

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 & Technology
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

Intro to Logic
3/28/2019



FOL

✓ FOL

✓ FOL

Epistemic + FOL
 $B_d B_v B_d V v$

✓ FOL

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

✓ FOL

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

✓ FOL

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

TOL

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

✓ FOL

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

TOL

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



✓ FOL

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

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

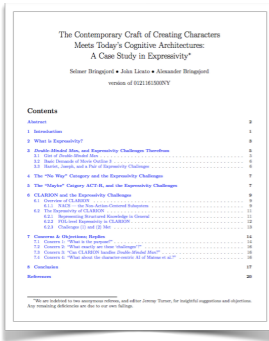
version of 0121161500NY

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 handle <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

A screenshot of a movie outline software interface. The window title is "Movie Outline - Double-Minded_Man_010316.mvo". The interface includes a toolbar with various editing tools, a font selection area (Arial, size 12, Scene Heading, 100%), and a tabbed menu with options like Outline, Script, Notes, Characters, FeelFactor, Reference, Library, PowerView, Step Cards, and Story Tasks. The "Outline" tab is active, showing a list of scenes on the right: 1. TWIRL - DAY, 2. YES, THAT'S HIM - LATER, and 3. SECOND HOME - LATER. The main text area displays the script for scene 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

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

Silver Stoppard • John Lewis • Alexander Stoppard
Journal of Expertise

Contents

Foreword 3

1 Introduction 4

2 What is Expertise? 5

3 Double-Minded Men, and Expertise Challenge Theories 6

3.1 The Expertise Challenge Theories 6

3.2 The Double-Minded Man 6

3.3 The Challenge of the Double-Minded Man 6

4 The "Big Step" Changes and the Expertise Challenge 7

5 The "Big Step" Changes and the Expertise Challenge 7

6 LABOUR and the Expertise Challenge 8

6.1 LABOUR and the Expertise Challenge 8

6.2 The Expertise Challenge and the Expertise Challenge 8

6.3 LABOUR and the Expertise Challenge 8

6.4 LABOUR and the Expertise Challenge 8

7 LABOUR and the Expertise Challenge 9

7.1 LABOUR and the Expertise Challenge 9

7.2 LABOUR and the Expertise Challenge 9

7.3 LABOUR and the Expertise Challenge 9

7.4 LABOUR and the Expertise Challenge 9

8 Conclusions 10

References 10

*This paper is published in the journal "Journal of Expertise" and is available for free download at the website of the "Journal of Expertise".

Double-Minded Man

The Contemporary Craft of Creating Characters
Meets Today's Cognitive Architecture:
A Case Study in Expertise*
Selmer Bringsjord & Alexander Bringsjord
version of 02/16/2016

Contents

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

*This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

Double-Minded Man

Double-Minded Man
by
S Bringsjord & A Bringsjord

DRAFT #5
© June 30 2016

Selmer.Bringsjord@gmail.com

The Contemporary Craft of Creating Characters
Meets Today's Cognitive Architecture:
A Case Study in Expertise*
Silver Stripling • John Linn • Alexander Stripling
ISSN 1524-6011

Contents

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

*This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. For more information, see <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

Double-Minded Man

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

Silver Stripling • John Linn • Alexander Stripling
 volume of 022.161007

Contents

Introduction

1. What is Expertise?

2. Double-Minded Men, and Expertise Challenge Theories

3. The "Two" Minds and the Expertise Challenge

4. The "Two" Minds and the Expertise Challenge

5. The "Two" Minds and the Expertise Challenge

6. The "Two" Minds and the Expertise Challenge

7. The "Two" Minds and the Expertise Challenge

8. The "Two" Minds and the Expertise Challenge

9. The "Two" Minds and the Expertise Challenge

10. The "Two" Minds and the Expertise Challenge

11. The "Two" Minds and the Expertise Challenge

12. The "Two" Minds and the Expertise Challenge

13. The "Two" Minds and the Expertise Challenge

14. The "Two" Minds and the Expertise Challenge

15. The "Two" Minds and the Expertise Challenge

16. The "Two" Minds and the Expertise Challenge

17. The "Two" Minds and the Expertise Challenge

18. The "Two" Minds and the Expertise Challenge

19. The "Two" Minds and the Expertise Challenge

20. The "Two" Minds and the Expertise Challenge

21. The "Two" Minds and the Expertise Challenge

22. The "Two" Minds and the Expertise Challenge

23. The "Two" Minds and the Expertise Challenge

24. The "Two" Minds and the Expertise Challenge

25. The "Two" Minds and the Expertise Challenge

26. The "Two" Minds and the Expertise Challenge

27. The "Two" Minds and the Expertise Challenge

28. The "Two" Minds and the Expertise Challenge

29. The "Two" Minds and the Expertise Challenge

30. The "Two" Minds and the Expertise Challenge

31. The "Two" Minds and the Expertise Challenge

32. The "Two" Minds and the Expertise Challenge

33. The "Two" Minds and the Expertise Challenge

34. The "Two" Minds and the Expertise Challenge

35. The "Two" Minds and the Expertise Challenge

36. The "Two" Minds and the Expertise Challenge

37. The "Two" Minds and the Expertise Challenge

38. The "Two" Minds and the Expertise Challenge

39. The "Two" Minds and the Expertise Challenge

40. The "Two" Minds and the Expertise Challenge

41. The "Two" Minds and the Expertise Challenge

42. The "Two" Minds and the Expertise Challenge

43. The "Two" Minds and the Expertise Challenge

44. The "Two" Minds and the Expertise Challenge

45. The "Two" Minds and the Expertise Challenge

46. The "Two" Minds and the Expertise Challenge

47. The "Two" Minds and the Expertise Challenge

48. The "Two" Minds and the Expertise Challenge

49. The "Two" Minds and the Expertise Challenge

50. The "Two" Minds and the Expertise Challenge

51. The "Two" Minds and the Expertise Challenge

52. The "Two" Minds and the Expertise Challenge

53. The "Two" Minds and the Expertise Challenge

54. The "Two" Minds and the Expertise Challenge

55. The "Two" Minds and the Expertise Challenge

56. The "Two" Minds and the Expertise Challenge

57. The "Two" Minds and the Expertise Challenge

58. The "Two" Minds and the Expertise Challenge

59. The "Two" Minds and the Expertise Challenge

60. The "Two" Minds and the Expertise Challenge

61. The "Two" Minds and the Expertise Challenge

62. The "Two" Minds and the Expertise Challenge

63. The "Two" Minds and the Expertise Challenge

64. The "Two" Minds and the Expertise Challenge

65. The "Two" Minds and the Expertise Challenge

66. The "Two" Minds and the Expertise Challenge

67. The "Two" Minds and the Expertise Challenge

68. The "Two" Minds and the Expertise Challenge

69. The "Two" Minds and the Expertise Challenge

70. The "Two" Minds and the Expertise Challenge

71. The "Two" Minds and the Expertise Challenge

72. The "Two" Minds and the Expertise Challenge

73. The "Two" Minds and the Expertise Challenge

74. The "Two" Minds and the Expertise Challenge

75. The "Two" Minds and the Expertise Challenge

76. The "Two" Minds and the Expertise Challenge

77. The "Two" Minds and the Expertise Challenge

78. The "Two" Minds and the Expertise Challenge

79. The "Two" Minds and the Expertise Challenge

80. The "Two" Minds and the Expertise Challenge

81. The "Two" Minds and the Expertise Challenge

82. The "Two" Minds and the Expertise Challenge

83. The "Two" Minds and the Expertise Challenge

84. The "Two" Minds and the Expertise Challenge

85. The "Two" Minds and the Expertise Challenge

86. The "Two" Minds and the Expertise Challenge

87. The "Two" Minds and the Expertise Challenge

88. The "Two" Minds and the Expertise Challenge

89. The "Two" Minds and the Expertise Challenge

90. The "Two" Minds and the Expertise Challenge

91. The "Two" Minds and the Expertise Challenge

92. The "Two" Minds and the Expertise Challenge

93. The "Two" Minds and the Expertise Challenge

94. The "Two" Minds and the Expertise Challenge

95. The "Two" Minds and the Expertise Challenge

96. The "Two" Minds and the Expertise Challenge

97. The "Two" Minds and the Expertise Challenge

98. The "Two" Minds and the Expertise Challenge

99. The "Two" Minds and the Expertise Challenge

100. The "Two" Minds and the Expertise Challenge

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.

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

Silver Stripling • John Linn • Alexander Stripling
Volume of EXPERTISE

Contents

Introduction	1
What is Expertise?	1
Double-Minded Men, and Expertise Challenge Theories	1
1.1 The Double-Minded Man	1
1.2 The Double-Minded Man and Expertise Challenge Theories	1
1.3 The Double-Minded Man and Expertise Challenge Theories	1
1.4 The Double-Minded Man and Expertise Challenge Theories	1
1.5 The Double-Minded Man and Expertise Challenge Theories	1
1.6 The Double-Minded Man and Expertise Challenge Theories	1
1.7 The Double-Minded Man and Expertise Challenge Theories	1
1.8 The Double-Minded Man and Expertise Challenge Theories	1
1.9 The Double-Minded Man and Expertise Challenge Theories	1
1.10 The Double-Minded Man and Expertise Challenge Theories	1
1.11 The Double-Minded Man and Expertise Challenge Theories	1
1.12 The Double-Minded Man and Expertise Challenge Theories	1
1.13 The Double-Minded Man and Expertise Challenge Theories	1
1.14 The Double-Minded Man and Expertise Challenge Theories	1
1.15 The Double-Minded Man and Expertise Challenge Theories	1
1.16 The Double-Minded Man and Expertise Challenge Theories	1
1.17 The Double-Minded Man and Expertise Challenge Theories	1
1.18 The Double-Minded Man and Expertise Challenge Theories	1
1.19 The Double-Minded Man and Expertise Challenge Theories	1
1.20 The Double-Minded Man and Expertise Challenge Theories	1
1.21 The Double-Minded Man and Expertise Challenge Theories	1
1.22 The Double-Minded Man and Expertise Challenge Theories	1
1.23 The Double-Minded Man and Expertise Challenge Theories	1
1.24 The Double-Minded Man and Expertise Challenge Theories	1
1.25 The Double-Minded Man and Expertise Challenge Theories	1
1.26 The Double-Minded Man and Expertise Challenge Theories	1
1.27 The Double-Minded Man and Expertise Challenge Theories	1
1.28 The Double-Minded Man and Expertise Challenge Theories	1
1.29 The Double-Minded Man and Expertise Challenge Theories	1
1.30 The Double-Minded Man and Expertise Challenge Theories	1
1.31 The Double-Minded Man and Expertise Challenge Theories	1
1.32 The Double-Minded Man and Expertise Challenge Theories	1
1.33 The Double-Minded Man and Expertise Challenge Theories	1
1.34 The Double-Minded Man and Expertise Challenge Theories	1
1.35 The Double-Minded Man and Expertise Challenge Theories	1
1.36 The Double-Minded Man and Expertise Challenge Theories	1
1.37 The Double-Minded Man and Expertise Challenge Theories	1
1.38 The Double-Minded Man and Expertise Challenge Theories	1
1.39 The Double-Minded Man and Expertise Challenge Theories	1
1.40 The Double-Minded Man and Expertise Challenge Theories	1
1.41 The Double-Minded Man and Expertise Challenge Theories	1
1.42 The Double-Minded Man and Expertise Challenge Theories	1
1.43 The Double-Minded Man and Expertise Challenge Theories	1
1.44 The Double-Minded Man and Expertise Challenge Theories	1
1.45 The Double-Minded Man and Expertise Challenge Theories	1
1.46 The Double-Minded Man and Expertise Challenge Theories	1
1.47 The Double-Minded Man and Expertise Challenge Theories	1
1.48 The Double-Minded Man and Expertise Challenge Theories	1
1.49 The Double-Minded Man and Expertise Challenge Theories	1
1.50 The Double-Minded Man and Expertise Challenge Theories	1
1.51 The Double-Minded Man and Expertise Challenge Theories	1
1.52 The Double-Minded Man and Expertise Challenge Theories	1
1.53 The Double-Minded Man and Expertise Challenge Theories	1
1.54 The Double-Minded Man and Expertise Challenge Theories	1
1.55 The Double-Minded Man and Expertise Challenge Theories	1
1.56 The Double-Minded Man and Expertise Challenge Theories	1
1.57 The Double-Minded Man and Expertise Challenge Theories	1
1.58 The Double-Minded Man and Expertise Challenge Theories	1
1.59 The Double-Minded Man and Expertise Challenge Theories	1
1.60 The Double-Minded Man and Expertise Challenge Theories	1
1.61 The Double-Minded Man and Expertise Challenge Theories	1
1.62 The Double-Minded Man and Expertise Challenge Theories	1
1.63 The Double-Minded Man and Expertise Challenge Theories	1
1.64 The Double-Minded Man and Expertise Challenge Theories	1
1.65 The Double-Minded Man and Expertise Challenge Theories	1
1.66 The Double-Minded Man and Expertise Challenge Theories	1
1.67 The Double-Minded Man and Expertise Challenge Theories	1
1.68 The Double-Minded Man and Expertise Challenge Theories	1
1.69 The Double-Minded Man and Expertise Challenge Theories	1
1.70 The Double-Minded Man and Expertise Challenge Theories	1
1.71 The Double-Minded Man and Expertise Challenge Theories	1
1.72 The Double-Minded Man and Expertise Challenge Theories	1
1.73 The Double-Minded Man and Expertise Challenge Theories	1
1.74 The Double-Minded Man and Expertise Challenge Theories	1
1.75 The Double-Minded Man and Expertise Challenge Theories	1
1.76 The Double-Minded Man and Expertise Challenge Theories	1
1.77 The Double-Minded Man and Expertise Challenge Theories	1
1.78 The Double-Minded Man and Expertise Challenge Theories	1
1.79 The Double-Minded Man and Expertise Challenge Theories	1
1.80 The Double-Minded Man and Expertise Challenge Theories	1
1.81 The Double-Minded Man and Expertise Challenge Theories	1
1.82 The Double-Minded Man and Expertise Challenge Theories	1
1.83 The Double-Minded Man and Expertise Challenge Theories	1
1.84 The Double-Minded Man and Expertise Challenge Theories	1
1.85 The Double-Minded Man and Expertise Challenge Theories	1
1.86 The Double-Minded Man and Expertise Challenge Theories	1
1.87 The Double-Minded Man and Expertise Challenge Theories	1
1.88 The Double-Minded Man and Expertise Challenge Theories	1
1.89 The Double-Minded Man and Expertise Challenge Theories	1
1.90 The Double-Minded Man and Expertise Challenge Theories	1
1.91 The Double-Minded Man and Expertise Challenge Theories	1
1.92 The Double-Minded Man and Expertise Challenge Theories	1
1.93 The Double-Minded Man and Expertise Challenge Theories	1
1.94 The Double-Minded Man and Expertise Challenge Theories	1
1.95 The Double-Minded Man and Expertise Challenge Theories	1
1.96 The Double-Minded Man and Expertise Challenge Theories	1
1.97 The Double-Minded Man and Expertise Challenge Theories	1
1.98 The Double-Minded Man and Expertise Challenge Theories	1
1.99 The Double-Minded Man and Expertise Challenge Theories	1
1.100 The Double-Minded Man and Expertise Challenge Theories	1

*This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

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.

The Contemporary Craft of Creating Characters
 Meets Today's Cognitive Architecture:
 A Case Study in Expertise*
 Silver Stripling • John Linn • Alexander Stripling
 volume of EXPERTISE

Contents

- 1 Introduction
- 2 What is Expertise?
- 3 Double-Mindedness and Expertise: Challenge Theories
- 4 The "Two-Step" Challenge and the Expertise Challenge
- 5 The "Single-Step" Challenge and the Expertise Challenge
- 6 CLARION and the Expertise Challenge
- 7 The "Two-Step" Challenge and the Expertise Challenge
- 8 The "Single-Step" Challenge and the Expertise Challenge
- 9 Challenge (1) and (2)
- 10 Challenge (3) and (4)
- 11 Conclusion
- 12 Appendix
- 13 Acknowledgments
- 14 References

*This work is licensed under a Creative Commons Attribution 4.0 International License. For more information, see <http://creativecommons.org/licenses/by/4.0/>.

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.

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

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

Silver Stroup† • John Lewis • Alexander Stroup‡
 volume of *COGNITIVE*

Contents

1 Introduction

2 What is Narrative?

3 Psychological Ills and Epistemic Challenge: Theories
 3.1 Psychological Ills
 3.2 Epistemic Challenge

4 The "Big" Concept and the Epistemic Challenge
 4.1 The "Big" Concept
 4.2 The Epistemic Challenge

5 *Character* and the Epistemic Challenge
 5.1 *Character*
 5.2 The Epistemic Challenge

6 *Character* and the Epistemic Challenge
 6.1 *Character*
 6.2 The Epistemic Challenge

7 *Character* and the Epistemic Challenge
 7.1 *Character*
 7.2 The Epistemic Challenge

8 *Character* and the Epistemic Challenge
 8.1 *Character*
 8.2 The Epistemic Challenge

9 *Character* and the Epistemic Challenge
 9.1 *Character*
 9.2 The Epistemic Challenge

10 *Character* and the Epistemic Challenge
 10.1 *Character*
 10.2 The Epistemic Challenge

11 *Character* and the Epistemic Challenge
 11.1 *Character*
 11.2 The Epistemic Challenge

12 *Character* and the Epistemic Challenge
 12.1 *Character*
 12.2 The Epistemic Challenge

13 *Character* and the Epistemic Challenge
 13.1 *Character*
 13.2 The Epistemic Challenge

14 *Character* and the Epistemic Challenge
 14.1 *Character*
 14.2 The Epistemic Challenge

15 *Character* and the Epistemic Challenge
 15.1 *Character*
 15.2 The Epistemic Challenge

16 *Character* and the Epistemic Challenge
 16.1 *Character*
 16.2 The Epistemic Challenge

17 *Character* and the Epistemic Challenge
 17.1 *Character*
 17.2 The Epistemic Challenge

18 *Character* and the Epistemic Challenge
 18.1 *Character*
 18.2 The Epistemic Challenge

19 *Character* and the Epistemic Challenge
 19.1 *Character*
 19.2 The Epistemic Challenge

20 *Character* and the Epistemic Challenge
 20.1 *Character*
 20.2 The Epistemic Challenge

21 *Character* and the Epistemic Challenge
 21.1 *Character*
 21.2 The Epistemic Challenge

22 *Character* and the Epistemic Challenge
 22.1 *Character*
 22.2 The Epistemic Challenge

23 *Character* and the Epistemic Challenge
 23.1 *Character*
 23.2 The Epistemic Challenge

24 *Character* and the Epistemic Challenge
 24.1 *Character*
 24.2 The Epistemic Challenge

25 *Character* and the Epistemic Challenge
 25.1 *Character*
 25.2 The Epistemic Challenge

26 *Character* and the Epistemic Challenge
 26.1 *Character*
 26.2 The Epistemic Challenge

27 *Character* and the Epistemic Challenge
 27.1 *Character*
 27.2 The Epistemic Challenge

28 *Character* and the Epistemic Challenge
 28.1 *Character*
 28.2 The Epistemic Challenge

29 *Character* and the Epistemic Challenge
 29.1 *Character*
 29.2 The Epistemic Challenge

30 *Character* and the Epistemic Challenge
 30.1 *Character*
 30.2 The Epistemic Challenge

31 *Character* and the Epistemic Challenge
 31.1 *Character*
 31.2 The Epistemic Challenge

32 *Character* and the Epistemic Challenge
 32.1 *Character*
 32.2 The Epistemic Challenge

33 *Character* and the Epistemic Challenge
 33.1 *Character*
 33.2 The Epistemic Challenge

34 *Character* and the Epistemic Challenge
 34.1 *Character*
 34.2 The Epistemic Challenge

35 *Character* and the Epistemic Challenge
 35.1 *Character*
 35.2 The Epistemic Challenge

36 *Character* and the Epistemic Challenge
 36.1 *Character*
 36.2 The Epistemic Challenge

37 *Character* and the Epistemic Challenge
 37.1 *Character*
 37.2 The Epistemic Challenge

38 *Character* and the Epistemic Challenge
 38.1 *Character*
 38.2 The Epistemic Challenge

39 *Character* and the Epistemic Challenge
 39.1 *Character*
 39.2 The Epistemic Challenge

40 *Character* and the Epistemic Challenge
 40.1 *Character*
 40.2 The Epistemic Challenge

41 *Character* and the Epistemic Challenge
 41.1 *Character*
 41.2 The Epistemic Challenge

42 *Character* and the Epistemic Challenge
 42.1 *Character*
 42.2 The Epistemic Challenge

43 *Character* and the Epistemic Challenge
 43.1 *Character*
 43.2 The Epistemic Challenge

44 *Character* and the Epistemic Challenge
 44.1 *Character*
 44.2 The Epistemic Challenge

45 *Character* and the Epistemic Challenge
 45.1 *Character*
 45.2 The Epistemic Challenge

46 *Character* and the Epistemic Challenge
 46.1 *Character*
 46.2 The Epistemic Challenge

47 *Character* and the Epistemic Challenge
 47.1 *Character*
 47.2 The Epistemic Challenge

48 *Character* and the Epistemic Challenge
 48.1 *Character*
 48.2 The Epistemic Challenge

49 *Character* and the Epistemic Challenge
 49.1 *Character*
 49.2 The Epistemic Challenge

50 *Character* and the Epistemic Challenge
 50.1 *Character*
 50.2 The Epistemic Challenge

51 *Character* and the Epistemic Challenge
 51.1 *Character*
 51.2 The Epistemic Challenge

52 *Character* and the Epistemic Challenge
 52.1 *Character*
 52.2 The Epistemic Challenge

53 *Character* and the Epistemic Challenge
 53.1 *Character*
 53.2 The Epistemic Challenge

54 *Character* and the Epistemic Challenge
 54.1 *Character*
 54.2 The Epistemic Challenge

55 *Character* and the Epistemic Challenge
 55.1 *Character*
 55.2 The Epistemic Challenge

56 *Character* and the Epistemic Challenge
 56.1 *Character*
 56.2 The Epistemic Challenge

57 *Character* and the Epistemic Challenge
 57.1 *Character*
 57.2 The Epistemic Challenge

58 *Character* and the Epistemic Challenge
 58.1 *Character*
 58.2 The Epistemic Challenge

59 *Character* and the Epistemic Challenge
 59.1 *Character*
 59.2 The Epistemic Challenge

60 *Character* and the Epistemic Challenge
 60.1 *Character*
 60.2 The Epistemic Challenge

61 *Character* and the Epistemic Challenge
 61.1 *Character*
 61.2 The Epistemic Challenge

62 *Character* and the Epistemic Challenge
 62.1 *Character*
 62.2 The Epistemic Challenge

63 *Character* and the Epistemic Challenge
 63.1 *Character*
 63.2 The Epistemic Challenge

64 *Character* and the Epistemic Challenge
 64.1 *Character*
 64.2 The Epistemic Challenge

65 *Character* and the Epistemic Challenge
 65.1 *Character*
 65.2 The Epistemic Challenge

66 *Character* and the Epistemic Challenge
 66.1 *Character*
 66.2 The Epistemic Challenge

67 *Character* and the Epistemic Challenge
 67.1 *Character*
 67.2 The Epistemic Challenge

68 *Character* and the Epistemic Challenge
 68.1 *Character*
 68.2 The Epistemic Challenge

69 *Character* and the Epistemic Challenge
 69.1 *Character*
 69.2 The Epistemic Challenge

70 *Character* and the Epistemic Challenge
 70.1 *Character*
 70.2 The Epistemic Challenge

71 *Character* and the Epistemic Challenge
 71.1 *Character*
 71.2 The Epistemic Challenge

72 *Character* and the Epistemic Challenge
 72.1 *Character*
 72.2 The Epistemic Challenge

73 *Character* and the Epistemic Challenge
 73.1 *Character*
 73.2 The Epistemic Challenge

74 *Character* and the Epistemic Challenge
 74.1 *Character*
 74.2 The Epistemic Challenge

75 *Character* and the Epistemic Challenge
 75.1 *Character*
 75.2 The Epistemic Challenge

76 *Character* and the Epistemic Challenge
 76.1 *Character*
 76.2 The Epistemic Challenge

77 *Character* and the Epistemic Challenge
 77.1 *Character*
 77.2 The Epistemic Challenge

78 *Character* and the Epistemic Challenge
 78.1 *Character*
 78.2 The Epistemic Challenge

79 *Character* and the Epistemic Challenge
 79.1 *Character*
 79.2 The Epistemic Challenge

80 *Character* and the Epistemic Challenge
 80.1 *Character*
 80.2 The Epistemic Challenge

81 *Character* and the Epistemic Challenge
 81.1 *Character*
 81.2 The Epistemic Challenge

82 *Character* and the Epistemic Challenge
 82.1 *Character*
 82.2 The Epistemic Challenge

83 *Character* and the Epistemic Challenge
 83.1 *Character*
 83.2 The Epistemic Challenge

84 *Character* and the Epistemic Challenge
 84.1 *Character*
 84.2 The Epistemic Challenge

85 *Character* and the Epistemic Challenge
 85.1 *Character*
 85.2 The Epistemic Challenge

86 *Character* and the Epistemic Challenge
 86.1 *Character*
 86.2 The Epistemic Challenge

87 *Character* and the Epistemic Challenge
 87.1 *Character*
 87.2 The Epistemic Challenge

88 *Character* and the Epistemic Challenge
 88.1 *Character*
 88.2 The Epistemic Challenge

89 *Character* and the Epistemic Challenge
 89.1 *Character*
 89.2 The Epistemic Challenge

90 *Character* and the Epistemic Challenge
 90.1 *Character*
 90.2 The Epistemic Challenge

91 *Character* and the Epistemic Challenge
 91.1 *Character*
 91.2 The Epistemic Challenge

92 *Character* and the Epistemic Challenge
 92.1 *Character*
 92.2 The Epistemic Challenge

93 *Character* and the Epistemic Challenge
 93.1 *Character*
 93.2 The Epistemic Challenge

94 *Character* and the Epistemic Challenge
 94.1 *Character*
 94.2 The Epistemic Challenge

95 *Character* and the Epistemic Challenge
 95.1 *Character*
 95.2 The Epistemic Challenge

96 *Character* and the Epistemic Challenge
 96.1 *Character*
 96.2 The Epistemic Challenge

97 *Character* and the Epistemic Challenge
 97.1 *Character*
 97.2 The Epistemic Challenge

98 *Character* and the Epistemic Challenge
 98.1 *Character*
 98.2 The Epistemic Challenge

99 *Character* and the Epistemic Challenge
 99.1 *Character*
 99.2 The Epistemic Challenge

100 *Character* and the Epistemic Challenge
 100.1 *Character*
 100.2 The Epistemic Challenge

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.



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

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^2 s 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

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

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

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

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

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

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

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

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

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

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

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

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

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.





“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

$$\mathbf{PA} = \mathbf{Z}_1$$

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

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

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

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

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

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

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

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.

$$\text{PA} = \text{Z}_1$$

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

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

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

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

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

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

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

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.

New!

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

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

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

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

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

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

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

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

New!

PA = Z₂

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

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

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

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

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

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

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

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

New!

PA = Z_2

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

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

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

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

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

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

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

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

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

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

John Licato, Naveen Sundar Govindarajulu, Selmer Bringsjord, Michael Pomeranz, Logan Gittensohn
Rensselaer Polytechnic Institute

Troy, NY

{licatj.govin,selmer.pomeranz,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- Σ_1 -recursiveness, 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's (1963), as explained in [Bringsjord, 1998], where it's shown that the proof of G1, because the set of premises includes an ingenious human-designed 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 (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.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 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://royen.sims.rpi.edu/GodelAbstract.in.Slate.mov>

G I

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

govin@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 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 great challenge to AI and the Church-Turing Thesis (e.g., see Bringsjord & Arkoufas 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: Arkoufas & 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 Skidmore 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).

G

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

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

{licatj.govinn,selmer.pomeranz,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's [1983], 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—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 [1969]. Finally, in much more recent and truly impressive work by Sieg 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 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.¹ 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.

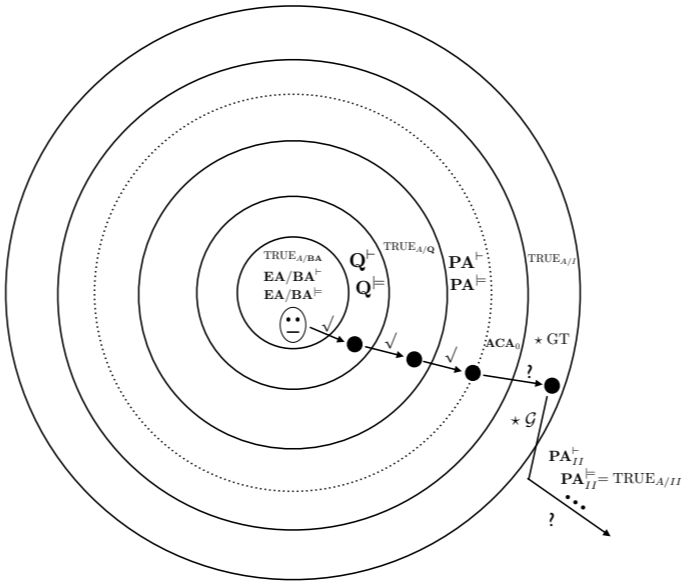
2 The Analogico-Deductive Approach

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

¹A vivid demonstration of the small-distance process can be found at http://royen.usm.rpi.edu/Godel/abstract_in_Siam.pdf

G I



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 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 great challenge to AI and the Church-Turing Thesis (e.g., see Bringsjord & Arkoulias 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: Arkoulias & 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 Skildris 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 Pomeroy, Logan Gittel
Rensselaer Polytechnic Institute
Troy, NY

{licatj.govinn,selmer.pomeroy,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 computer reasoning system can, at present, achieve this sort of 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 certain 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.

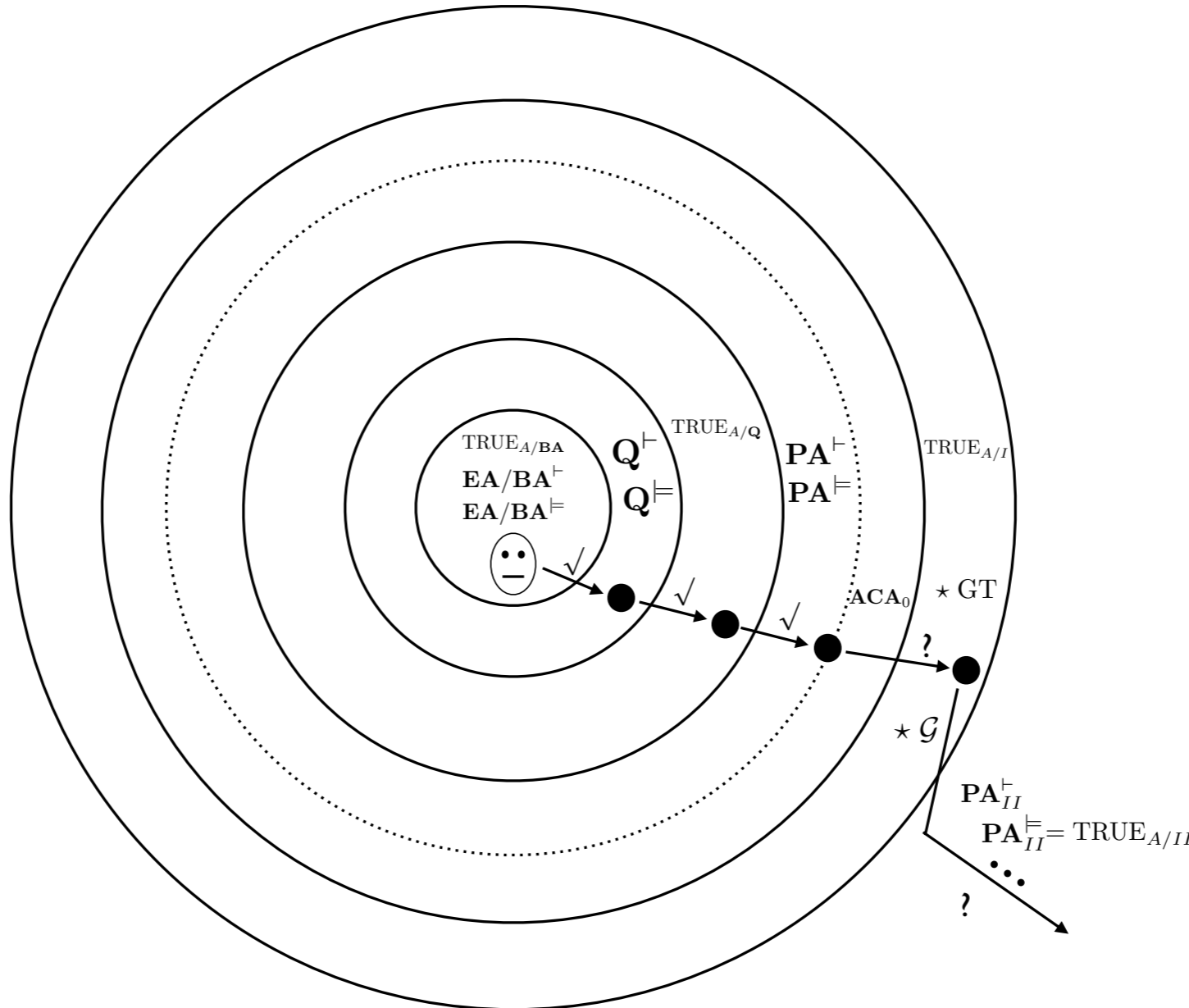
Likewise, [Ammon, 1993] is an exact parallel of the human derived proof given by Kleene, 1969. Finally, in our more recent and truly impressive work by [Sing and Fur 2002], there is a move to natural-deduction formalisms, which we upland—but the machine essentially begins its proceeding at a point exceedingly close to where it needs to end: An Sieg and Field consider: "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 was the daunting formal and engineering challenge of producing a computational system able to produce G1 without clemency from a human remains entirely unmet.

2 The Analogico-Deductive Approach

2.1 Conjecture Generation

The problem with the purely deductive method is simple: 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 "15 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://beyon.com/rpi.edu/GodelAbstract.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 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 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 correlates 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 *wiseman 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 Skidmore 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 Gittensohn
Rensselaer Polytechnic Institute
Troy, NY

{licatj.govinn.selmer.pomeranz.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's (1983), as explained in (Bringsjord, 1998), where it's shown that the proof of G1, because the set of premises includes an ingenious human-designed 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, 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.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.

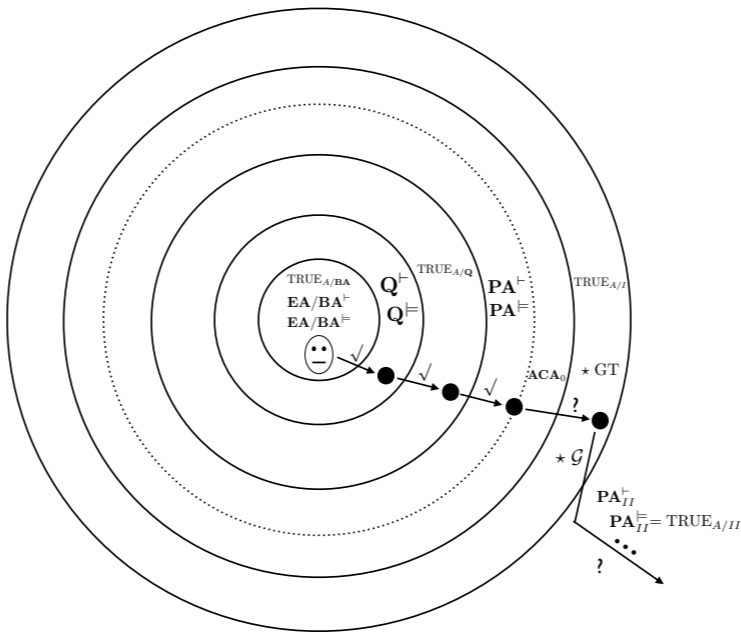
2 The Analogico-Deductive Approach

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 vivid demonstration of the small-distance process can be found at http://royan.stm.rpi.edu/Godel/abstract_in_Siam.pdf

G I



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 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 great challenge to AI and the Church-Turing Thesis (e.g., see Bringsjord & Arkoulias 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: Arkoulias & 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 Skildrae 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 Pomeroy, Logan Gittensohn
Rensselaer Polytechnic Institute
Troy, NY
{licatj.govinn.selmer.pomeroy.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

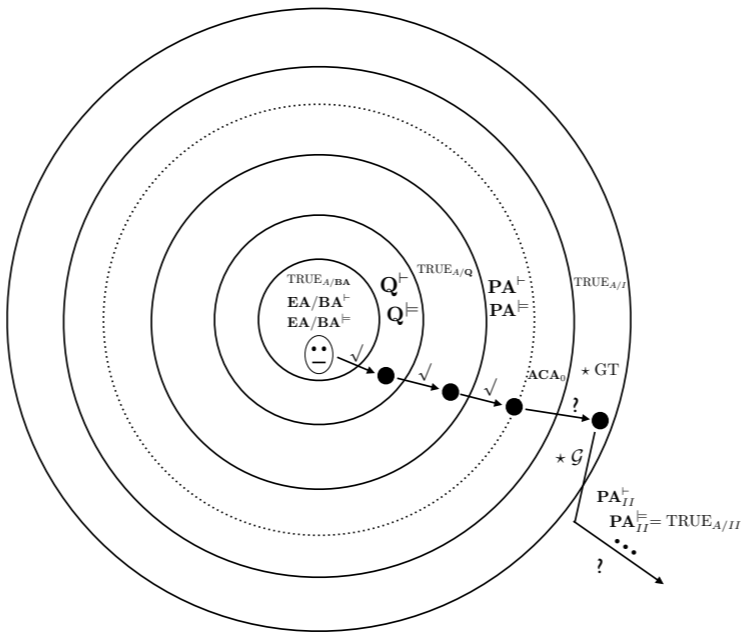
Prior work devoted to producing computational systems able to prove G1 have yielded systems able to prove this theorem only when the distance between this result and the starting point is quite small. This for example holds for the first (and certainly seminal) foray, i.e., for Quine's (1983), as explained 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—in the point of being at the level of proofs requested from students in introductory mathematical logic classes.

2 The Analogico-Deductive Approach

Like-wise, (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.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.

²A vivid demonstration of the small-distance process can be found at http://royen.stm.rpi.edu/Godel/abstract_in_Siam.pdf

G I



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 outlining 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 great challenge to AI and the Church-Turing Thesis (e.g., see Bringsjord & Arkoulias 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: Arkoulias & 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 Skildrae 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).

Gödel's Speedup Theorem

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

John Licato, Naveen Sundar Govindarajulu, Selmer Bringsjord, Michael Pomeroy, Logan Gittensohn
Rensselaer Polytechnic Institute
Troy, NY
{licatj.govin,selmer.pomeroy,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 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's (1983), 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—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, 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.¹ 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.

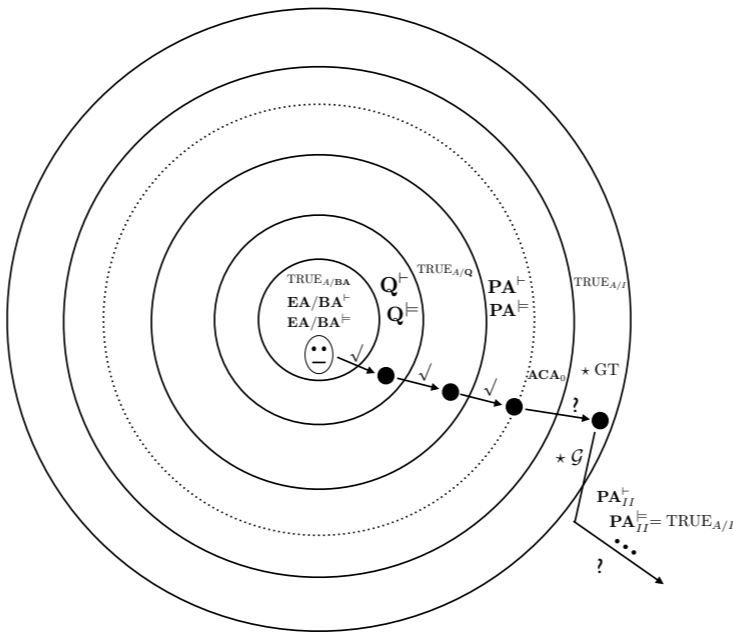
2 The Analogico-Deductive Approach

2.1 Conjecture Generation

The problem with the purely deductive method is 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

¹A vivid demonstration of the small-distance process can be found at http://royan.msu.rpi.edu/Godel/abstract_in_Siam.pdf

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
govin@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 outlining 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 great challenge to AI and the Church-Turing Thesis (e.g., see Bringsjord & Arkovos 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 *wiseman* puzzle: Arkovos & 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 Skidmore 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).

Gödel's Speedup Theorem

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

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

John Licato, Naveen Sundar Govindarajulu, Selmer Bringsjord, Michael Pomeranz, Logan Gittensohn
Rensselaer Polytechnic Institute
Troy, NY
{licatj.govinn,selmer.pomeranz,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 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

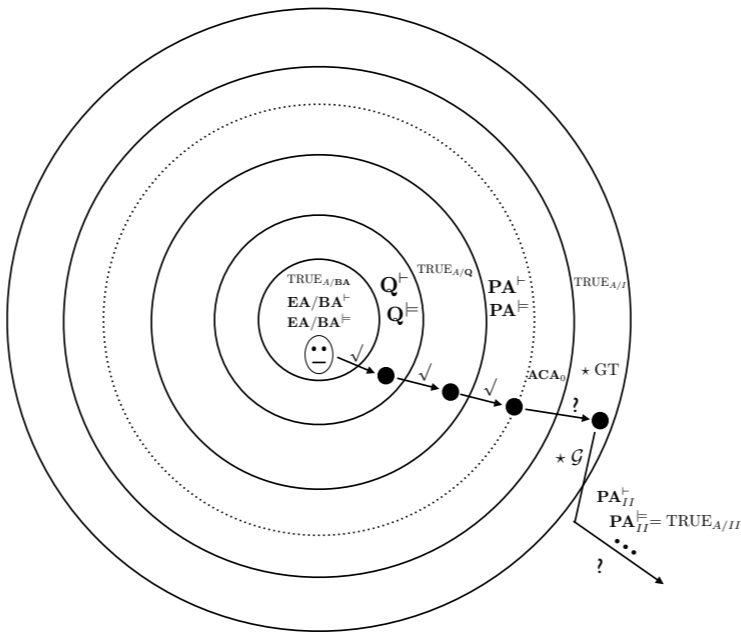
Prior work devoted to producing computational systems able to prove G1 have yielded systems able to prove this theorem only when the distance between this result and the starting point is quite small. This for example holds for the first (and certainly seminal) foray, i.e., for [Quine, 1968], as explained 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—in the point of being at the level of proofs requested from students in introductory mathematical logic classes.

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 vivid demonstration of the small-distance process can be found at http://royan.stm.rpi.edu/Godel/abstract_in_Siam.pdf

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 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 outlining 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 great challenge to AI and the Church-Turing Thesis (e.g., see Bringsjord & Arkovos 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 *wiseman* puzzle: Arkovos & 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 Skildrae 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).

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:

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

John Licato, Naveen Sundar Govindarajulu, Selmer Bringsjord, Michael Pomeranz, Logan Gittensohn
Rensselaer Polytechnic Institute
Troy, NY
{licatj.govin,selmer.pomeranz,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 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's (1983), as explained in [Bringsjord, 1998], where it's shown that the proof of G1, because the set of premises includes an ingenious human-designed 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 (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.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.

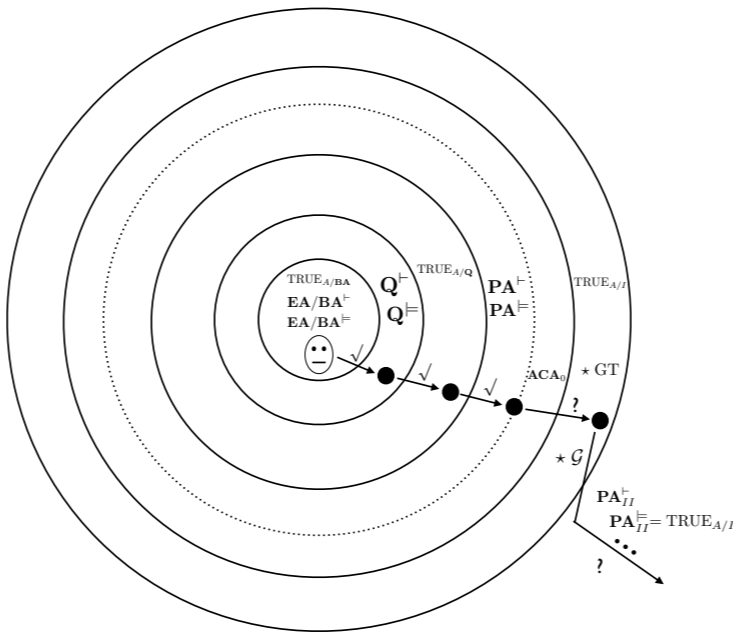
2 The Analogico-Deductive Approach

2.1 Conjecture Generation

The problem with the purely deductive method is simple: 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 vivid demonstration of the small-distance process can be found at http://royan.stm.rpi.edu/Godel/abstract_in_Siam.pdf

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
govin@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 outlining 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 great challenge to AI and the Church-Turing Thesis (e.g., see Bringsjord & Arkovos 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 *wiseman* puzzle: Arkovos & 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 Shilladay 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).

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:



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

John Licato, Naveen Sundar Govindarajulu, Selmer Bringsjord, Michael Pomeranz, Logan Gittensohn
Rensselaer Polytechnic Institute
Troy, NY
{licatj.govinn,selmer.pomeranz,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 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's (1983), as explained in (Bringsjord, 1998), where it's shown that the proof of G1, because the set of premises includes an ingenious human-designed 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 (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.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.

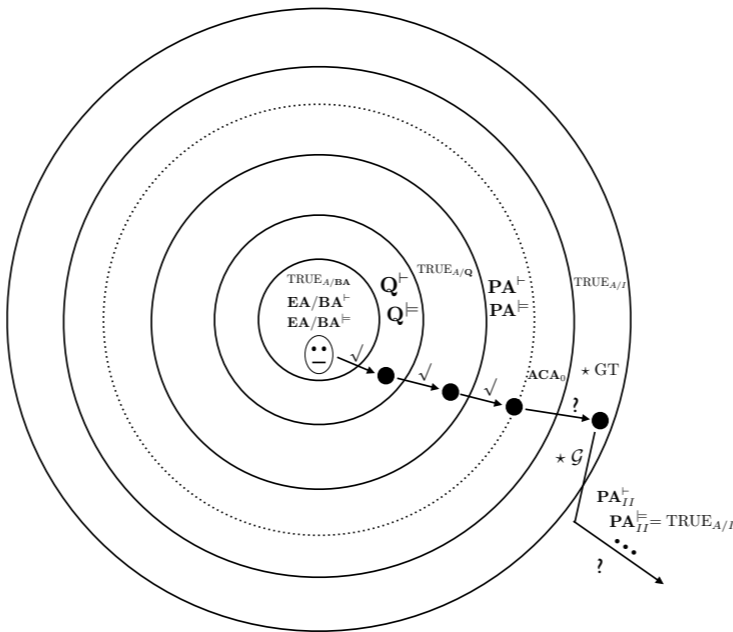
2 The Analogico-Deductive Approach

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 vivid demonstration of the small-distance process can be found at http://royan.stim.rpi.edu/Godel/abstract_in_Siam.pdf

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 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 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 as 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 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 correlates a great challenge to AI and the Church-Turing Thesis (e.g., see Bringsjord & Arkovos 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 *wiseman* puzzle: Arkovos & 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 Shilladay 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).

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 .



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

John Licato, Naveen Sundar Govindarajulu, Selmer Bringsjord, Michael Pomeranz, Logan Gittensohn
Rensselaer Polytechnic Institute
Troy, NY
{licatj.govinn,selmer.pomeranz,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.

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's (1963), as explained in (Bringsjord, 1998), where it's shown that the proof of G1, because the set of premises includes an ingenious human-designed encoding scheme, is very easy—in the point of being at the level of proofs requested from students in introductory mathematical logic classes.

Like-wise, (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.¹ 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.

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),² the level of creativity and philosophical clarity required to actually tie the two concepts together and produce a valid proof is staggering: it certainly

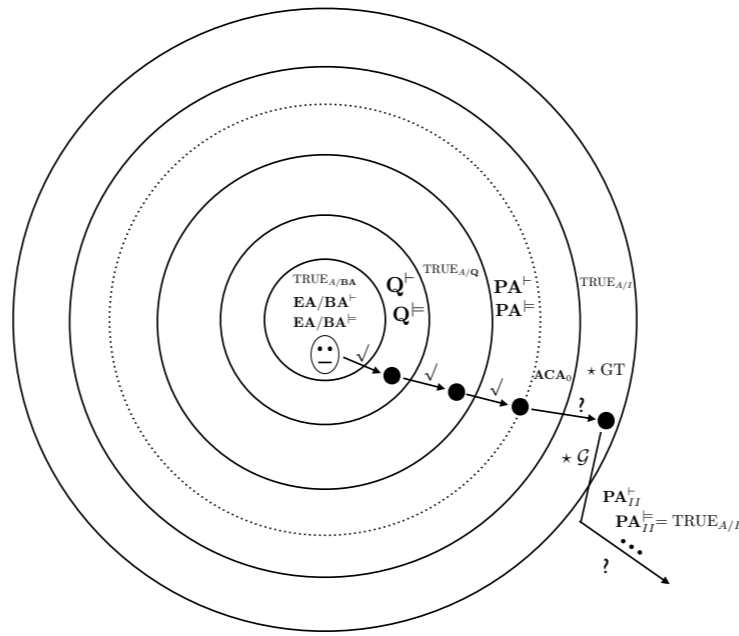
2 The Analogico-Deductive Approach

2.1 Conjecture Generation
The problem with the purely deductive method is 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 vivid demonstration of the small-distance process can be found at http://royan.ams.rpi.edu/Godel/abstract_in_Siam.pdf

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 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 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 correlates a great challenge to AI and the Church-Turing Thesis (e.g., see Bringsjord & Arkovna 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: Arkovna & 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 Shilladay 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).

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 .



Can you build an AI that can prove this??

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

John Licato, Naveen Sundar Govindarajulu, Selmer Bringsjord, Michael Pomeranz, Logan Gittensohn
Rensselaer Polytechnic Institute
Troy, NY
{licatj.govinn,selmer.pomeranz,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 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's (1983), as explained in (Bringsjord, 1998), where it's shown that the proof of G1, because the set of premises includes an ingenious human-designed 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 (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.¹ 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.

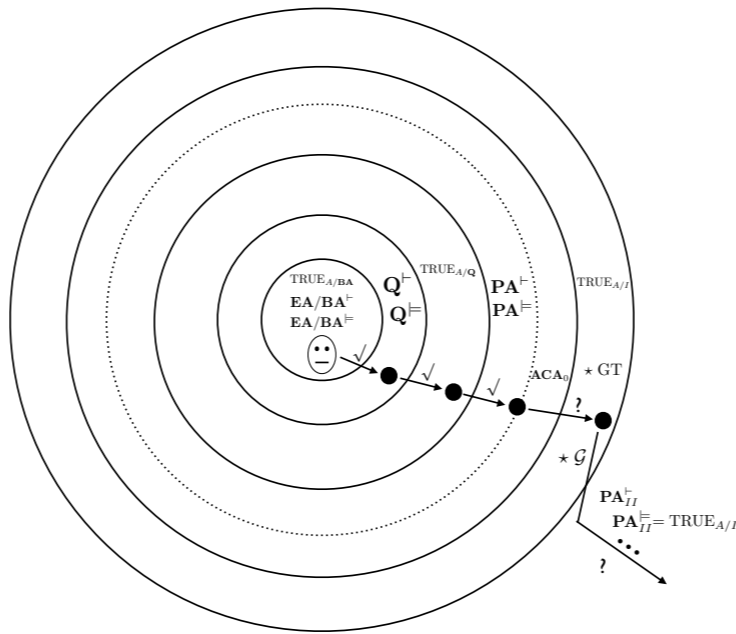
2 The Analogico-Deductive Approach

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

¹A vivid demonstration of the small-distance process can be found at http://royan.ams.rpi.edu/Godel/abstract_in_Siam.pdf

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 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 outlining 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 as 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 correlates a great challenge to AI and the Church-Turing Thesis (e.g., see Bringsjord & Arkovos 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 *wiseman* puzzle: Arkovos & 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 Shilladay 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).

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 .



Can you build an AI that can prove this??

Yes. Somehow ...