to normal speed. The body is now lying on the incoming lane to the left of Harriet's Subaru, perfectly still on the blacktop, the head twisted into an impossible angle. Blood seeps from a nostril. Beside the lifeless head, a BMW medallion lies on the pavement, glinting in the sunlight.

The Contemporary Craft of Creating Characters
Meets Today's Cognitive Architectures:
A Case Study in Expository Writing

Salvatore Strappalà • John Lurie • Alexander Strappalà
version of 02/16/2007

Contents

1. Introduction	1
2. What is Expository?	2
3. Double-Minded Man, and Expository Challenges: Theoretical	3
4. The "Double-Minded" Man	4
5. The "Double-Minded" Man	5
6. The "Double-Minded" Man	6
7. The "Double-Minded" Man	7
8. The "Double-Minded" Man	8
9. The "Double-Minded" Man	9
10. The "Double-Minded" Man	10
11. The "Double-Minded" Man	11
12. The "Double-Minded" Man	12
13. The "Double-Minded" Man	13
14. The "Double-Minded" Man	14
15. The "Double-Minded" Man	15
16. The "Double-Minded" Man	16
17. The "Double-Minded" Man	17
18. The "Double-Minded" Man	18
19. The "Double-Minded" Man	19
20. The "Double-Minded" Man	20
21. The "Double-Minded" Man	21
22. The "Double-Minded" Man	22
23. The "Double-Minded" Man	23
24. The "Double-Minded" Man	24
25. The "Double-Minded" Man	25
26. The "Double-Minded" Man	26
27. The "Double-Minded" Man	27
28. The "Double-Minded" Man	28
29. The "Double-Minded" Man	29
30. The "Double-Minded" Man	30
31. The "Double-Minded" Man	31
32. The "Double-Minded" Man	32
33. The "Double-Minded" Man	33
34. The "Double-Minded" Man	34
35. The "Double-Minded" Man	35
36. The "Double-Minded" Man	36
37. The "Double-Minded" Man	37
38. The "Double-Minded" Man	38
39. The "Double-Minded" Man	39
40. The "Double-Minded" Man	40
41. The "Double-Minded" Man	41
42. The "Double-Minded" Man	42
43. The "Double-Minded" Man	43
44. The "Double-Minded" Man	44
45. The "Double-Minded" Man	45
46. The "Double-Minded" Man	46
47. The "Double-Minded" Man	47
48. The "Double-Minded" Man	48
49. The "Double-Minded" Man	49
50. The "Double-Minded" Man	50
51. The "Double-Minded" Man	51
52. The "Double-Minded" Man	52
53. The "Double-Minded" Man	53
54. The "Double-Minded" Man	54
55. The "Double-Minded" Man	55
56. The "Double-Minded" Man	56
57. The "Double-Minded" Man	57
58. The "Double-Minded" Man	58
59. The "Double-Minded" Man	59
60. The "Double-Minded" Man	60
61. The "Double-Minded" Man	61
62. The "Double-Minded" Man	62
63. The "Double-Minded" Man	63
64. The "Double-Minded" Man	64
65. The "Double-Minded" Man	65
66. The "Double-Minded" Man	66
67. The "Double-Minded" Man	67
68. The "Double-Minded" Man	68
69. The "Double-Minded" Man	69
70. The "Double-Minded" Man	70
71. The "Double-Minded" Man	71
72. The "Double-Minded" Man	72
73. The "Double-Minded" Man	73
74. The "Double-Minded" Man	74
75. The "Double-Minded" Man	75
76. The "Double-Minded" Man	76
77. The "Double-Minded" Man	77
78. The "Double-Minded" Man	78
79. The "Double-Minded" Man	79
80. The "Double-Minded" Man	80
81. The "Double-Minded" Man	81
82. The "Double-Minded" Man	82
83. The "Double-Minded" Man	83
84. The "Double-Minded" Man	84
85. The "Double-Minded" Man	85
86. The "Double-Minded" Man	86
87. The "Double-Minded" Man	87
88. The "Double-Minded" Man	88
89. The "Double-Minded" Man	89
90. The "Double-Minded" Man	90
91. The "Double-Minded" Man	91
92. The "Double-Minded" Man	92
93. The "Double-Minded" Man	93
94. The "Double-Minded" Man	94
95. The "Double-Minded" Man	95
96. The "Double-Minded" Man	96
97. The "Double-Minded" Man	97
98. The "Double-Minded" Man	98
99. The "Double-Minded" Man	99
100. The "Double-Minded" Man	100

References

We will not be in the same position, and other things. There are bright, colorful and vibrant, but nothing different and due to our own feelings.

Double-Minded Man



$$\exists X[X(joseph) \wedge \neg X(m(harriet, joseph)) \wedge Sleazy(X)]$$

1. TWIRL - DAY

68-year-old Harriet Smith sits with two wrinkled hands firmly on the wheel of her rust-eaten Subaru wagon, staring straight ahead through the top level of bifocals as she waits serenely at a red light.

Harriet is alone in the car. To her right is another vehicle, also waiting, in this case to make a right turn; it's a sleek, low-slung, black Camaro.

We are inside the cabin with Harriet. The Subaru's sound system softly plays choral music. Harriet's lips move slightly as she internally sings along, mouthing a slow aria. Her head weaves slightly side to side, in the rhythm with the music.

Things are calm as can be here inside the car with Harriet. There are a pair of well-worn Bibles on the empty passenger seat beside her, one with a gold-lettered 'Harriet' on its leather front cover, the other with a matching 'Joseph' on its front cover.

Harriet's eyes swivel up to the light: still red. We wait with her.

Suddenly there is a piercing SCREECH outside. Harriet jerks her head to the right and we follow her line of sight.

A sleek motorcycle has swerved out of its lane and is now streaking straight for the right side of the Camaro beside Harriet's car.

The bike slams with CLANG into the side of the Camaro. Its rider is flung up and forward into the air, twirling passed Harriet's windshield.

We now watch from Harriet's POV, in slow motion. The black-leather-clad motorcyclist sails by Harriet's windshield, airborne. We see a man's face, clearly: His elephant-hide skin tells us that he is well beyond middle-age. Yet thick, black curls of youthful hair emerge from under his helmet. The rider has only one half of a black, bushy, swept-out, waxed mustache. His eyes are weary and grey, and appear to lock with Harriet's for an instant.

We return to normal speed. The body is now lying on the incoming lane to the left of Harriet's Subaru, perfectly still on the blacktop, the head twisted into an impossible angle. Blood seeps from a nostril. Beside the lifeless head, a BMW medallion lies on the pavement, glinting in the sunlight.

Climbing the k -order Ladder

Climbing the k -order Ladder

a is a llama, as is b , a likes b , and the father of a is a llama as well.

Climbing the k -order Ladder

$Llama(a) \wedge Llama(b) \wedge Likes(a, b) \wedge Llama(fatherOf(a))$

a is a llama, as is b , a likes b , and the father of a is a llama as well.

Climbing the k -order Ladder

ZOL $Llama(a) \wedge Llama(b) \wedge Likes(a, b) \wedge Llama(fatherOf(a))$

a is a llama, as is b , a likes b , and the father of a is a llama as well.

Climbing the k -order Ladder

There's some thing which is a llama and likes b (which is also a llama), and whose father is a llama too.

ZOL $Llama(a) \wedge Llama(b) \wedge Likes(a, b) \wedge Llama(fatherOf(a))$

a is a llama, as is b , a likes b , and the father of a is a llama as well.

Climbing the k -order Ladder

$$\exists x [Llama(x) \wedge Llama(b) \wedge Likes(x, b) \wedge Llama(fatherOf(x))]$$

There's some thing which is a llama and likes b (which is also a llama), and whose father is a llama too.

ZOL $Llama(a) \wedge Llama(b) \wedge Likes(a, b) \wedge Llama(fatherOf(a))$

a is a llama, as is b , a likes b , and the father of a is a llama as well.

Climbing the k -order Ladder

FOL $\exists x [Llama(x) \wedge Llama(b) \wedge Likes(x, b) \wedge Llama(fatherOf(x))]$

There's some thing which is a llama and likes b (which is also a llama), and whose father is a llama too.

ZOL $Llama(a) \wedge Llama(b) \wedge Likes(a, b) \wedge Llama(fatherOf(a))$

a is a llama, as is b , a likes b , and the father of a is a llama as well.

Climbing the k -order Ladder

Things x and y , along with the father of x ,
share a certain property (and x likes y).

FOL $\exists x [Llama(x) \wedge Llama(b) \wedge Likes(x, b) \wedge Llama(fatherOf(x))]$

There's some thing which is a llama and likes b (which
is also a llama), and whose father is a llama too.

ZOL $Llama(a) \wedge Llama(b) \wedge Likes(a, b) \wedge Llama(fatherOf(a))$

a is a llama, as is b , a likes b , and the father of a is a llama as well.

Climbing the k -order Ladder

$$\exists x \exists y \exists R [R(x) \wedge R(y) \wedge Likes(x, y) \wedge R(fatherOf(x))]$$

Things x and y , along with the father of x ,
share a certain property (and x likes y).

FOL
$$\exists x [Llama(x) \wedge Llama(b) \wedge Likes(x, b) \wedge Llama(fatherOf(x))]$$

There's some thing which is a llama and likes b (which
is also a llama), and whose father is a llama too.

ZOL
$$Llama(a) \wedge Llama(b) \wedge Likes(a, b) \wedge Llama(fatherOf(a))$$

a is a llama, as is b , a likes b , and the father of a is a llama as well.

Climbing the k -order Ladder

SOL $\exists x \exists y \exists R [R(x) \wedge R(y) \wedge Likes(x, y) \wedge R(fatherOf(x))]$

Things x and y , along with the father of x ,
share a certain property (and x likes y).

FOL $\exists x [Llama(x) \wedge Llama(b) \wedge Likes(x, b) \wedge Llama(fatherOf(x))]$

There's some thing which is a llama and likes b (which
is also a llama), and whose father is a llama too.

ZOL $Llama(a) \wedge Llama(b) \wedge Likes(a, b) \wedge Llama(fatherOf(a))$

a is a llama, as is b , a likes b , and the father of a is a llama as well.

Climbing the k -order Ladder

Things x and y , along with the father of x , share a certain property; and, $x R^2s y$, where R^2 is a positive property.

SOL $\exists x \exists y \exists R [R(x) \wedge R(y) \wedge Likes(x, y) \wedge R(fatherOf(x))]$

Things x and y , along with the father of x , share a certain property (and x likes y).

FOL $\exists x [Llama(x) \wedge Llama(b) \wedge Likes(x, b) \wedge Llama(fatherOf(x))]$

There's some thing which is a llama and likes b (which is also a llama), and whose father is a llama too.

ZOL $Llama(a) \wedge Llama(b) \wedge Likes(a, b) \wedge Llama(fatherOf(a))$

a is a llama, as is b , a likes b , and the father of a is a llama as well.

Climbing the k -order Ladder

$$\exists x, y \exists R, R^2 [R(x) \wedge R(y) \wedge R^2(x, y) \wedge \textit{Positive}(R^2) \wedge R(\textit{fatherOf}(x))]$$

Things x and y , along with the father of x , share a certain property; and, x R^2 s y , where R^2 is a positive property.

SOL $\exists x \exists y \exists R [R(x) \wedge R(y) \wedge \textit{Likes}(x, y) \wedge R(\textit{fatherOf}(x))]$

Things x and y , along with the father of x , share a certain property (and x likes y).

FOL $\exists x [Llama(x) \wedge Llama(b) \wedge \textit{Likes}(x, b) \wedge Llama(\textit{fatherOf}(x))]$

There's some thing which is a llama and likes b (which is also a llama), and whose father is a llama too.

ZOL $Llama(a) \wedge Llama(b) \wedge \textit{Likes}(a, b) \wedge Llama(\textit{fatherOf}(a))$

a is a llama, as is b , a likes b , and the father of a is a llama as well.

Climbing the k -order Ladder

TOL $\exists x, y \exists R, R^2 [R(x) \wedge R(y) \wedge R^2(x, y) \wedge \textit{Positive}(R^2) \wedge R(\textit{fatherOf}(x))]$

Things x and y , along with the father of x , share a certain property; and, x R^2 s y , where R^2 is a positive property.

SOL $\exists x \exists y \exists R [R(x) \wedge R(y) \wedge \textit{Likes}(x, y) \wedge R(\textit{fatherOf}(x))]$

Things x and y , along with the father of x , share a certain property (and x likes y).

FOL $\exists x [Llama(x) \wedge Llama(b) \wedge \textit{Likes}(x, b) \wedge Llama(\textit{fatherOf}(x))]$

There's some thing which is a llama and likes b (which is also a llama), and whose father is a llama too.

ZOL $Llama(a) \wedge Llama(b) \wedge \textit{Likes}(a, b) \wedge Llama(\textit{fatherOf}(a))$

a is a llama, as is b , a likes b , and the father of a is a llama as well.

Climbing the k -order Ladder

TOL $\exists x, y \exists R, R^2 [R(x) \wedge R(y) \wedge R^2(x, y) \wedge \textit{Positive}(R^2) \wedge R(\textit{fatherOf}(x))]$

\mathcal{L}_3 Things x and y , along with the father of x , share a certain property; and, x R^2 s y , where R^2 is a positive property.

SOL $\exists x \exists y \exists R [R(x) \wedge R(y) \wedge \textit{Likes}(x, y) \wedge R(\textit{fatherOf}(x))]$

\mathcal{L}_2 Things x and y , along with the father of x , share a certain property (and x likes y).

FOL $\exists x [Llama(x) \wedge Llama(b) \wedge \textit{Likes}(x, b) \wedge Llama(\textit{fatherOf}(x))]$

\mathcal{L}_1 There's some thing which is a llama and likes b (which is also a llama), and whose father is a llama too.

ZOL $Llama(a) \wedge Llama(b) \wedge \textit{Likes}(a, b) \wedge Llama(\textit{fatherOf}(a))$

\mathcal{L}_0 a is a llama, as is b , a likes b , and the father of a is a llama as well.

Climbing the k -order Ladder

⋮

TOL $\exists x, y \exists R, R^2 [R(x) \wedge R(y) \wedge R^2(x, y) \wedge \textit{Positive}(R^2) \wedge R(\textit{fatherOf}(x))]$

\mathcal{L}_3 Things x and y , along with the father of x , share a certain property; and, x R^2 s y , where R^2 is a positive property.

SOL $\exists x \exists y \exists R [R(x) \wedge R(y) \wedge \textit{Likes}(x, y) \wedge R(\textit{fatherOf}(x))]$

\mathcal{L}_2 Things x and y , along with the father of x , share a certain property (and x likes y).

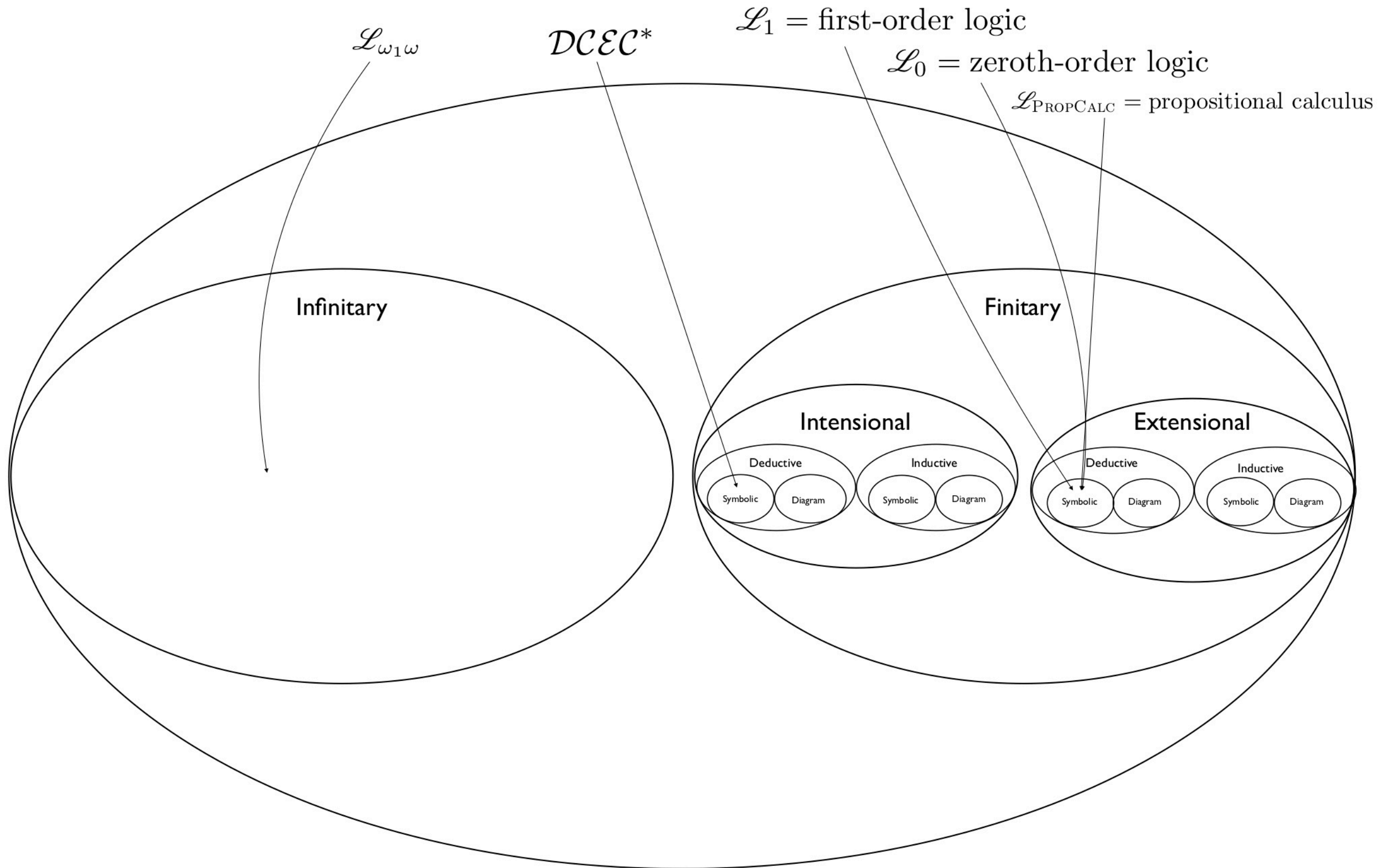
FOL $\exists x [Llama(x) \wedge Llama(b) \wedge \textit{Likes}(x, b) \wedge Llama(\textit{fatherOf}(x))]$

\mathcal{L}_1 There's some thing which is a llama and likes b (which is also a llama), and whose father is a llama too.

ZOL $Llama(a) \wedge Llama(b) \wedge \textit{Likes}(a, b) \wedge Llama(\textit{fatherOf}(a))$

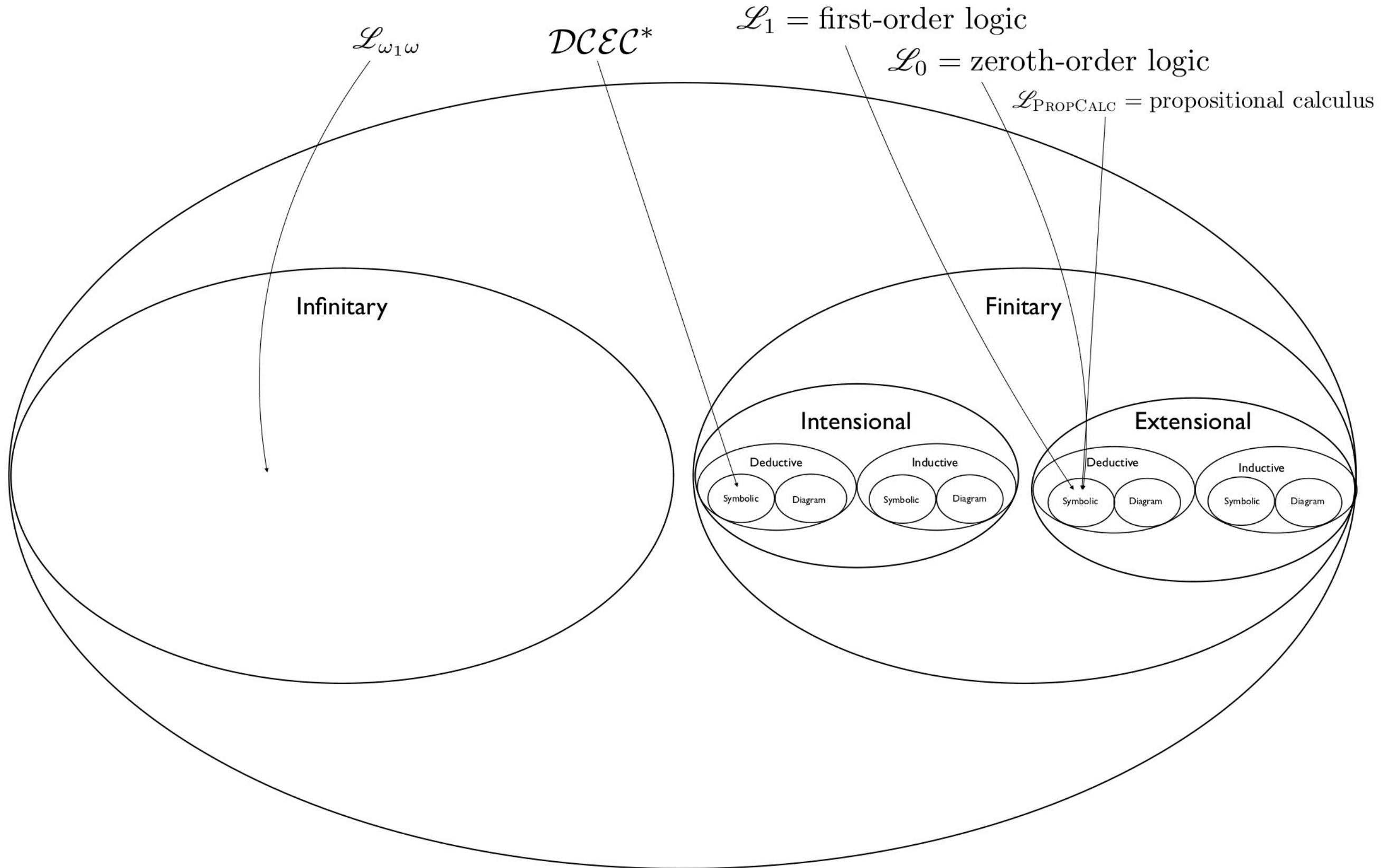
\mathcal{L}_0 a is a llama, as is b , a likes b , and the father of a is a llama as well.

The Universe of Logics

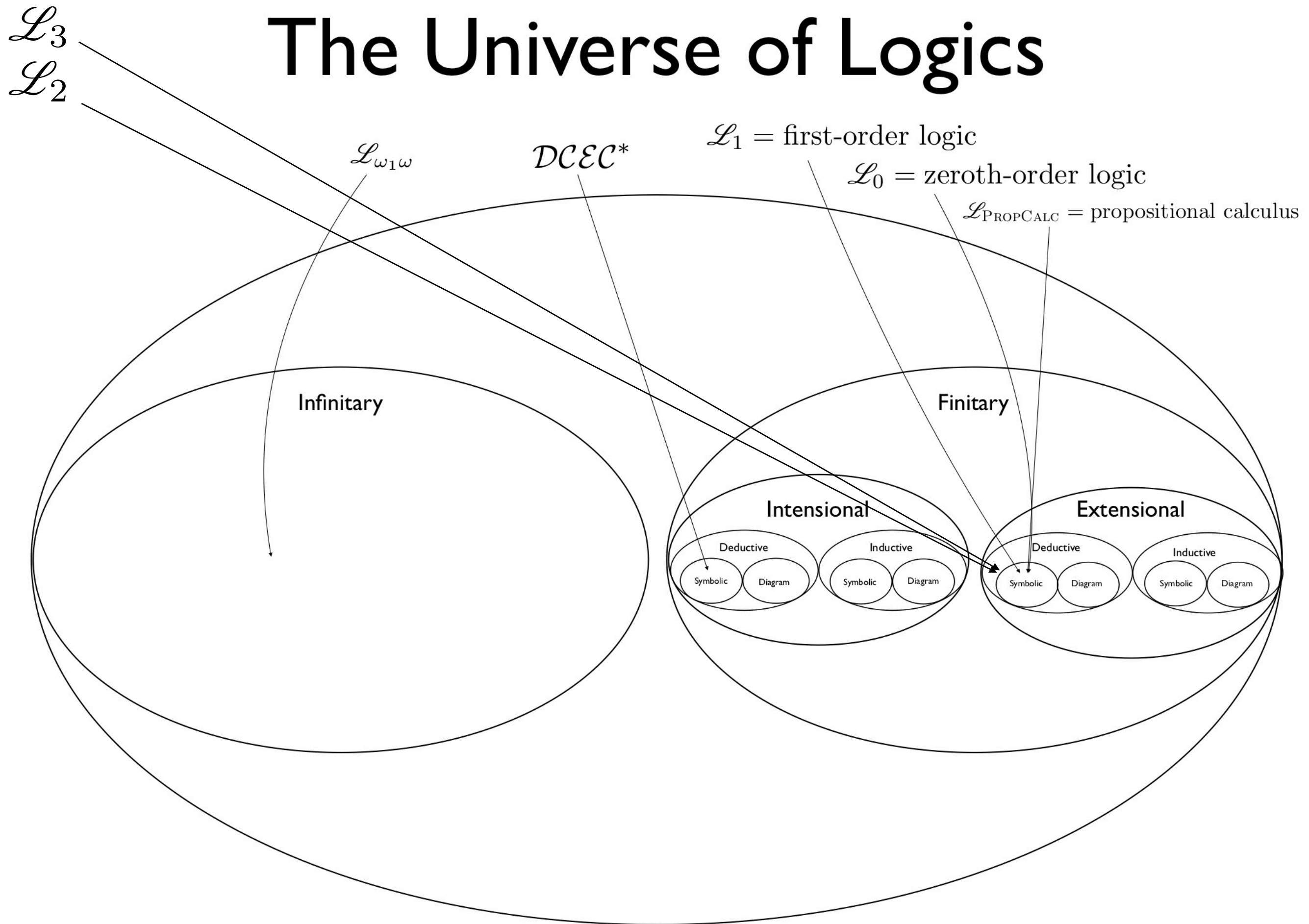


\mathcal{L}_3
 \mathcal{L}_2

The Universe of Logics



The Universe of Logics



Climbing the k -order Ladder

⋮

TOL $\exists x, y \exists R, R^2 [R(x) \wedge R(y) \wedge R^2(x, y) \wedge \textit{Positive}(R^2) \wedge R(\textit{fatherOf}(x))]$

\mathcal{L}_3 Things x and y , along with the father of x , share a certain property; and, x R^2 s y , where R^2 is a positive property.

SOL $\exists x \exists y \exists R [R(x) \wedge R(y) \wedge \textit{Likes}(x, y) \wedge R(\textit{fatherOf}(x))]$

\mathcal{L}_2 Things x and y , along with the father of x , share a certain property (and x likes y).

FOL $\exists x [Llama(x) \wedge Llama(b) \wedge \textit{Likes}(x, b) \wedge Llama(\textit{fatherOf}(x))]$

\mathcal{L}_1 There's some thing which is a llama and likes b (which is also a llama), and whose father is a llama too.

ZOL $Llama(a) \wedge Llama(b) \wedge \textit{Likes}(a, b) \wedge Llama(\textit{fatherOf}(a))$

\mathcal{L}_0 a is a llama, as is b , a likes b , and the father of a is a llama as well.

Astrologic; Second-Order Logic and the k -order Ladder; Second-Order Axiomatized Arithmetic; Gödel's “God Theorem” & Speedup Theorem

Selmer Bringsjord

Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Lally School of Management
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

Intro to Logic
3/26/2020



Astrologic; Second-Order Logic and the k -order Ladder; Second-Order Axiomatized Arithmetic; Gödel's “God Theorem” & Speedup Theorem

Selmer Bringsjord

Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Lally School of Management
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

Intro to Logic
3/26/2020







“Leibniz was right that Descartes was right that ... : God exists, necessarily.”

Gödel's "God Theorem"

		Part I	
	(1')	The absence of a positive property is not positive.	premise
	(1)	The absence of a positive property is not positive; and if a property isn't positive, the absence of that property is.	premise
	(2)	Any property entailed by a positive property is itself positive.	premise
∴	(3)	Every positive property P is possibly possessed by something.	(1), (2)
		Part II	
	(4)	Anything that is God has all positive properties.	definition
	(5)	The property of being God is itself a positive property.	premise
∴	(6)	It's possible that God exists.	(3), (5)
		Part III	
	(7)	Positive properties are necessarily positive.	premise
	(8)	A thing x has an essence E if and only if (i) E is a property x has; and (ii) for any property P that x has, x 's having this property P is necessarily implied by x 's having essence E .	definition
∴	(9)	The property of being God is an essence of any thing that has this property.	(8), (7), (4), (1)
∴	(9)	The property of <i>being God</i> ($= G$) is an essential property of any thing that has G .	(8), (7), (4), (1)
		Part IV	
	(10)	A thing has necessary existence if and only if all the essences that thing has imply that something exists and has all those essences.	definition
	(10)	A thing has necessary existence if and only if all the essential properties that thing has imply that something exists and has all those essential properties.	definition
	(11)	Necessary existence is a positive property.	premise
∴	(12)	Necessarily, God exists.	(6), (9), (10), (11)
			QED

Gödel's “God Theorem” (formalized, machine verified)

	(1)	$\forall P [Pos(\neg P) \leftrightarrow \neg Pos(P)]$	premise
	(2)	$\forall P_1 \forall P_2 \{Pos(P_1) \wedge \Box \forall x [P_1(x) \rightarrow P_2(x)] \rightarrow Pos(P_2)\}$	premise
\therefore	(3)	$\forall P [Pos(P) \rightarrow \Diamond \exists x P(x)]$	theorem
	(4)	$\forall x [G(x) \leftrightarrow \forall P [Pos(P) \rightarrow P(x)]]$	definition
	(5)	$Pos(G)$	premise
\therefore	(6)	$\Diamond \exists x G(x)$	corollary
	(7)	$\forall P [Pos(P) \rightarrow \Box Pos(P)]$	premise
	(8)	$\forall x \forall P \{Ess(P, x) \leftrightarrow [P(x) \wedge \forall P' (P'(x) \rightarrow \Box \forall y (P(y) \rightarrow P'(y)))]\}$	definition
\therefore	(9)	$\forall x [G(x) \rightarrow Ess(G, x)]$	theorem
	(10)	$\forall x \{NE(x) \leftrightarrow \forall P [Ess(P, x) \rightarrow \Box \exists y P(y)]\}$	definition
	(11)	$Pos(NE)$	premise
\therefore	(12)	$\Box \exists x G(x)$ (a.k.a. “Necessarily, God exists.”)	theorem

Gödel's “God Theorem” (formalized, machine verified)

	(1)	$\forall P [Pos(\neg P) \leftrightarrow \neg Pos(P)]$	premise
	(2)	$\forall P_1 \forall P_2 \{Pos(P_1) \wedge \Box \forall x [P_1(x) \rightarrow P_2(x)] \rightarrow Pos(P_2)\}$	premise
\therefore	(3)	$\forall P [Pos(P) \rightarrow \Diamond \exists x P(x)]$	theorem
	(4)	$\forall x [G(x) \leftrightarrow \forall P [Pos(P) \rightarrow P(x)]]$	definition
	(5)	$Pos(G)$	premise
\therefore	(6)	$\Diamond \exists x G(x)$	corollary
	(7)	$\forall P [Pos(P) \rightarrow \Box Pos(P)]$	premise
	(8)	$\forall x \forall P \{Ess(P, x) \leftrightarrow [P(x) \wedge \forall P' (P'(x) \rightarrow \Box \forall y (P(y) \rightarrow P'(y)))]\}$	definition
\therefore	(9)	$\forall x [G(x) \rightarrow Ess(G, x)]$	theorem
	(10)	$\forall x \{NE(x) \leftrightarrow \forall P [Ess(P, x) \rightarrow \Box \exists y P(y)]\}$	definition
	(11)	$Pos(NE)$	premise
\therefore	(12)	$\Box \exists x G(x)$ (a.k.a. “Necessarily, God exists.”)	theorem

Astrologic; Second-Order Logic and the k -order Ladder; Second-Order Axiomatized Arithmetic; Gödel's “God Theorem” & Speedup Theorem

Selmer Bringsjord

Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Lally School of Management
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

Intro to Logic
3/26/2020



Astrologic; Second-Order Logic and the k -order Ladder; Second-Order Axiomatized Arithmetic; Gödel's “God Theorem” & Speedup Theorem

Selmer Bringsjord

Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Lally School of Management
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

Intro to Logic
3/26/2020



$$\mathbf{PA}_I = \mathbf{Z}_I$$

$$\text{A1} \quad \forall x(0 \neq s(x))$$

$$\text{A2} \quad \forall x \forall y (s(x) = s(y) \rightarrow x = y)$$

$$\text{A3} \quad \forall x (x \neq 0 \rightarrow \exists y (x = s(y)))$$

$$\text{A4} \quad \forall x (x + 0 = x)$$

$$\text{A5} \quad \forall x \forall y (x + s(y) = s(x + y))$$

$$\text{A6} \quad \forall x (x \times 0 = 0)$$

$$\text{A7} \quad \forall x \forall y (x \times s(y) = (x \times y) + x)$$

And, every sentence that is the universal closure of an instance of

$$([\phi(0) \wedge \forall x(\phi(x) \rightarrow \phi(s(x)))] \rightarrow \forall x \phi(x))$$

where $\phi(x)$ is open wff with variable x , and perhaps others, free.

$$\text{PA}_I = \text{Z}_I$$

$$\text{A1} \quad \forall x(0 \neq s(x))$$

$$\text{A2} \quad \forall x \forall y (s(x) = s(y) \rightarrow x = y)$$

$$\text{A3} \quad \forall x (x \neq 0 \rightarrow \exists y (x = s(y)))$$

$$\text{A4} \quad \forall x (x + 0 = x)$$

$$\text{A5} \quad \forall x \forall y (x + s(y) = s(x + y))$$

$$\text{A6} \quad \forall x (x \times 0 = 0)$$

$$\text{A7} \quad \forall x \forall y (x \times s(y) = (x \times y) + x)$$

And, every sentence that is the universal closure of an instance of

$$([\phi(0) \wedge \forall x(\phi(x) \rightarrow \phi(s(x)))] \rightarrow \forall x \phi(x)) \quad \infty$$

where $\phi(x)$ is open wff with variable x , and perhaps others, free.

$$\mathsf{PA}_2 = \mathsf{Z}_2$$

$$\mathsf{A1} \quad \forall x(0 \neq s(x))$$

$$\mathsf{A2} \quad \forall x \forall y (s(x) = s(y) \rightarrow x = y)$$

$$\mathsf{A3} \quad \forall x (x \neq 0 \rightarrow \exists y (x = s(y)))$$

$$\mathsf{A4} \quad \forall x (x + 0 = x)$$

$$\mathsf{A5} \quad \forall x \forall y (x + s(y) = s(x + y))$$

$$\mathsf{A6} \quad \forall x (x \times 0 = 0)$$

$$\mathsf{A7} \quad \forall x \forall y (x \times s(y) = (x \times y) + x)$$

(if we drop any restriction C, we have full second-order arithmetic.)

$$\mathbf{PA}_2 = \mathbf{Z}_2$$

$$\text{A1} \quad \forall x(0 \neq s(x))$$

$$\text{A2} \quad \forall x \forall y (s(x) = s(y) \rightarrow x = y)$$

$$\text{A3} \quad \forall x (x \neq 0 \rightarrow \exists y (x = s(y)))$$

$$\text{A4} \quad \forall x (x + 0 = x)$$

$$\text{A5} \quad \forall x \forall y (x + s(y) = s(x + y))$$

$$\text{A6} \quad \forall x (x \times 0 = 0)$$

$$\text{A7} \quad \forall x \forall y (x \times s(y) = (x \times y) + x)$$

$$\text{Induction Axiom} \quad \forall X ([X(0) \wedge \forall x (X(x) \rightarrow X(s(x)))] \rightarrow \forall x X(x))$$

(if we drop any restriction C, we have full second-order arithmetic.)

$$\mathbf{PA}_2 = \mathbf{Z}_2$$

$$\text{A1} \quad \forall x(0 \neq s(x))$$

$$\text{A2} \quad \forall x \forall y (s(x) = s(y) \rightarrow x = y)$$

$$\text{A3} \quad \forall x (x \neq 0 \rightarrow \exists y (x = s(y)))$$

$$\text{A4} \quad \forall x (x + 0 = x)$$

$$\text{A5} \quad \forall x \forall y (x + s(y) = s(x + y))$$

$$\text{A6} \quad \forall x (x \times 0 = 0)$$

$$\text{A7} \quad \forall x \forall y (x \times s(y) = (x \times y) + x)$$

$$\text{Induction Axiom} \quad \forall X ([X(0) \wedge \forall x (X(x) \rightarrow X(s(x)))] \rightarrow \forall x X(x))$$

$$\text{Comprehension Schema} \quad \exists X (\forall x X(x) \leftrightarrow \phi(x)) \quad \text{where } \phi(x) \in \mathcal{C}$$

(if we drop any restriction \mathcal{C} , we have full second-order arithmetic.)

Astrologic; Second-Order Logic and the k -order Ladder; Second-Order Axiomatized Arithmetic; Gödel's “God Theorem” & Speedup Theorem


Selmer Bringsjord

Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Lally School of Management
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

Intro to Logic
3/26/2020



Astrologic; Second-Order Logic and the k -order Ladder; Second-Order Axiomatized Arithmetic; Gödel's “God Theorem” & Speedup Theorem



Selmer Bringsjord

Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Lally School of Management
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

Intro to Logic
3/26/2020



Analogico-Deductive Generation of Gödel's First Incompleteness Theorem from the Liar Paradox

John Licato, Naveen Sundar Govindarajulu, Selmer Bringsjord, Michael Pomeranz, Logan Gittelson
 Rensselaer Polytechnic Institute
 Troy, NY

{licatj.govinn,selmer.pomerm,gittel}@rpi.edu

Abstract

Gödel's proof of his famous first incompleteness theorem (G1) has quite understandably long been a tantalizing target for those wanting to engineer impressively intelligent computational systems. After all, in establishing G1, Gödel did something that by any metric must be classified as stunningly intelligent. We observe that it has long been understood that there is some sort of analogical relationship between the Liar Paradox (LP) and G1, and that Gödel himself appreciated and exploited the relationship. Yet the exact nature of the relationship has hitherto not been uncovered, by which we mean that the following question has not been answered: Given a description of LP and the suspicion that it may somehow be used by a suitably programmed computing machine to find a proof of the incompleteness of Peano Arithmetic, can such a machine, provided this description as input, produce as output a complete and verifiably correct proof of G1? In this paper, we summarize engineering that entails an affirmative answer to this question. Our approach uses what we call *analogico-deductive reasoning* (ADR), which combines analogical and deductive reasoning to produce a full deductive proof of G1 from LP. Our engineering uses a form of ADR based on our META-R system, and a connection between the Liar Sentence in LP and Gödel's Fixed Point Lemma, from which G1 follows quickly.

1 Introduction

Gödel's proofs of his incompleteness theorems are among the greatest intellectual achievements of the 20th century. Even armed with the suggestion that the Liar Paradox (LP) might somehow be useful as a guide to proving the incompleteness of Peano Arithmetic (PA)¹, the level of creativity and philosophical clarity required to actually tie the two concepts together and produce a valid proof is staggering: it certainly

¹G1 of course applies to any axiom system meeting the standard conditions (Turing-decidability, representability, consistency), but we tend to refer to PA for concreteness.

should not be controversial to claim that no computational reasoning system can, at present, achieve this sort of feat without significant human assistance.

1.1 Automating the Proof of G1

Prior work devoted to producing computational systems able to prove G1 have yielded systems able to prove this theorem only when the distance between this result and the starting point is quite small. This for example holds for the first (and certainly seminal) foray, i.e., for [Quine, 1988], as explained in [Bringsjord, 1998], where it's shown that the proof of G1, because the set of premises includes an ingenious human-derived encoding scheme, is very easy—in the point of being at the level of proofs requested from students in introductory mathematical logic classes.

Likewise, [Ammon, 1993] is an exact parallel of the human-derived proof given by [Kleene, 1966]. Finally, in much more recent and truly impressive work by [Sing and Field, 2002], there is a move to natural-deduction formats, which we applaud—but the machine essentially begins its processing at a point exceedingly close to where it needs to end up. As Sing and Field concede, "As axioms we take for granted the representability and derivability conditions for the central syntactic notions as well as the diagonal lemma for constructing self-referential sentences." If one takes for granted such things, finding a proof of G1 is effortless for a computing machine.² In sum, while a lot of commendable work has been done to build the foundation for our perspective work, the daunting formal and engineering challenge of producing a computational system able to produce G1 without clever seeding from a human remains entirely unmet.

2 The Analogico-Deductive Approach

2.1 Conjecture Generation

The problem with the purely deductive method is simple: that it does not allow us to come close to the type of model-based reasoning that great thinkers are known to have used. Gödel himself has been described as having a "line of thought [which] seems to move from conjecture to conjecture" [Wang, 1995]. Reasoners in general are known to conjecture through analogy when a straightforward answer

²A video demonstration of the small-distance process can be found at <http://kryon.msu.ru/cda/GodelLabraces.in.Slate.mov>

G

G

Small Steps Toward Hypercomputation via Infinitary Machine Proof Verification and Proof Generation

Naveen Sundar Govindarajulu, John Licato, and Selmer Bringsjord
 Department of Computer Science
 Department of Cognitive Science

Rensselaer AI & Reasoning Laboratory
 govinn@rpi.edu • licatj@rpi.edu • selmer@rpi.edu

Rensselaer Polytechnic Institute
 110 8th Street, Troy, NY 12180 USA

Abstract. After setting a context based on two general points (that humans appear to reason in infinitary fashion, and too, that actual hypercomputers aren't currently available to directly model and replicate such infinitary reasoning), we set a humble engineering goal of taking initial steps toward a computing machine that can reason in infinitary fashion. The initial steps consist in our outline of automated proof-verification and proof-discovery techniques for theorems independent of PA that seem to require an understanding and use of infinitary concepts. We specifically focus on infinitary techniques that make use of a marriage of analogical and deductive reasoning (which we call *analogico-deductive reasoning*).

A Context: Infinitary Reasoning, Hypercomputation, and Humble Engineering

Bringsjord has repeatedly pointed out the obvious fact that the behavior of formal scientists, taken at face value, involve various infinitary structures and reasoning. (We say "at face value" to simply indicate we don't presuppose some view that denies the reality of infinite entities routinely involved in the formal sciences.) For example, in [Bringsjord & van Hevelin 2003], Bringsjord himself operates as such a scientist in presenting an infinitary paradox which to his knowledge has yet to be solved. And he has argued that apparently infinitary behavior constitutes a grave challenge to AI and the Church-Turing Thesis (e.g., see Bringsjord & Arkouas 2006, Bringsjord & Zenzen 2003). More generally, Bringsjord conjectures that every human-produced proof of a theorem independent of Peano Arithmetic (PA) will make use of infinitary structures and reasoning, when those structures are taken at face value.¹ We have ourselves designed logico-computational logics for handling infinitary reasoning (e.g., see the treatment of the infinitized wise-man puzzle: Arkouas & Bringsjord 2005), but this work simply falls back on the human ability to carry out induction on the natural numbers: it doesn't dissect and explain this ability. Finally, it must be admitted by all that there is simply no systematic, comprehensive model or framework anywhere in the formal/computational approach to understanding human knowledge and intelligence that provides a theory about how humans are able to engage with infinitary structures. This is revealed perhaps most clearly when one studies the fruit produced by the part of formal AI devoted to producing discovery systems: such fruit is embarrassingly feeble (e.g., see Stillinger 2009).

Given this context, we are interested in exploring how one might give a machine the ability to reason in infinitary fashion. We are not saying that we in fact have figured out how to give such ability to a computing machine. Our objective here is much more humble and limited: it is to push forward in the attempt to engineer a computing machine that has the ability to reason in infinitary fashion. Ultimately, if such an attempt is to succeed, the computing machine in question will presumably be capable of outright hypercomputation. But the fact is that from an engineering perspective, we don't know how to create and harness a hypercomputer. So what we must first try to do, as explained in [Bringsjord & Zenzen 2003], is pursue engineering that initiates the attempt to engineer a hypercomputer, and takes the first few steps. In the present paper, the engineering is aimed specifically at giving a computing machine the ability to, in a limited but well-defined sense, reason in infinitary fashion. Even more specifically, our engineering is aimed at building a machine capable of at least providing a strong case for a result which, in the human sphere, has hitherto required use of infinitary techniques.

¹A weaker conjecture along the same line has been ventured by Isaacson, and is elegantly discussed by Smith (2007).

Analogico-Deductive Generation of Gödel's First Incompleteness Theorem from the Liar Paradox

John Licato, Naveen Sundar Govindarajulu, Selmer Bringsjord, Michael Pomeranz, Logan Gittelson
 Rensselaer Polytechnic Institute
 Troy, NY
 {licatj.govinn.selmer.pomerm.gittel}@rpi.edu

Abstract

Gödel's proof of his famous first incompleteness theorem (G1) has quite understandably long been a tantalizing target for those wanting to engineer impressively intelligent computational systems. After all, in establishing G1, Gödel did something that by any metric must be classified as stunningly intelligent. We observe that it has long been understood that there is some sort of analogical relationship between the Liar Paradox (LP) and G1, and that Gödel himself appreciated and exploited the relationship. Yet the exact nature of the relationship has hitherto not been uncovered, by which we mean that the following question has not been answered: Given a description of LP and the suspicion that it may somehow be used by a suitably programmed computing machine to find a proof of the incompleteness of Peano Arithmetic, can such a machine, provided this description as input, produce as output a complete and verifiably correct proof of G1? In this paper, we summarize engineering that entails an affirmative answer to this question. Our approach uses what we call *analogico-deductive reasoning* (ADR), which combines analogical and deductive reasoning to produce a full deductive proof of G1 from LP. Our engineering uses a form of ADR based on our META-R system, and a connection between the Liar Sentence in LP and Gödel's Fixed Point Lemma, from which G1 follows quickly.

1 Introduction

Gödel's proofs of his incompleteness theorems are among the greatest intellectual achievements of the 20th century. Even armed with the suggestion that the Liar Paradox (LP) might somehow be useful as a guide to proving the incompleteness of Peano Arithmetic (PA),¹ the level of creativity and philosophical clarity required to actually tie the two concepts together and produce a valid proof is staggering: it certainly

should not be controversial to claim that no computational reasoning system can, at present, achieve this sort of feat without significant human assistance.

1.1 Automating the Proof of G1

Prior work devoted to producing computational systems able to prove G1 have yielded systems able to prove this theorem only when the distance between this result and the starting point is quite small. This for example holds for the first (and certainly seminal) foray, i.e., for [Quine, 1988], as explained in [Bringsjord, 1998], where it's shown that the proof of G1, because the set of premises includes an ingenious human-derived encoding scheme, is very easy—in the point of being at the level of proofs requested from students in introductory mathematical logic classes.

Likewise, [Ammon, 1993] is an exact parallel of the human-derived proof given by [Kleene, 1996]. Finally, in much more recent and truly impressive work by [Sing and Field, 2002], there is a move to natural-deduction formats, which we applaud—but the machine essentially begins its processing at a point exceedingly close to where it needs to end up. As Sing and Field concede, "As axioms we take for granted the representability and derivability conditions for the central syntactic notions as well as the diagonal lemma for constructing self-referential sentences." If one takes for granted such things, finding a proof of G1 is effortless for a computing machine!² In sum, while a lot of commendable work has been done to build the foundation for our perspective work, the daunting formal and engineering challenge of producing a computational system able to produce G1 without clever seeding from a human remains entirely unmet.

2 The Analogico-Deductive Approach

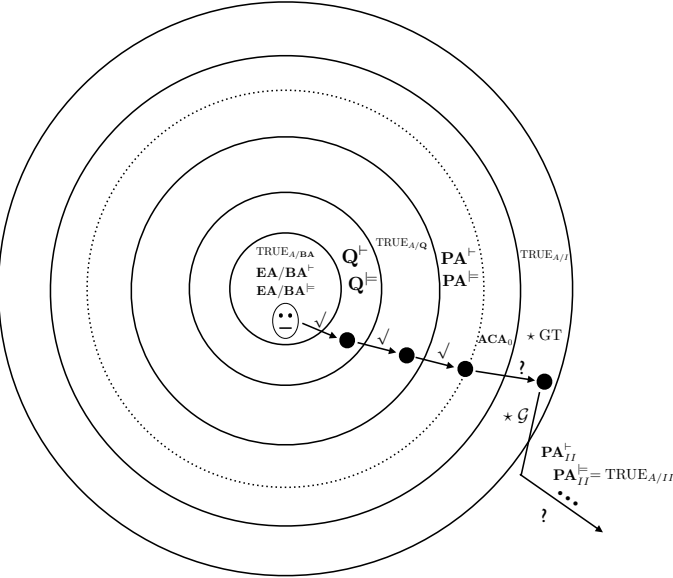
2.1 Conjecture Generation

The problem with the purely deductive method is simple: that it does not allow us to come close to the type of model-based reasoning that great thinkers are known to have used. Gödel himself has been described as having a "line of thought [which] seems to move from conjecture to conjecture" [Wang, 1995]. Reasoners in general are known to conjecture through analogy when a straightforward answer

¹G1 of course applies to any axiom system meeting the standard conditions (Turing-decidability, representability, consistency), but we tend to refer to PA for convenience.

²A video demonstration of the small-distance process can be found at <http://kryten.msu.ru/cda/GodelLab.html>. In [Sing, 2002].

G1



G

Small Steps Toward Hypercomputation via Infinitary Machine Proof Verification and Proof Generation

Naveen Sundar Govindarajulu, John Licato, and Selmer Bringsjord
 Department of Computer Science
 Department of Cognitive Science
 Rensselaer AI & Reasoning Laboratory
 govinn@rpi.edu • licatj@rpi.edu • selmer@rpi.edu
 Rensselaer Polytechnic Institute
 110 8th Street, Troy, NY 12180 USA

Abstract. After setting a context based on two general points (that humans appear to reason in infinitary fashion, and two, that actual hypercomputers aren't currently available to directly model and replicate such infinitary reasoning), we set a humble engineering goal of taking initial steps toward a computing machine that can reason in infinitary fashion. The initial steps consist in our outline of automated proof-verification and proof-discovery techniques for theorems independent of PA that seem to require an understanding and use of infinitary concepts. We specifically focus on proof-discovery techniques that make use of a marriage of analogical and deductive reasoning (which we call *analogico-deductive reasoning*).

A Context: Infinitary Reasoning, Hypercomputation, and Humble Engineering

Bringsjord has repeatedly pointed out the obvious fact that the behavior of formal scientists, taken at face value, involve various infinitary structures and reasoning. (We say "at face value" to simply indicate we don't presuppose some view that denies the reality of infinite entities routinely involved in the formal sciences.) For example, in [Bringsjord & van Hecke, 2003], Bringsjord himself operates as such a scientist in presenting an infinitary paradox which to his knowledge has yet to be solved. And he has argued that apparently infinitary behavior constitutes a great challenge to AI and the Church-Turing Thesis (e.g., see Bringsjord & Arkouas, 2006; Bringsjord & Zenzen, 2003). More generally, Bringsjord conjectures that every human-produced proof of a theorem independent of Peano Arithmetic (PA) will make use of infinitary structure and reasoning, when those structures are taken at face value.¹ We have ourselves designed logico-computational logics for handling infinitary reasoning (e.g., see the treatment of the infinitized wise-man puzzle: Arkouas & Bringsjord 2005), but this work simply falls back on the human ability to carry out induction on the natural numbers: it doesn't discover and explain this ability. Finally, it must be admitted by all that there is simply no systematic, comprehensive model or framework anywhere in the formal/computational approach to understanding human knowledge and intelligence that provides a theory about how humans are able to engage with infinitary structures. This is revealed perhaps most clearly when one studies the fruit produced by the part of formal AI devoted to producing discovery systems: such fruit is embarrassingly finitary (e.g., see Stillinger 2009).

Given this context, we are interested in exploring how one might give a machine the ability to reason in infinitary fashion. We are not saying that we in fact have figured out how to give such ability to a computing machine. Our objective here is much more humble and limited: it is to push forward in the attempt to engineer a computing machine that has the ability to reason in infinitary fashion. Ultimately, if such an attempt is to succeed, the computing machine in question will presumably be capable of outright hypercomputation. But the fact is that from an engineering perspective, we don't know how to create and harness a hypercomputer. So what we must first try to do, as explained in [Bringsjord & Zenzen 2003], is pursue engineering that initiates the attempt to engineer a hypercomputer, and takes the first few steps. In the present paper, the engineering is aimed specifically at giving a computing machine the ability to, in a limited but well-defined sense, reason in infinitary fashion. Even more specifically, our engineering is aimed at building a machine capable of at least providing a strong case for a result which, in the human sphere, has hitherto required use of infinitary techniques.

¹ A weaker conjecture along the same line has been ventured by Isaacson, and is elegantly discussed by Smith (2007).

Analogico-Deductive Generation of Gödel's First Incompleteness Theorem from the Liar Paradox

John Licato, Naveen Sundar Govindarajulu, Selmer Bringsjord, Michael Pomeranz, Logan Gittel
Rensselaer Polytechnic Institute
Troy, NY
{licatj.govinn.selmer.pomerm.gittel}@rpi.edu

Abstract

Gödel's proof of his famous first incompleteness theorem (G1) has quite understandably long been a tantalizing target for those wanting to engineer impressively intelligent computational systems. After all, in establishing G1, Gödel did something that by any metric must be classified as stunningly intelligent. We observe that it has long been understood that there is some sort of analogical relationship between the Liar Paradox (LP) and G1, and that Gödel himself appreciated and exploited the relationship. Yet the exact nature of the relationship has hitherto not been uncovered, by which we mean that the following question has not been answered: Given a description of LP and the suspicion that it may somehow be used by a suitably programmed computing machine to find a proof of the incompleteness of Peano Arithmetic, can such a machine, provided this description as input, produce as output a complete and verifiably correct proof of G1? In this paper, we summarize engineering that entails an affirmative answer to this question. Our approach uses what we call *analogico-deductive reasoning* (ADR), which combines analogical and deductive reasoning to produce a full deductive proof of G1 from LP. Our engineering uses a form of ADR based on our META-R system, and a connection between the Liar Sentence in LP and Gödel's Fixed Point Lemma, from which G1 follows quickly.

1 Introduction

Gödel's proofs of his incompleteness theorems are among the greatest intellectual achievements of the 20th century. Even armed with the suggestion that the Liar Paradox (LP) might somehow be useful as a guide to proving the incompleteness of Peano Arithmetic (PA),¹ the level of creativity and philosophical clarity required to actually tie the two concepts together and produce a valid proof is staggering: it certainly

¹G1 of course applies to any axiom system meeting the standard conditions (Turing-decidability, representability, consistency), but we tend to refer to PA for convenience.

should not be controversial to claim that no computer reasoning system can, at present, achieve this sort of feat without significant human assistance.

1.1 Automating the Proof of G1

Prior work devoted to producing computational systems able to prove G1 have yielded systems able to prove this theorem only when the distance between this result and the starting point is quite small. This for example holds for the first (a certainly seminal) foray, i.e., for [Quine, 1988], as explained in [Bringsjord, 1998], where it's shown that the proof of G because the set of premises includes an ingenious human devised encoding scheme, is very easy—in the point of view at the level of proofs requested from students in introductory mathematical logic classes.

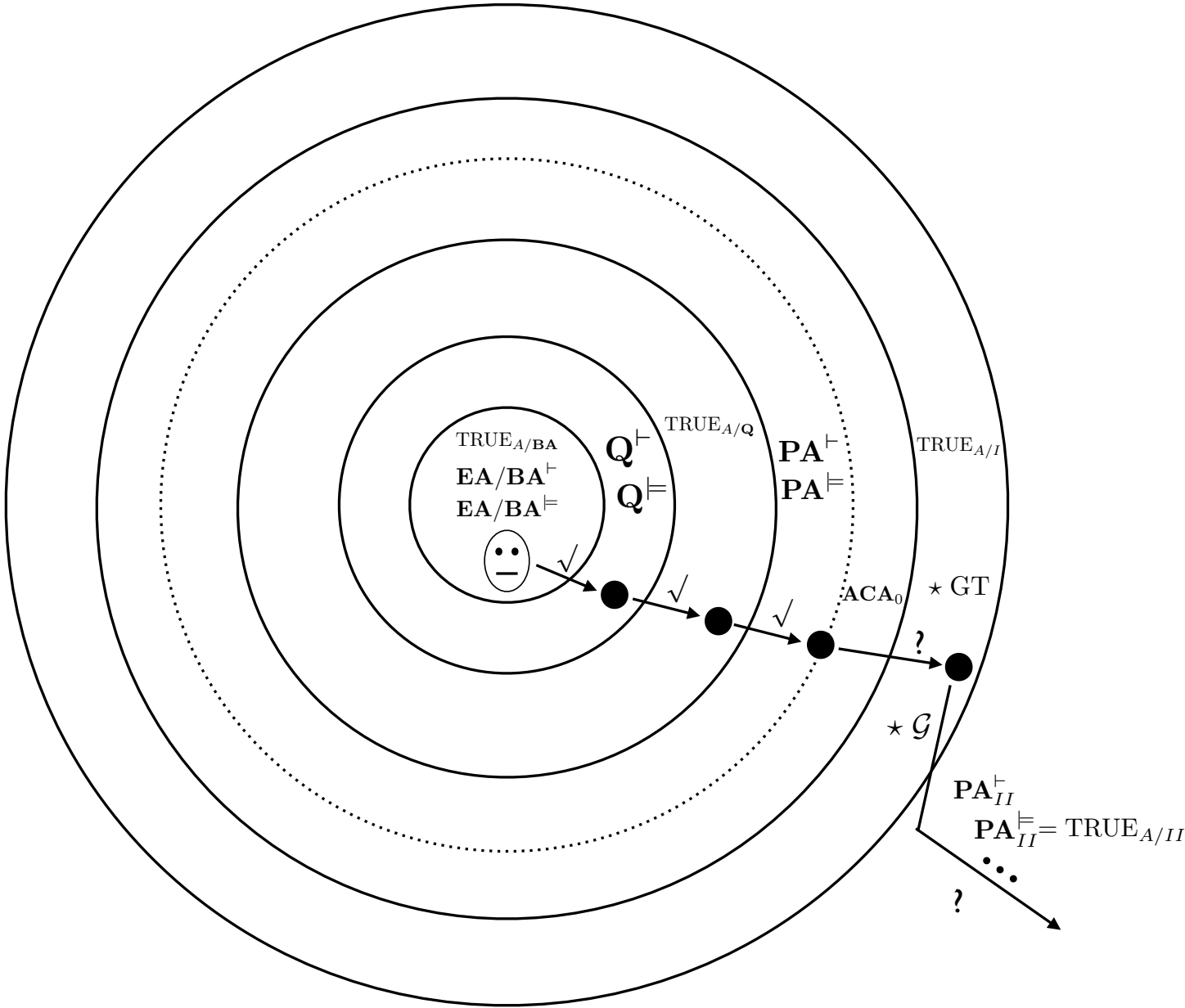
Likewise, [Ammon, 1993] is an exact parallel of the human derived proof given by [Kleene, 1969]. Finally, in my more recent and truly impressive work by [Sing and Fur 2005], there is a move to natural-deduction formats, which we explain—but the machine essentially begins its proving at a point exceedingly close to where it needs to end: *À la* Sieg and Field concede: "As axioms we take for granted the representability and derivability conditions for the central syntactic notions as well as the diagonal lemma for constructing self-referential sentences." If one takes for granted such things, finding a proof of G1 is effortless for a computing machine!" In sum, while a lot of commendable work has been done to build the foundation for our perspective was the daunting formal and engineering challenge of producing a computational system able to produce G1 without clues

2 The Analogico-Deductive Approach

2.1 Conjecture Generation

The problem with the purely deductive method is simple: that it does not allow us to come close to the type model-based reasoning that great thinkers are known to have used. Gödel himself has been described as having a "li of thought" (which) seems to move from conjecture to conjecture" [Wang, 1995]. Reasoners in general are known to conjecture through analogy when a straightforward answer

²A video demonstration of the small-distance process can be found at <http://kryon.msu.rpi.edu/GodelLabratory.in.Slate.mov>



Small Steps Toward Hypercomputation via Infinitary Machine Proof Verification and Proof Generation

Naveen Sundar Govindarajulu, John Licato, and Selmer Bringsjord
Department of Computer Science
Department of Cognitive Science
Rensselaer AI & Reasoning Laboratory
govinn@rpi.edu • licatj@rpi.edu • selmer@rpi.edu
Rensselaer Polytechnic Institute
110 8th Street, Troy, NY 12180 USA

Abstract. After setting a context based on two general points (that humans appear to reason in infinitary fashion, and two, that actual hypercomputers aren't currently available to directly model and replicate such infinitary reasoning), we set a humble engineering goal of taking initial steps toward a computing machine that can reason in infinitary fashion. The initial steps consist in our outline of automated proof-verification and proof-discovery techniques for theorems independent of PA that seem to require an understanding and use of infinitary concepts. We specifically focus on proof-discovery techniques that make use of a marriage of analogical and deductive reasoning (which we call *analogico-deductive reasoning*).

A Context: Infinitary Reasoning, Hypercomputation, and Humble Engineering

Bringsjord has repeatedly pointed out the obvious fact that the behavior of formal scientists, taken at face value, involve various infinitary structures and reasoning. (We say "at face value" to simply indicate we don't presuppose some view that denies the reality of infinite entities routinely involved in the formal sciences.) For example, in [Bringsjord & van Hecke 2003], Bringsjord himself operates as such a scientist in presenting an infinitary paradox which to his knowledge has yet to be solved. And he has argued that apparently infinitary behavior constitutes a great challenge to AI and the Church-Turing Thesis (e.g., see Bringsjord & Arkouas 2006, Bringsjord & Zenzen 2003). More generally, Bringsjord conjectures that every human-produced proof of a theorem independent of Peano Arithmetic (PA) will make use of infinitary structure and reasoning, when those structures are taken at face value.¹ We have ourselves designed logico-computational logics for handling infinitary reasoning (e.g., see the treatment of the infinitized wise-man puzzle: Arkouas & Bringsjord 2005), but this work simply falls back on the human ability to carry out induction on the natural numbers: it doesn't discuss and explain this ability. Finally, it must be admitted by all that there is simply no systematic, comprehensive model or framework anywhere in the formal/computational approach to understanding human knowledge and intelligence that provides a theory about how humans are able to engage with infinitary structures. This is revealed perhaps most clearly when one studies the fruit produced by the part of formal AI devoted to producing discovery systems: such fruit is embarrassingly finitary (e.g., see Stillinger 2009).

Given this context, we are interested in exploring how one might give a machine the ability to reason in infinitary fashion. We are not saying that we in fact have figured out how to give such ability to a computing machine. Our objective here is much more humble and limited: it is to push forward in the attempt to engineer a computing machine that has the ability to reason in infinitary fashion. Ultimately, if such an attempt is to succeed, the computing machine in question will presumably be capable of outright hypercomputation. But the fact is that from an engineering perspective, we don't know how to create and harness a hypercomputer. So what we must first try to do, as explained in (Bringsjord & Zenzen 2003), is pursue engineering that initiates the attempt to engineer a hypercomputer, and takes the first few steps. In the present paper, the engineering is aimed specifically at giving a computing machine the ability to, in a limited but well-defined sense, reason in infinitary fashion. Even more specifically, our engineering is aimed at building a machine capable of at least providing a strong case for a result which, in the human sphere, has hitherto required use of infinitary techniques.

¹A weaker conjecture along the same line has been ventured by Isaacson, and is elegantly discussed by Smith (2007).

Analogico-Deductive Generation of Gödel's First Incompleteness Theorem from the Liar Paradox

John Licato, Naveen Sundar Govindarajulu, Selmer Bringsjord, Michael Pomeranz, Logan Gittelson
 Rensselaer Polytechnic Institute
 Troy, NY

{licatj.govinn,selmer.pomerm,gittel}@rpi.edu

Abstract

Gödel's proof of his famous first incompleteness theorem (G1) has quite understandably long been a tantalizing target for those wanting to engineer impressively intelligent computational systems. After all, in establishing G1, Gödel did something that by any metric must be classified as stunningly intelligent. We observe that it has long been understood that there is some sort of analogical relationship between the Liar Paradox (LP) and G1, and that Gödel himself appreciated and exploited the relationship. Yet the exact nature of the relationship has hitherto not been uncovered, by which we mean that the following question has not been answered: Given a description of LP and the suspicion that it may somehow be used by a suitably programmed computing machine to find a proof of the incompleteness of Peano Arithmetic, can such a machine, provided this description as input, produce as output a complete and verifiably correct proof of G1? In this paper, we summarize engineering that entails an affirmative answer to this question. Our approach uses what we call *analogico-deductive reasoning* (ADR), which combines analogical and deductive reasoning to produce a full deductive proof of G1 from LP. Our engineering uses a form of ADR based on our META-R system, and a connection between the Liar Sentence in LP and Gödel's Fixed Point Lemma, from which G1 follows quickly.

1 Introduction

Gödel's proofs of his incompleteness theorems are among the greatest intellectual achievements of the 20th century. Even armed with the suggestion that the Liar Paradox (LP) might somehow be useful as a guide to proving the incompleteness of Peano Arithmetic (PA), the level of creativity and philosophical clarity required to actually tie the two concepts together and produce a valid proof is staggering: it certainly

¹G1 of course applies to any axiom system meeting the standard conditions (Turing-decidability, representability, consistency), but we tend to refer to PA for convenience.

should not be controversial to claim that no computational reasoning system can, at present, achieve this sort of feat without significant human assistance.

1.1 Automating the Proof of G1

Prior work devoted to producing computational systems able to prove G1 have yielded systems able to prove this theorem only when the distance between this result and the starting point is quite small. This for example holds for the first (and certainly seminal) foray, i.e., for [Quine, 1988], as explained in [Bringsjord, 1998], where it's shown that the proof of G1, because the set of premises includes an ingenious human-derived encoding scheme, is very easy—to the point of being at the level of proofs requested from students in introductory mathematical logic classes.

Likewise, [Ammon, 1993] is an exact parallel of the human-derived proof given by [Kleene, 1969]. Finally, in much more recent and truly impressive work by [Sing and Field, 2002], there is a move to natural-deduction formats, which we applaud—but the machine essentially begins its processing at a point exceedingly close to where it needs to end up. As Sing and Field concede, "As axioms we take for granted the representability and derivability conditions for the central syntactic notions as well as the diagonal lemma for constructing self-referential sentences." If one takes for granted such things, finding a proof of G1 is effortless for a computing machine.¹In sum, while a lot of commendable work has been done to build the foundation for our perspective work, the daunting formal and engineering challenge of producing a computational system able to produce G1 without clever seeding from a human remains entirely unmet.

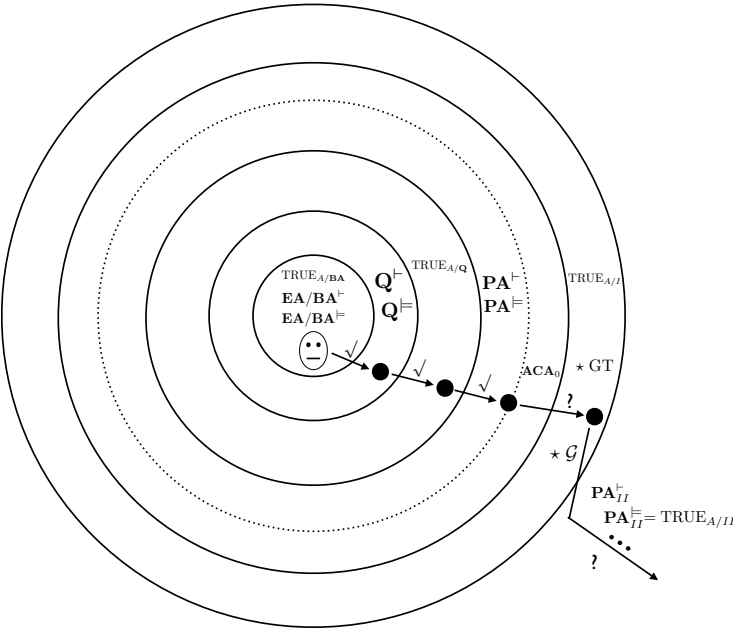
2 The Analogico-Deductive Approach

2.1 Conjecture Generation

The problem with the purely deductive method is simple that it does not allow us to come close to the type of model-based reasoning that great thinkers are known to have used. Gödel himself has been described as having a "line of thought [which] seems to move from conjecture to conjecture" [Wang, 1995]. Reasoners in general are known to conjecture through analogy when a straightforward answer

²A video demonstration of the small-distance process can be found at <http://kryon.msu.ru/cda/GodelLabraces.in.Slate.mov>

G1



G

Small Steps Toward Hypercomputation via Infinitary Machine Proof Verification and Proof Generation

Naveen Sundar Govindarajulu, John Licato, and Selmer Bringsjord

Department of Computer Science
 Department of Cognitive Science
 Rensselaer AI & Reasoning Laboratory
 govinn@rpi.edu • licatj@rpi.edu • selmer@rpi.edu

Rensselaer Polytechnic Institute
 110 8th Street, Troy, NY 12180 USA

Abstract. After setting a context based on two general points (that humans appear to reason in infinitary fashion, and two, that actual hypercomputers aren't currently available to directly model and replicate such infinitary reasoning), we set a humble engineering goal of taking initial steps toward a computing machine that can reason in infinitary fashion. The initial steps consist in our outline of automated proof-verification and proof-discovery techniques for theorems independent of PA that seem to require an understanding and use of infinitary concepts. We specifically focus on proof-discovery techniques that make use of a marriage of analogical and deductive reasoning (which we call *analogico-deductive reasoning*).

A Context: Infinitary Reasoning, Hypercomputation, and Humble Engineering

Bringsjord has repeatedly pointed out the obvious fact that the behavior of formal scientists, taken at face value, involve various infinitary structures and reasoning. (We say "at face value" to simply indicate we don't presuppose some view that denies the reality of infinite entities routinely involved in the formal sciences.) For example, in [Bringsjord & van Heuveln 2003], Bringsjord himself operates as such a scientist in presenting an infinitary paradox which to his knowledge has yet to be solved. And he has argued that apparently infinitary behavior constitutes a grave challenge to AI and the Church-Turing Thesis (e.g., see Bringsjord & Arkouas 2006, Bringsjord & Zenzen 2003). More generally, Bringsjord conjectures that every human-produced proof of a theorem independent of Peano Arithmetic (PA) will make use of infinitary structure and reasoning, when those structures are taken at face value.¹We have ourselves designed logico-computational logics for handling infinitary reasoning (e.g., see the treatment of the infinitized wise-man puzzle: Arkouas & Bringsjord 2005), but this work simply falls back on the human ability to carry out induction on the natural numbers: it doesn't dissect and explain this ability. Finally, it must be admitted by all that there is simply no systematic, comprehensive model or framework anywhere in the formal/computational approach to understanding human knowledge and intelligence that provides a theory about how humans are able to engage with infinitary structures. This is revealed perhaps most clearly when one studies the fruit produced by the part of formal AI devoted to producing discovery systems: such fruit is embarrassingly finitary (e.g., see Stillinger 2009).

Given this context, we are interested in exploring how one might give a machine the ability to reason in infinitary fashion. We are not saying that we in fact have figured out how to give such ability to a computing machine. Our objective here is much more humble and limited: it is to push forward in the attempt to engineer a computing machine that has the ability to reason in infinitary fashion. Ultimately, if such an attempt is to succeed, the computing machine in question will presumably be capable of outright hypercomputation. But the fact is that from an engineering perspective, we don't know how to create and harness a hypercomputer. So what we must first try to do, as explained in (Bringsjord & Zenzen 2003), is pursue engineering that initiates the attempt to engineer a hypercomputer, and takes the first few steps. In the present paper, the engineering is aimed specifically at giving a computing machine the ability to, in a limited but well-defined sense, reason in infinitary fashion. Even more specifically, our engineering is aimed at building a machine capable of at least providing a strong case for a result which, in the human sphere, has hitherto required use of infinitary techniques.

¹A weaker conjecture along the same line has been ventured by Isaacson, and is elegantly discussed by Smith (2007).

Analógico-Deductive Generation of Gödel's First Incompleteness Theorem from the Liar Paradox

John Licato, Naveen Sundar Govindarajulu, Selmer Bringsjord, Michael Pomeranz, Logan Gittelsohn
Rensselaer Polytechnic Institute
Troy, NY
{licatj.govinn,selmer,pomerm,gittel}@rpi.edu

Abstract

Gödel's proof of his famous first incompleteness theorem (G1) has quite understandably long been a tantalizing target for those wanting to engineer imitations. In the present paper, we shall focus on G1, establishing G1, Gödel did something that by any metric must be classified as stunningly ingenious. We shall argue, however, that Gödel understood that there is some sort of analogical relationship between the Liar Paradox (LP) and G1, and that this understanding is crucial to understanding the relationship. Yet the exact nature of the relationship has hitherto not been uncovered, by which we mean that no one has been able to answer the following question: Given a description of LP, and the supposition that it may somehow be used by a suitably programmed computing machine to find a proof of its own inconsistency, can we then use the description of a machine, providing this description as input, produce, as output, a complete and verifiably correct and consistent paper on the subject of the Liar Paradox that entails an affirmative answer to this question? Our approach uses what we call analogical reasoning (AR) to answer this question. AR is a logical and deductive reasoning to produce a full deductive proof of G1 from LP. Our engineering approach is to use the description of LP as input, and a connection between the Liar Sentence in LP and G1's Fixed Point Lemma, from which G1 is

1 Introduction

Gödel's proofs of his incompleteness theorems are among the greatest intellectual achievements of the 20th century. Ever armed with the suggestion that the Liar Paradox (LP) might somehow be useful as a guide to proving the incompleteness of Peano Arithmetic (PA),¹ the level of creativity and philosophical clarity required to actually tie the two concepts together and produce a valid proof is staggering; it certainly

should not be controversial to claim that no computational reasoning system can, at present, achieve this sort of feat without significant human assistance.

1.1 Automating the Proof of G1

Prior work devoted to producing computational systems able to prove G1 have yielded systems able to prove this theorem only when the distance between this result and the starting point is quite small. This for example holds for the first (and certainly seminal) foray; i.e., for (Quaife, 1988), as explained in (Bringsjord, 1998), where it's shown that the proof of G1, because the set of premises includes an ingenious human-devised encoding scheme, is very easy—to the point of being at the level of proofs requested from students in introductory mathematical logic classes.

Likewise, [Amon, 1993] is an exact parallel of the human-derived proof given by [Kleene, 1996]. Finally, in machine more recent and truly impressive work by [Siegel and Field, 2005], there is a move to natural-deduction formats, which we applaud—but the machine essentially begins its processing at a point exceedingly close to where it needs to end up. As Siegel and Field concede: “As axioms we take for granted the representability and derivability conditions for the central syntactic notions, we will also the diagonalization and the self-referential constructions that are essential for constructing such things, finding a proof of G1 is effortless for a computing machine.” In sum, while a lot of commendable work has been done to build the foundation for our prospective work, the daunting formal and engineering challenge of producing a computational system able to produce G1 without clever

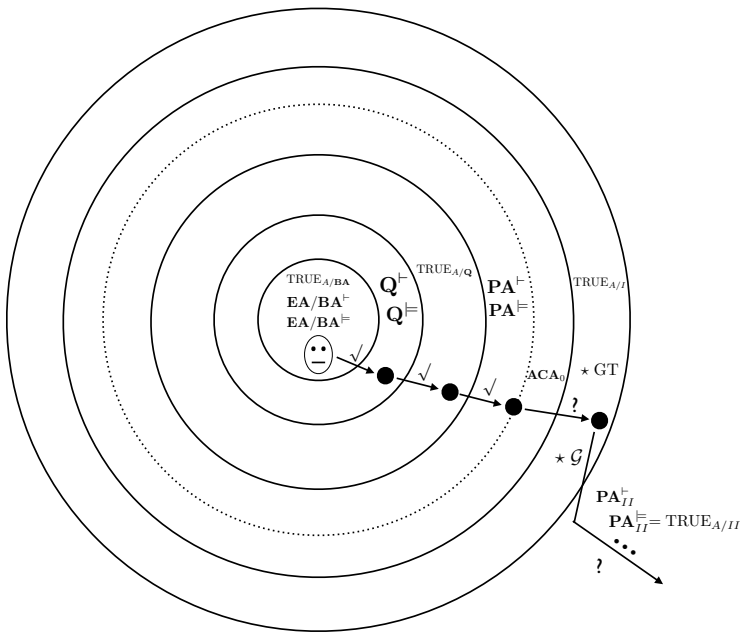
2 The Analogico-Deductive Approach

2.1 Conjecture Generation

The problem with the purely deductive method is simply that it does not allow us to come close to the type of model-based reasoning that great thinkers are known to have used. Gödel himself has been described as having a "line of thought [which] seems to move from conjecture to conjecture" (Wang, 1995). Reasoners in general are known to conjecture through analogy when a straightforward answer

²A video demonstration of the small-distance process can be found at http://kryten.mim.rpi.edu/Codell_abstract_in_Slats.mov

²A video demonstration of the small-distance process can be found at <http://kryten.mn.rpi.edu/Godell.abstract.in.Slate.mov>.



G

Small Steps Toward Hypercomputation via
Infinitary Machine Proof Verification and Proof Generation

Naveen Sundar Govindarajulu, John Licato, and Selmer Bringsjord
Department of Computer Science
Department of Cognitive Science
Rensselaer AI & Reasoning Laboratory
govinn@rpi.edu • licatj@rpi.edu • selmer@rpi.edu
Rensselaer Polytechnic Institute
110 8th Street, Troy, NY 12180 USA

Abstract. After setting a context based on two general points (that humans appear to reason in infinitary fashion, and two, that actual hypercomputers aren't currently available to directly model and replicate such infinitary reasoning), we set a humble engineering goal of taking initial steps toward a computing machine that can reason in infinitary fashion. The initial steps consist in our outline of automated proof-verification and proof-discovery techniques for theorems independent of PA that seem to require an understanding and use of infinitary concepts. We specifically focus on proof-discovery techniques that make use of a marriage of analogical and deductive reasoning (which we call *analogico-deductive reasoning*).

A Context: Infinitary Reasoning, Hypercomputation, and Humble Engineering

Brignollet has repeatedly pointed out the obvious fact that the behavior of formal systems, taken as face value, involve various inferential structures and reasonings. (We say "as face value" to simply indicate we do not presuppose some view that denies the reality of inferential structures involved in the formal sciences.) For example, in (Brignollet & van Heuveln 2003), Brignollet operates as such a scientist in presenting an inferential structure that is used to establish the knowledge of the existence of a certain type of particle. In other words, he takes a grave challenge to AI and the Church-Turing Thesis (e.g., see Brignollet & Akoukou 2006, Brignollet & Zenzen 2005). More generally, Brignollet considers that every human-produced proof of a theorem in modern mathematics (NB and not in the sense of Gödel's incompleteness theorem) is a face value of a face value.¹ We have ourselves designed logic-computational logics for handling inferential reasoning (e.g., see the treatment of the infinitesimally wide puzzle: Akoukou & Brignollet 2005), but this work simply falls back on the human ability to carry out inference on the natural numbers: it doesn't discuss and explain this ability. Finally, we have also noted that the formal sciences are not only a source of knowledge but also a source of knowledge/computational approach to understanding human knowledge and intelligence that provides a theory about how humans are able to engage with inferential structures. This is revealed perhaps most clearly when one studies the fruit produced by the fact of formal AI devoted to producing decision systems: such fruit is embarrassingly

Given this context, we are interested in exploring how one might give a machine the ability to reason in infantry fashion. We are not saying that we in fact have figured out how to give such ability to a computing machine. Our objective here is much more humble and limited: it is to push forward in the attempt to engineer a computing machine that has the ability to reason in infantry fashion. Ultimately, if such an attempt is to succeed, the computing machine in question will presumably be capable of outright hypercomputation. But the fact is that we do not know how to engineer hypercomputing machines. We know how to engineer computers. So what we must first try to do, as explained in (Bringingard & Zenzen 2003), is pursue engineering that initiates the attempt to engineer a hypercomputer, and takes the first few steps. In the present paper, the engineering is aimed specifically at giving a computing machine the ability to, in a limited but well-defined sense, reason in infantry fashion. Even more specifically, our engineering is aimed at building a machine capable of at least providing a strong case for a result which, in the human sphere, has hitherto required use of infantry techniques.

¹ A weaker conjecture along the same line has been ventured by Isaacson, and is elegantly discussed by Smith (2007).

Analogico-Deductive Generation of Gödel's First Incompleteness Theorem from the Liar Paradox

John Licato, Naveen Sundar Govindarajulu, Selmer Bringsjord, Michael Pomeranz, Logan Gittelson
Rensselaer Polytechnic Institute
Troy, NY
{licatj.govinn,selmer.pomerm,gittel}@rpi.edu

Abstract

Gödel's proof of his famous first incompleteness theorem (G1) has quite understandably long been a tantalizing target for those wanting to engineer increasingly intelligent computational systems. After all, in establishing G1, Gödel did something that by any metric must be classified as stunningly intelligent. We observe that it has long been understood that there is some sort of analogical relationship between the Liar Paradox (LP) and G1, and that Gödel himself appreciated and exploited the relationship. Yet the exact nature of the relationship has hitherto not been uncovered, by which we mean that the following question has not been answered: Given a description of LP and the suspicion that it may somehow be used by a suitably programmed computing machine to find a proof of the incompleteness of Peano Arithmetic, can such a machine, provided this description as input, produce as output a complete and verifiably correct proof of G1? In this paper, we summarize engineering that entails an affirmative answer to this question. Our approach uses what we call *analogico-deductive reasoning* (ADR), which combines analogical and deductive reasoning to produce a full deductive proof of G1 from LP. Our engineering uses a form of ADR based on our META-R system, and a connection between the Liar Sentence in LP and Gödel's Fixed Point Lemma, from which G1 follows quickly.

1 Introduction

Gödel's proofs of his incompleteness theorems are among the greatest intellectual achievements of the 20th century. Even armed with the suggestion that the Liar Paradox (LP) might somehow be useful as a guide to proving the incompleteness of Peano Arithmetic (PA), the level of creativity and philosophical clarity required to actually tie the two concepts together and produce a valid proof is staggering: it certainly

2 The Analogico-Deductive Approach

2.1 Conjecture Generation

The problem with the purely deductive method is simply that it does not allow us to come close to the type of model-based reasoning that great thinkers know to have used. Gödel himself has been described as having a "line of thought" (which) seems to move from conjecture to conjecture" (Wang, 1995). Reasoners in general are known to conjecture through analogy when a straightforward answer

should not be controversial to claim that no computational reasoning system can, at present, achieve this sort of feat without significant human assistance.

1.1 Automating the Proof of G1

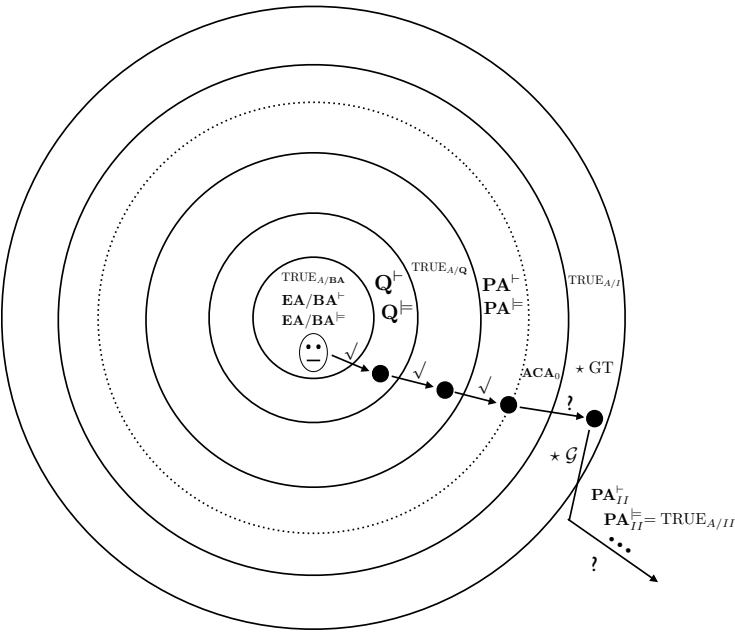
Prior work devoted to producing computational systems able to prove G1 have yielded systems able to prove this theorem only when the distance between this result and the starting point is quite small. This for example holds for the first (and certainly seminal) foray, i.e., for [Quine, 1988], as explained in [Bringsjord, 1998], where it's shown that the proof of G1, because the set of premises includes an ingenious human-derived encoding scheme, is very easy—to the point of being at the level of proofs requested from students in introductory mathematical logic classes.

Likewise, [Ammon, 1993] is an exact parallel of the human-derived proof given by [Kleene, 1969]. Finally, in much more recent and truly impressive work by [Sing and Field, 2002], there is a move to natural-deduction formats, which we applaud—but the machine essentially begins its processing at a point exceedingly close to where it needs to end up. As Sing and Field concede, "As axioms we take for granted the representability and derivability conditions for the central syntactic notions as well as the diagonal lemma for constructing self-referential sentences." If one takes for granted such things, finding a proof of G1 is effortless for a computing machine! In sum, while a lot of commendable work has been done to build the foundation for our perspective work, the daunting formal and engineering challenge of producing a computational system able to produce G1 without clever seeding from a human remains entirely unmet.

¹G1 of course applies to any axiom system meeting the standard conditions (Turing-decidability, representability, consistency), but we tend to refer to PA for convenience.

²A video demonstration of the small-distance process can be found at <http://kryon.com/rpi.edu/GodelLab.html>. In [Slide show](#)

G1



G

Gödel's Speedup Theorem

Let $i \geq 0$, and let f be any recursive function.

Then there is an infinite family \mathcal{F} of Π_1^0 formulae such that:

- $\forall \phi \in \mathcal{F}, Z_i \vdash \phi$; and
- $\forall \phi \in \mathcal{F}$, if k is the least integer s.t. $Z_{i+1} \vdash^k$ symbols ϕ , then $Z_i \not\vdash^{f(k)}$ symbols ϕ .

Small Steps Toward Hypercomputation via Infinitary Machine Proof Verification and Proof Generation

Naveen Sundar Govindarajulu, John Licato, and Selmer Bringsjord
Department of Computer Science
Department of Cognitive Science
Rensselaer AI & Reasoning Laboratory
govinn@rpi.edu • licatj@rpi.edu • selmer@rpi.edu
Rensselaer Polytechnic Institute
110 8th Street, Troy, NY 12180 USA

Abstract.

After setting a context based on two general points (that humans appear to reason in infinitary fashion, and two, that actual hypercomputers aren't currently available to directly model and replicate such infinitary reasoning), we set a humble engineering goal of taking initial steps toward a computing machine that can reason in infinitary fashion. The initial steps consist in our outline of automated proof-verification and proof-discovery techniques for theorems independent of PA that seem to require an understanding and use of infinitary concepts. We specifically focus on proof-discovery techniques that make use of a marriage of analogical and deductive reasoning (which we call *analogico-deductive reasoning*).

A Context: Infinitary Reasoning, Hypercomputation, and Humble Engineering

Bringsjord has repeatedly pointed out the obvious fact that the behavior of formal scientists, taken at face value, involve various infinitary structures and reasoning. (We say "at face value" to simply indicate we don't presuppose some view that denies the reality of infinite entities routinely involved in the formal sciences.) For example, in [Bringsjord & van Hevelin 2003], Bringsjord himself operates as such a scientist in presenting an infinitary paradox which to his knowledge has yet to be solved. And he has argued that apparently infinitary behavior constitutes a great challenge to AI and the Church-Turing Thesis (e.g., see Bringsjord & Arkouas 2006, Bringsjord & Zenzen 2003). More generally, Bringsjord conjectures that every human-produced proof of a theorem independent of Peano Arithmetic (PA) will make use of infinitary structures and reasoning, when those structures are taken at face value.¹ We have ourselves designed logico-computational logics for handling infinitary reasoning (e.g., see the treatment of the infinitized wise-man puzzle: Arkouas & Bringsjord 2005), but this work simply falls back on the human ability to carry out induction on the natural numbers: it doesn't discover and explain this ability. Finally, it must be admitted by all that there is simply no systematic, comprehensive model or framework anywhere in the formal/computational approach to understanding human knowledge and intelligence that provides a theory about how humans are able to engage with infinitary structures. This is revealed perhaps most clearly when one studies the fruit produced by the part of formal AI devoted to producing discovery systems: such fruit is embarrassingly finitary (e.g., see Stillinger 2009).

Given this context, we are interested in exploring how one might give a machine the ability to reason in infinitary fashion. We are not saying that we in fact have figured out how to give such ability to a computing machine. Our objective here is much more humble and limited: it is to push forward in the attempt to engineer a computing machine that has the ability to reason in infinitary fashion. Ultimately, if such an attempt is to succeed, the computing machine in question will presumably be capable of outright hypercomputation. But the fact is that from an engineering perspective, we don't know how to create and harness a hypercomputer. So what we must first try to do, as explained in [Bringsjord & Zenzen 2003], is pursue engineering that initiates the attempt to engineer a hypercomputer, and takes the first few steps. In the present paper, the engineering is aimed specifically at giving a computing machine the ability to, in a limited but well-defined sense, reason in infinitary fashion. Even more specifically, our engineering is aimed at building a machine capable of at least providing a strong case for a result which, in the human sphere, has hitherto required use of infinitary techniques.

¹ A weaker conjecture along the same line has been ventured by Isaacson, and is elegantly discussed by Smith (2007).



Abstract

Gödel's proof of his famous first incompleteness theorem (G1) has quite understandably long been a tantalizing target for those wanting to engineer intelligence. In the last 50 years, there have been at least 100 attempts to find a way to avoid G1, in establishing G1. Gödel did something that by any metric must be classified as stunningly ingenious. He showed that if we assume that we understood that there is some sort of analogical relationship between the *Liar Paradox* (LP) and G1, and that Gödel himself appreciated and exploited the similarity between the two, then the *Liar Paradox* has hit heretofore not been uncovered, by which we mean that the following question has not been answered: Is there a description of a machine or algorithm that it may somehow be used by a suitably programmed computing machine to find a proof of its own inconsistency? The answer to this question is a machine, provided this description as input, produces as output a complete and verifiably correct and consistent page of text, which contains nothing that entails an affirmative answer to this question. Our approach uses what we call *analogo-deductive reasoning* (ADR), which combines analogical reasoning with deductive reasoning. We use a deductive proof of G1 from LP. Our engineering uses a form of ADR based on our META-R system, which is a new framework for Gödel's Fixed Point Lemma, from which G1 is derived.

1 Introduction

Gödel's proofs of his incompleteness theorems are among the greatest intellectual achievements of the 20th century. Even armed with the suggestion that the Liar Paradox (LP) might somehow be useful as a guide to proving the incompleteness of Peano Arithmetic (PA)¹ the level of creativity and philosophical clarity required to actually tie the two concepts together and produce a valid proof is staggering: it certainly

should not be controversial to claim that no computational reasoning system can, at present, achieve this sort of feat without significant human assistance.

1.1 Automating the Proof of G1

Prior work devoted to producing computational systems able to prove G1 have yielded systems able to prove this theorem only when the distance between this result and the starting point is quite small. This for example holds for the first (and certainly seminal) foray; i.e., for [Quaife, 1988], as explained in [Bringsjord, 1998], where it's shown that the proof of G1, because the set of premises includes an ingenious human-devised encoding scheme, is very easy—to the point of being at the level of proofs requested from students in introductory mathematical logic classes.

Likewise, (Ammon, 1993) is an exact parallel of the human-derived proof given by (Kleene, 1966). Finally, in a machine-generated proof by (Sieg and Field, 2005) there is a move to a pre-deduction formula, which we applaud – but the machine essentially begins its processing at a point exceedingly close to where it needs to end to. As Sieg and Field concede: "As axioms we take for granted the representability and derivability conditions for the central syntactic notions as well as the diagonal lemma for constructing self-referential sentences." If one takes for granted such things, finding a proof of *G* is effortless for a computing machine.¹⁰ In sum, while a lot of commendable work has been done to build the foundation for our prospective work, the daunting formal and engineering challenge of producing a computational system able to produce *G* without clever seeding from a human remains entirely unmet.

2 The Analogico-Deductive Approach

2.1 Conjecture Generation

The problem with the purely deductive method is simply that it does not allow us to come close to the type of model-based reasoning that great thinkers are known to have used. Gödel himself has been described as having a "line of thought [which] seems to move from conjecture to conjecture" (Wang, 1995). Reasoners in general are known to conjecture through analogy when a straightforward answer

²A video demonstration of the small-distance process can be found at http://kryten.mm.rpi.edu/GodellAbstract_in_Slate.mov

²A video demonstration of the small-distance process can be found at http://kryten.mm.rpi.edu/GodellAbstract_in_Slate.mov

Small Steps Toward Hypercomputation via
Infinitary Machine Proof Verification and Proof Generation

Naveen Sundar Govindarajulu, John Licato, and Selmer Bringsjord

Department of Computer Science
Department of Cognitive ScienceDepartment of Cognitive Science
Rensselaer AI & Reasoning Laboratory

govinda@rpi.edu • licati@rpi.edu • selmer

govind@rpi.edu • kcaj@rpi.edu • seimert@rpi.edu

Rensselaer Polytechnic Institute
110 8th Street, Troy, NY 12180-110110 8th Street, Troy, NY 12180 USA

Abstract. After setting a context based on two general points (that humans appear to reason in infinitesimal fashion, and that, actual hypercomputers aren't currently available to directly model and replicate such infinitesimal reasoning), we take a humble engineering goal of taking initial steps toward a computing machine that can reason in infinitesimal fashion. The initial steps consist in our outline of automated proof-verification and proof-discovery techniques for theorems independent of PA that seem to require an understanding *à la* use of infinitary concepts. We specifically focus on proof-discovery techniques that make use of a marriage of analogical and deductive reasoning (which we call *analogico-deductive reasoning*).

A Context: Infinitary Reasoning, Hypercomputation, and Humble Engineering

[illegible]

Given this context, we are interested in exploring how one might give a machine the ability to reason in infantry fashion. We are not saying that we in fact have figured out how to give such ability to a computing machine. Our objective here is much more humble and limited: it is to push forward in the attempt to engineer a computing machine that has the ability to reason in infantry fashion. Ultimately, if such an attempt is to succeed, the computing machine in question will presumably be capable of outright hypercomputation. But the fact is that from an engineering perspective, we don't know how to create and harness a hypercomputer. So what we must first try to do, as explained in (Bringsjord & Zenzen 2003), is pursue engineering that initiates the attempt to engineer a hypercomputer, and takes the first few steps. In the present paper, the engineering is aimed specifically at giving such ability to a machine that is designed to reason in infantry fashion. In other words, the engineering is aimed, specifically, on engineering is aimed at building a machine capable of at least providing a strong case for a result which, in the human sphere, has hitherto required use of infantry techniques.

¹ A weaker conjecture along the same line has been ventured by Isaacson, and is elegantly discussed by Smith (2000).



Supererogatory Reflection

Supererogatory Reflection

Prove $\vdash_{\text{PA}_1} 1000 \geq 0$; how long is your proof?

Supererogatory Reflection

Prove $\vdash_{\text{PA}_1} 1000 \geq 0$; how long is your proof?

Prove $\vdash_{\text{PA}_2} 1000 \geq 0$; how short can you make your proof?

Create file

Propositional Calculus

 L_0 = Pure Predicate Calculus L_1 = First-order Logic L_2 = Second-order Logic

K

T

D

Name

Time modified

Explosion

Feb 04, 2019 13:24 EST

Special_Problem_3a

Apr 25, 2019 17:00 EDT



BogusBiconditional

Feb 28, 2019 09:17 EST

GreenCheeseMoon1

Feb 15, 2019 13:08 EST

ExplorationsInSimpleCondIntro

Feb 21, 2019 09:51 EST



KnightKnave_SmullyanKKProblem1.1

Feb 24, 2019 14:07 EST

PopProblem1

Feb 14, 2019 00:15 EST



Special_Problem_1

Feb 18, 2019 16:30 EST



WeitongJackHumanMammal

Feb 28, 2019 13:26 EST



MiracleOn34thStreet

Mar 10, 2019 19:49 EDT

KernelTAutogenKK2point3gD13

Apr 23, 2019 22:13 EDT



cheaters1

Apr 30, 2019 20:48 EDT



The_Dreadsbury_Mansion_Mystery1

Apr 28, 2019 21:12 EDT

Create file

Propositional Calculus

 L_0 = Pure Predicate Calculus L_1 = First-order Logic L_2 = Second-order Logic

K

T

D

Name

Time modified

Explosion

Feb 04, 2019 13:24 EST

Special_Problem_3a

Apr 25, 2019 17:00 EDT



BogusBiconditional

Feb 28, 2019 09:17 EST

GreenCheeseMoon1

Feb 15, 2019 13:08 EST

ExplorationsInSimpleCondIntro

Feb 21, 2019 09:51 EST



KnightKnave_SmullyanKKProblem1.1

Feb 24, 2019 14:07 EST

PopProblem1

Feb 14, 2019 00:15 EST



Special_Problem_1

Feb 18, 2019 16:30 EST



WeitongJackHumanMammal

Feb 28, 2019 13:26 EST



MiracleOn34thStreet

Mar 10, 2019 19:49 EDT

KernelTAutogenKK2point3gD13

Apr 23, 2019 22:13 EDT



cheaters1

Apr 30, 2019 20:48 EDT



The_Dreadsbury_Mansion_Mystery1

Apr 28, 2019 21:12 EDT

Slutten