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

Selmer Bringsjord

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

Intro to Logic
3/26/2020

RAIR
Rensselaer AI and Reasoning Lab

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

Selmer Bringsjord

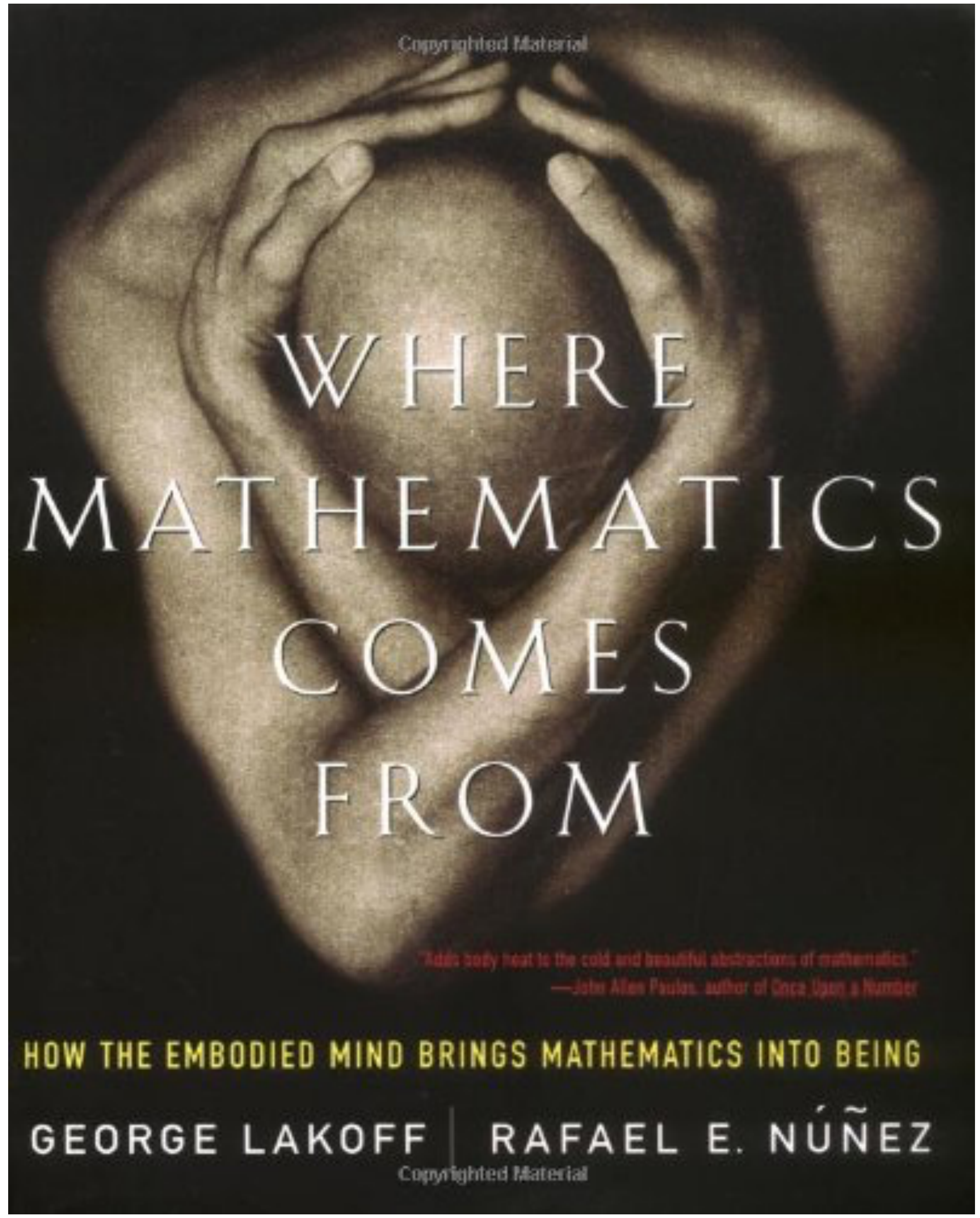
Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Lally School of Management
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

Intro to Logic
3/26/2020

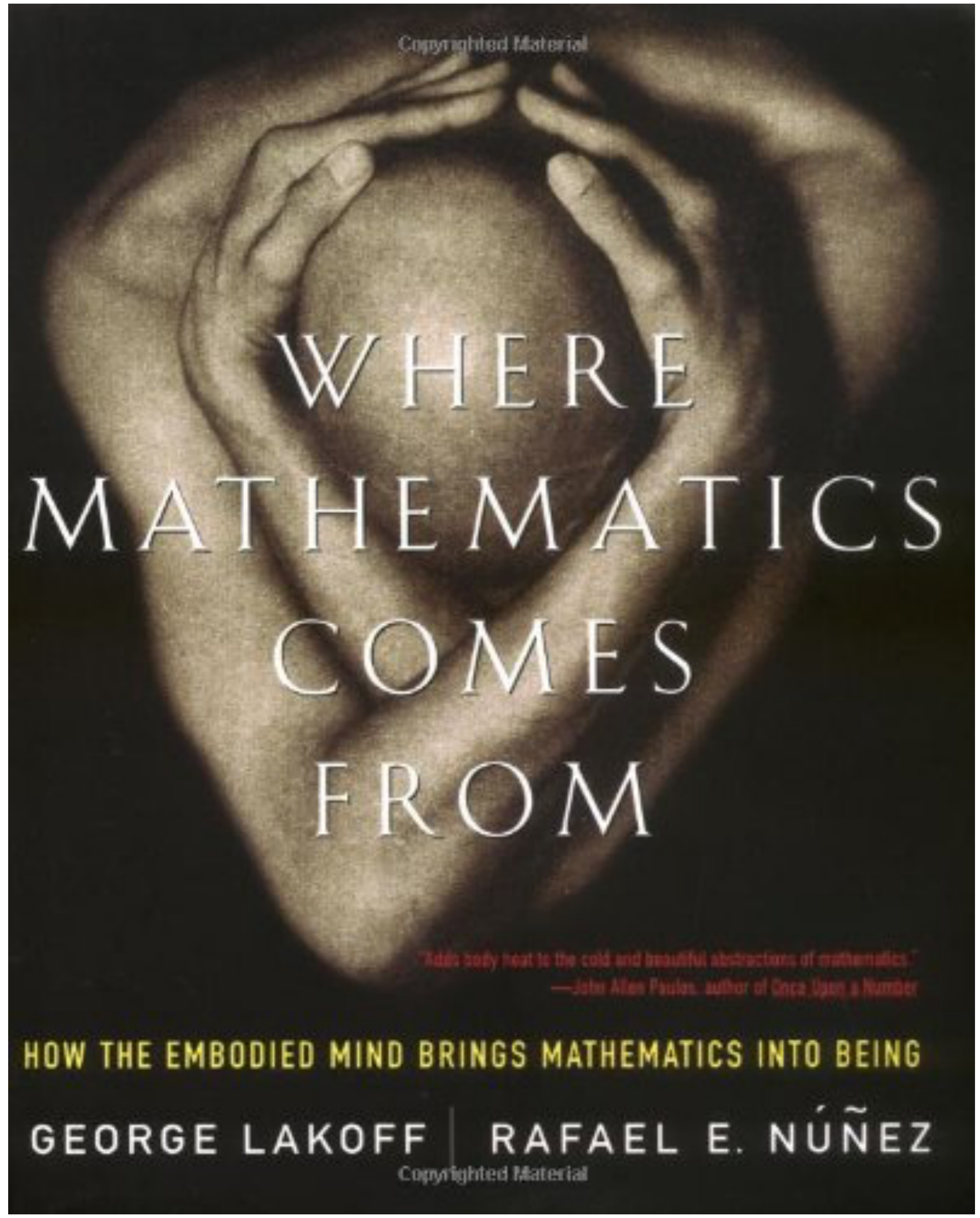
Rensselaer AI and Reasoning Lab

Do we just manufacture mathematics?

Do we just manufacture mathematics?



Do we just manufacture mathematics?



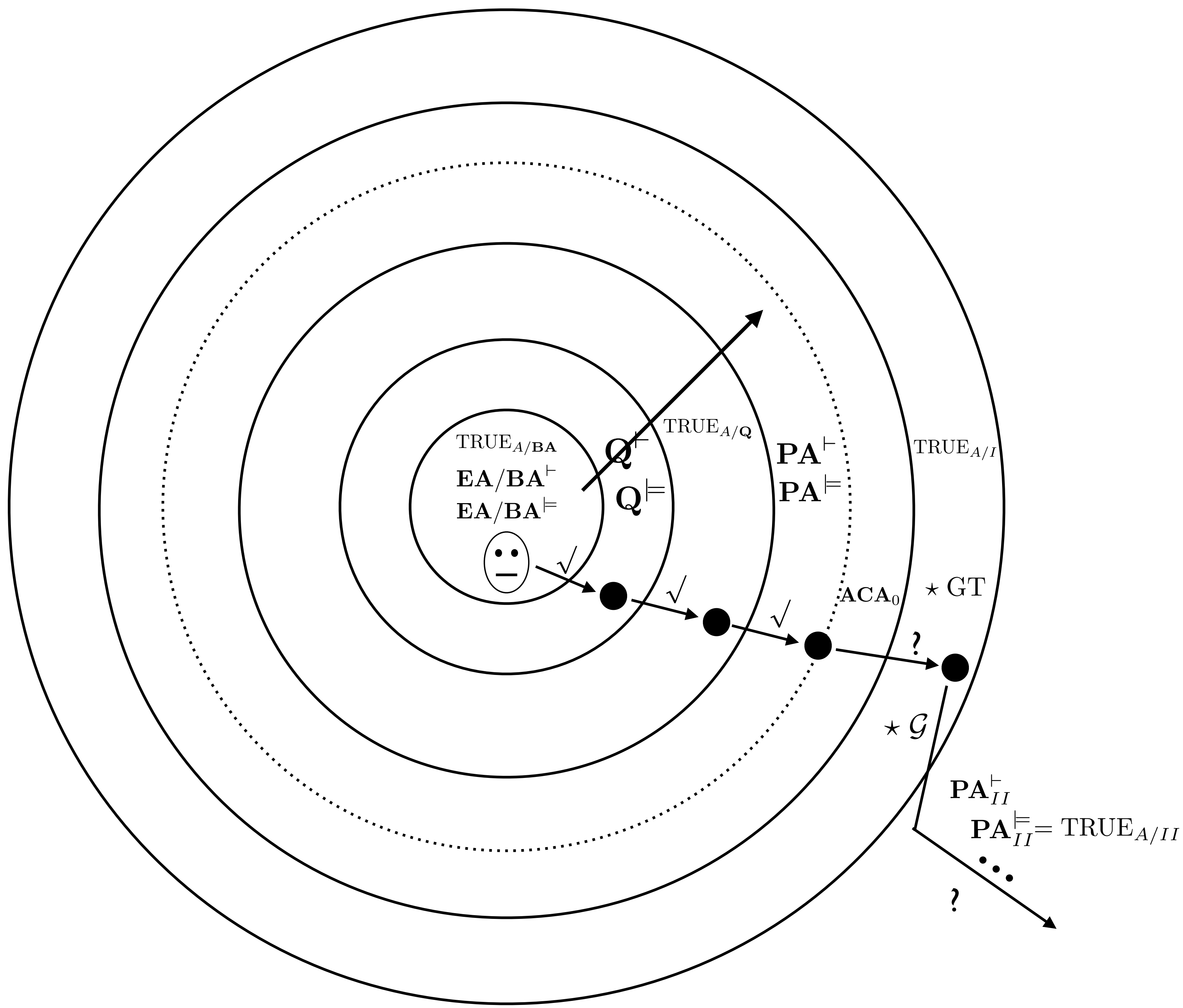
No! We tap into deep, underlying reality — and aliens do/would too …

Do we just manufacture mathematics?

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

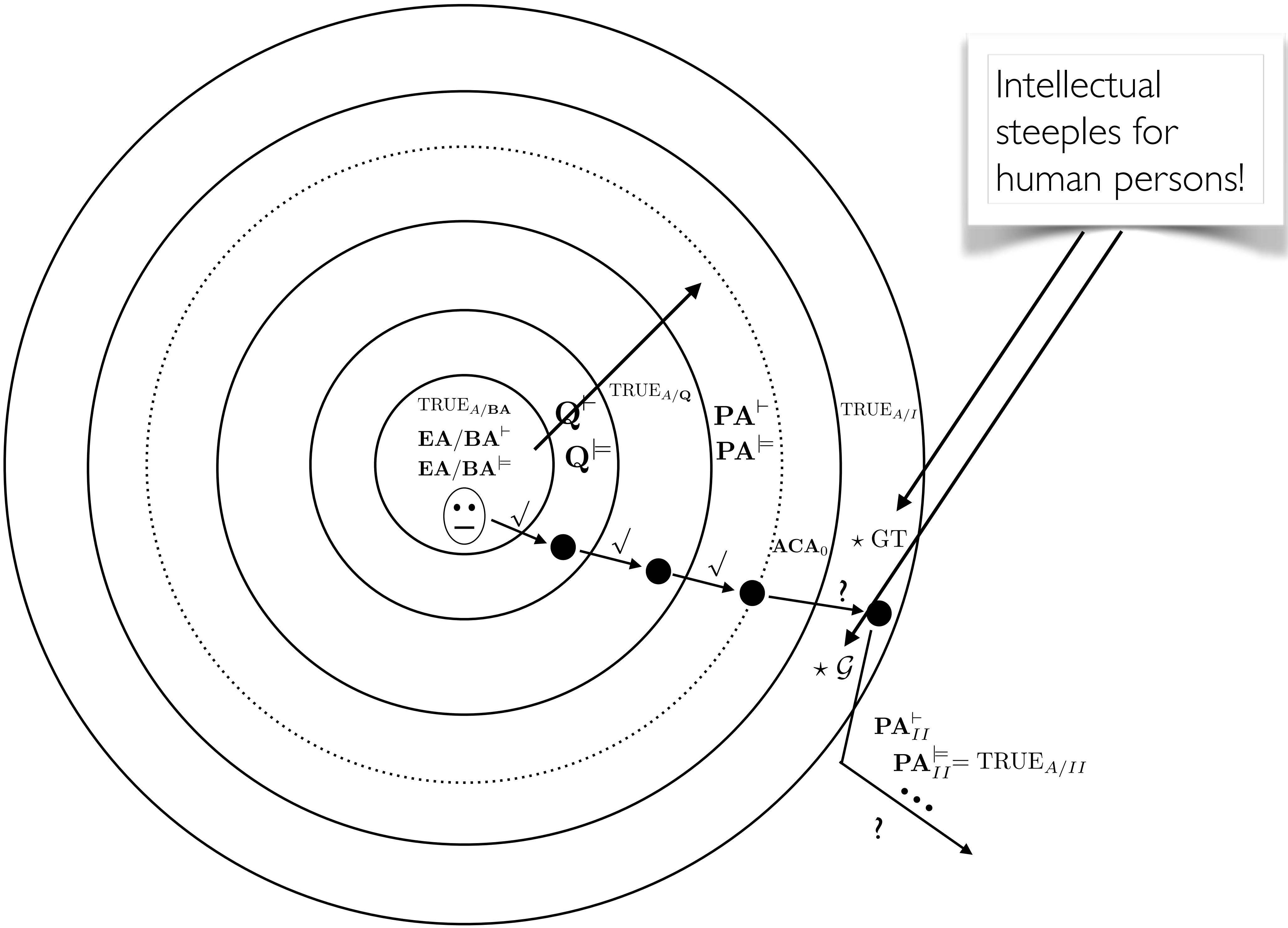
Astrologic

(Aliens & Angels on the Same "Race Track")



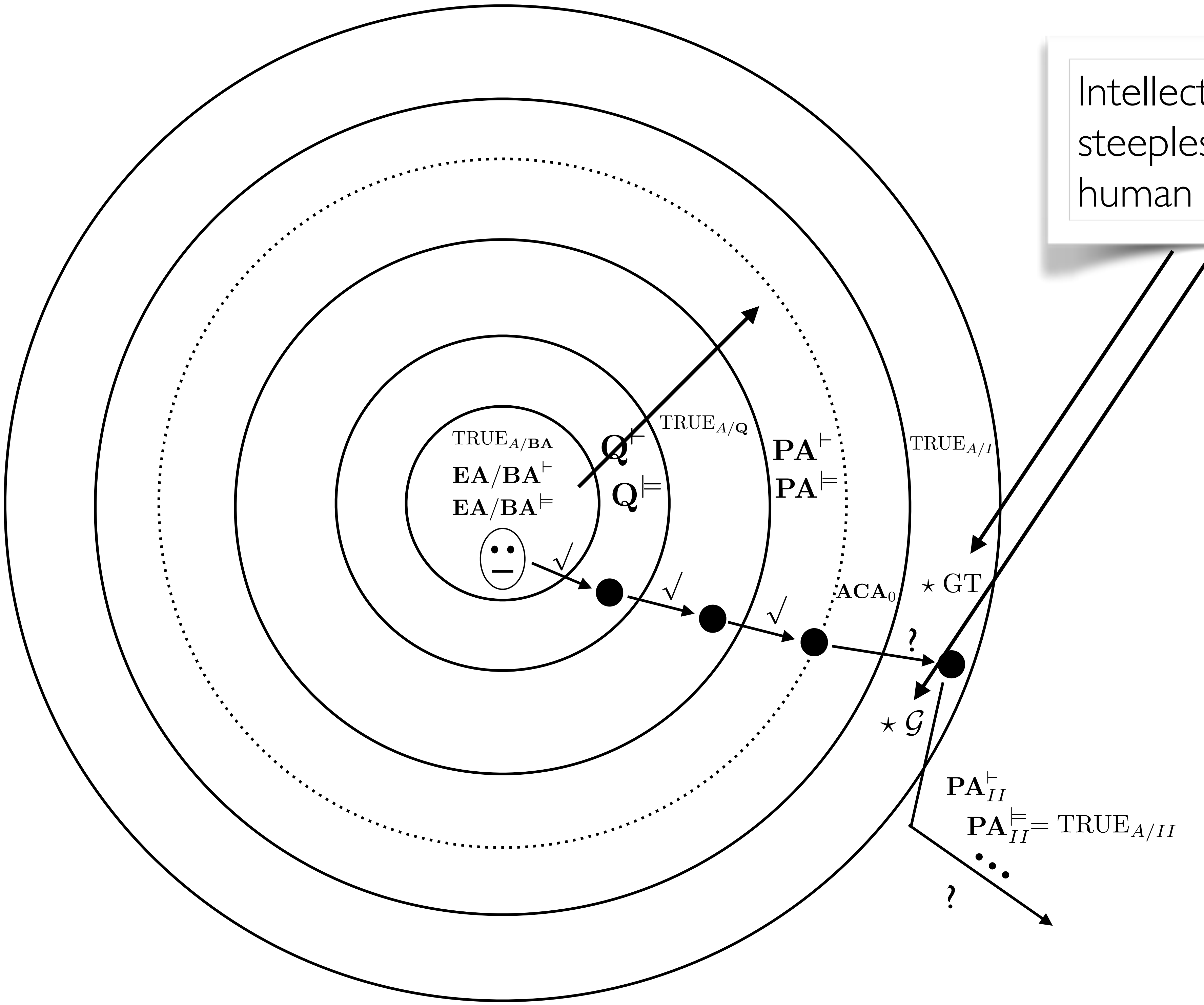
Astrologic

(Aliens & Angels on the Same "Race Track")



Astrologic

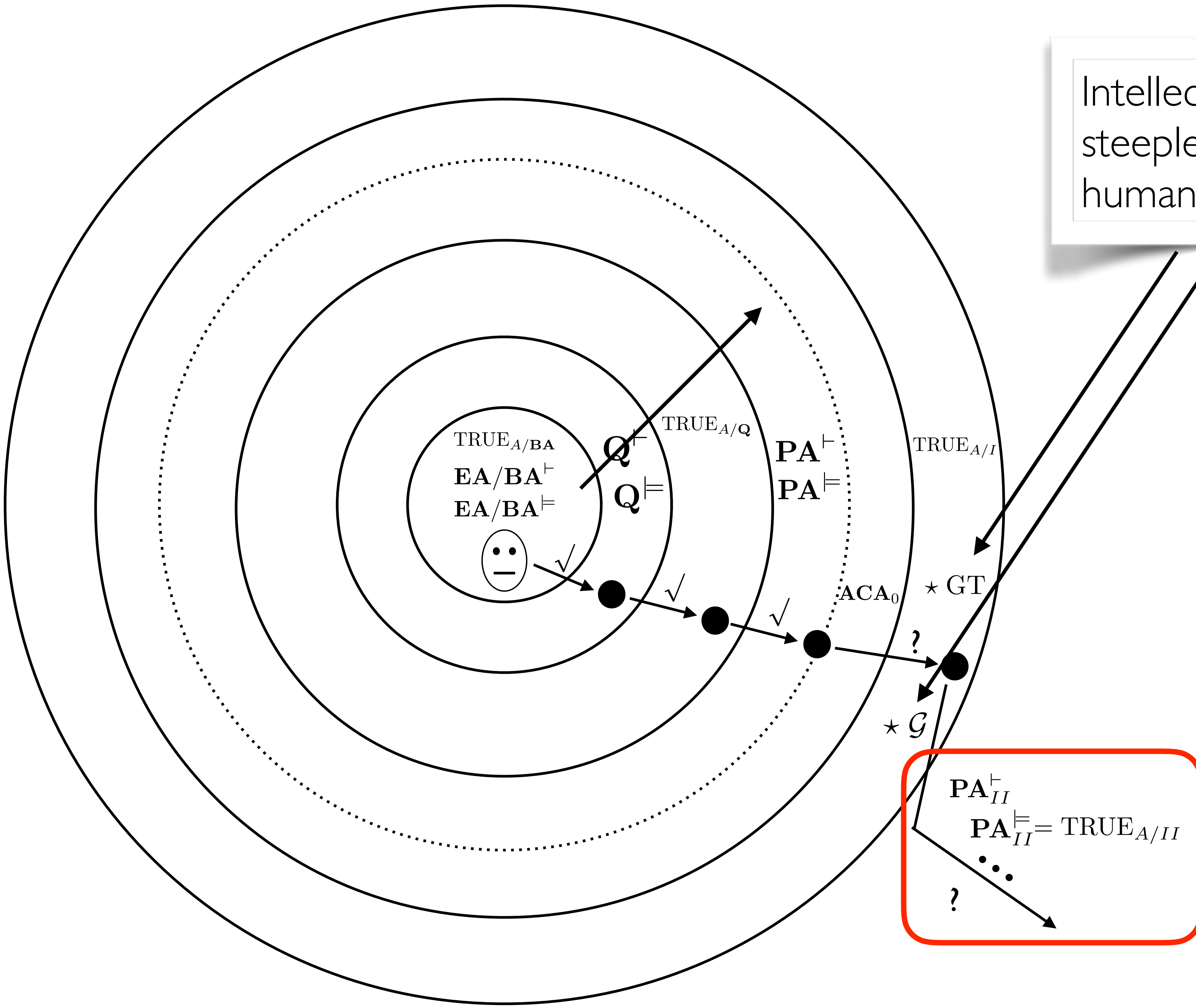
(Aliens & Angels on the Same "Race Track")



Intellectual steeples for human persons!

Astrologic

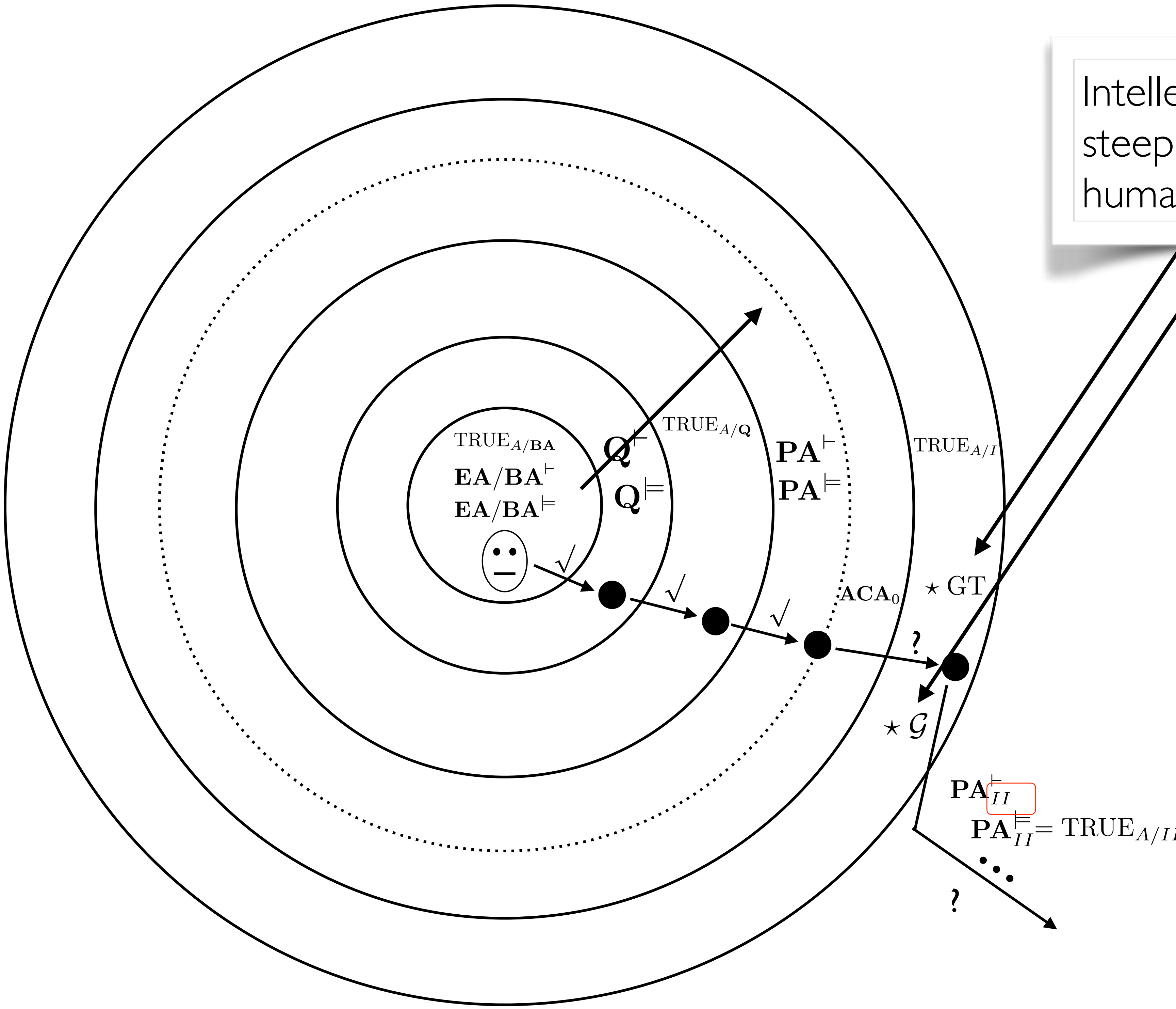
(Aliens & Angels on the Same "Race Track")



Intellectual steeples for human persons!

Astrologic

(Aliens & Angels on the Same "Race Track")



Intellectual steeples for human persons!

Q (= Robinson Arithmetic)

$$
\begin{array}{ll}
\text{A1} & \forall x(0 \neq s(x)) \\
\text{A2} & \forall x \forall y(s(x) = s(y) \rightarrow x = y) \\
\text{A3} & \forall x(x \neq 0 \rightarrow \exists y(x = s(y))) \\
\text{A4} & \forall x(x + 0 = x) \\
\text{A5} & \forall x \forall y(x + s(y) = s(x + y)) \\
\text{A6} & \forall x(x \times 0 = 0) \\
\text{A7} & \forall x \forall y(x \times s(y) = (x \times y) + x)
\end{array}
$$

Q (= Robinson Arithmetic)

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

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

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

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

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

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

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

PA (Peano Arithmetic)

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

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

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

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

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

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

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

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

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

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

PA (Peano Arithmetic)

$$
\begin{array}{ll}
\text{A1} & \forall x(0 \neq s(x)) \\
\text{A2} & \forall x \forall y(s(x) = s(y) \rightarrow x = y) \\
\text{A3} & \forall x(x \neq 0 \rightarrow \exists y(x = s(y)) \\
\text{A4} & \forall x(x + 0 = x) \\
\text{A5} & \forall x \forall y(x + s(y) = s(x + y)) \\
\text{A6} & \forall x(x \times 0 = 0) \\
\text{A7} & \forall x \forall y(x \times s(y) = (x \times y) + x)
\end{array}
$$

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

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

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

PA (Peano Arithmetic)

A1	$\forall x(0 \neq s(x))$
A2	$\forall x \forall y(s(x) = s(y) \rightarrow x = y)$
A3	$\forall x(x \neq 0 \rightarrow \exists y(x = s(y))$
A4	$\forall x(x + 0 = x)$
A5	$\forall x \forall y(x + s(y) = s(x + y))$
A6	$\forall x(x \times 0 = 0)$
A7	$\forall x \forall y(x \times s(y) = (x \times y) + x)$

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

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

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

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

Selmer Bringsjord

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

Intro to Logic
3/26/2020

RAIR
Rensselaer AI and Reasoning Lab

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

Selmer Bringsjord

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

Intro to Logic
3/26/2020

RAIR
Rensselaer AI and Reasoning Lab

FOL

√ FOL

√ FOL

Epistemic + FOL

$\mathbf{B}_d \mathbf{B}_v \mathbf{B}_d V v$

√ FOL

Epistemic + FOL (for coverage of “killer” robots, later)

$\mathbf{B}_d \mathbf{B}_v \mathbf{B}_d V v$

√ FOL

√ Epistemic + FOL (for coverage of “killer” robots, later)

$\mathbf{B}_d \mathbf{B}_v \mathbf{B}_d V v$

√ FOL

√ Epistemic + FOL (for coverage of "killer" robots, later)

$\mathbf{B}_d \mathbf{B}_v \mathbf{B}_d V v$

TOL

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

√ FOL

√ Epistemic + FOL (for coverage of "killer" robots, later)

$\mathbf{B}_d \mathbf{B}_v \mathbf{B}_d V v$

TOL

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

√ FOL

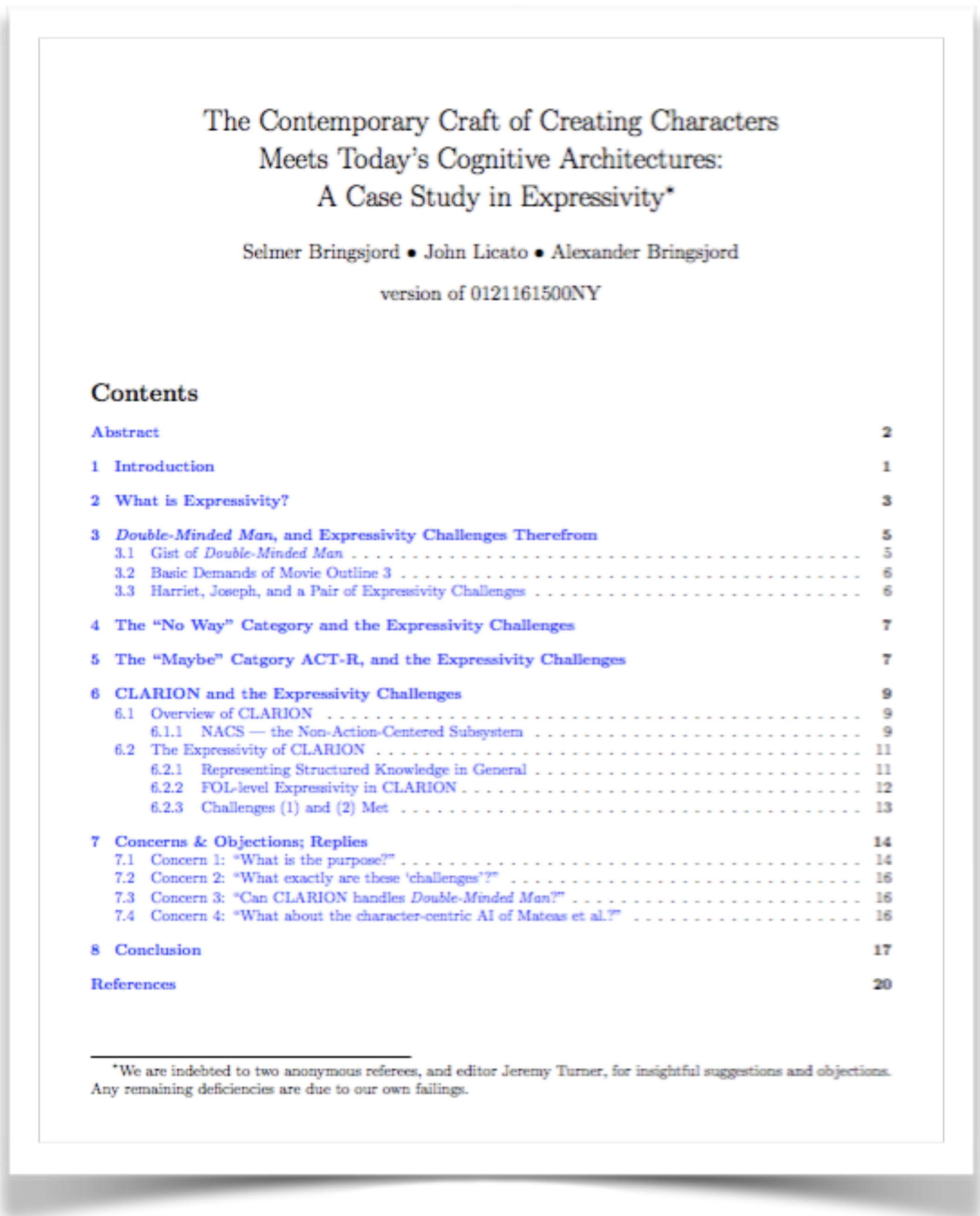
√ Epistemic + FOL (for coverage of “killer” robots, later)

$\mathbf{B}_d \mathbf{B}_v \mathbf{B}_d V v$

TOL

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

Double-Minded Man



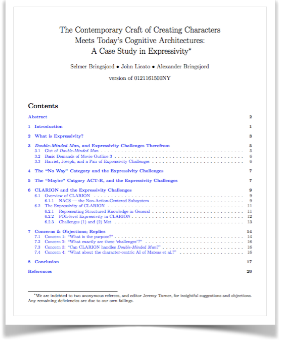
The Contemporary Craft of Creating Characters
Meets Today's Cognitive Architectures:
A Case Study in Expressivity*

Selmer Bringsjord • John Licato • Alexander Bringsjord

version of 0121161500NY

Contents

*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.



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

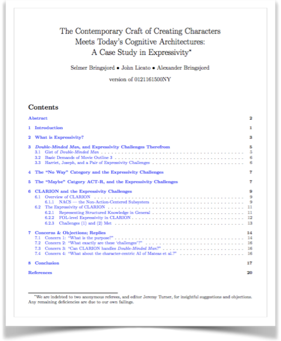
Selmer Bringsjord • John Licato • Alexander Bringsjord

version of 0121161500NY

Contents

*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



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

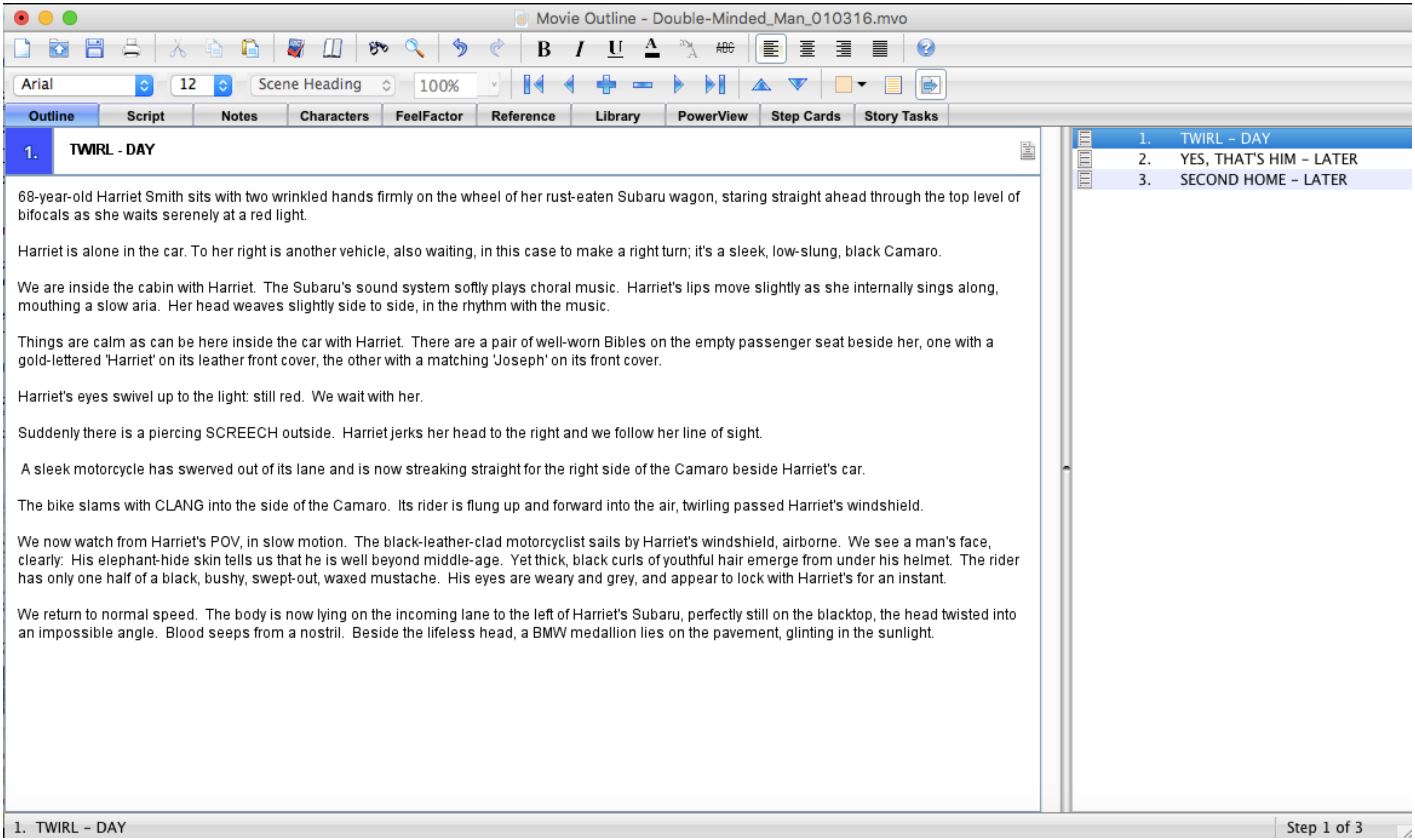
Selmer Bringsjord • John Licato • Alexander Bringsjord

version of 0121161500NY

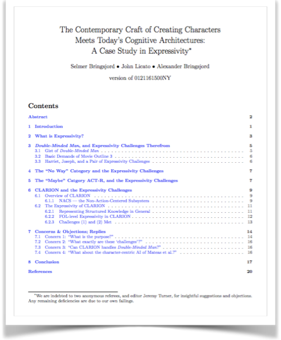
Contents

*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



Double-Minded Man

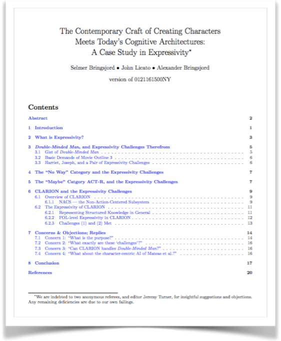

The Contemporary Craft of Creating Characters
Meets Today's Cognitive Architectures:
A Case Study in Expressivity*

Selmer Bringsjord • John Licato • Alexander Bringsjord

version of 0121161500NY

Contents

*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.


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

*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

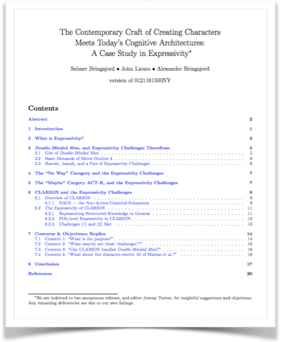
Double-Minded Man
by

S Bringsjord & A Bringsjord

DRAFT #5
© June 30 2016

Selmer.Bringsjord@gmail.com

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

*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.

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

Double-Minded Man

1. TWIRL - DAY

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

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

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

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

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

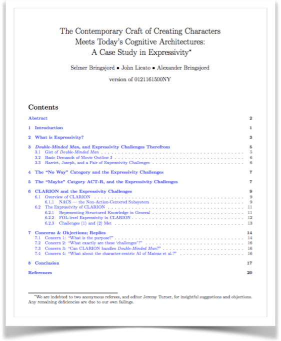
Suddenly there is a piercing SCREECH outside. Harriet jerks her head to the right and we follow her line of sight.

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

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

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

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



Double-Minded Man

1. TWIRL - DAY

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

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

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

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

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

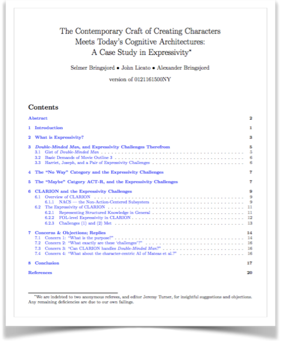
Suddenly there is a piercing SCREECH outside. Harriet jerks her head to the right and we follow her line of sight.

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

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

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

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



Double-Minded Man

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.

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

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

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

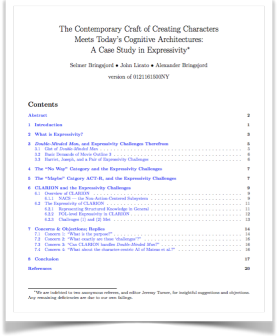
Suddenly there is a piercing SCREECH outside. Harriet jerks her head to the right and we follow her line of sight.

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

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

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

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



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

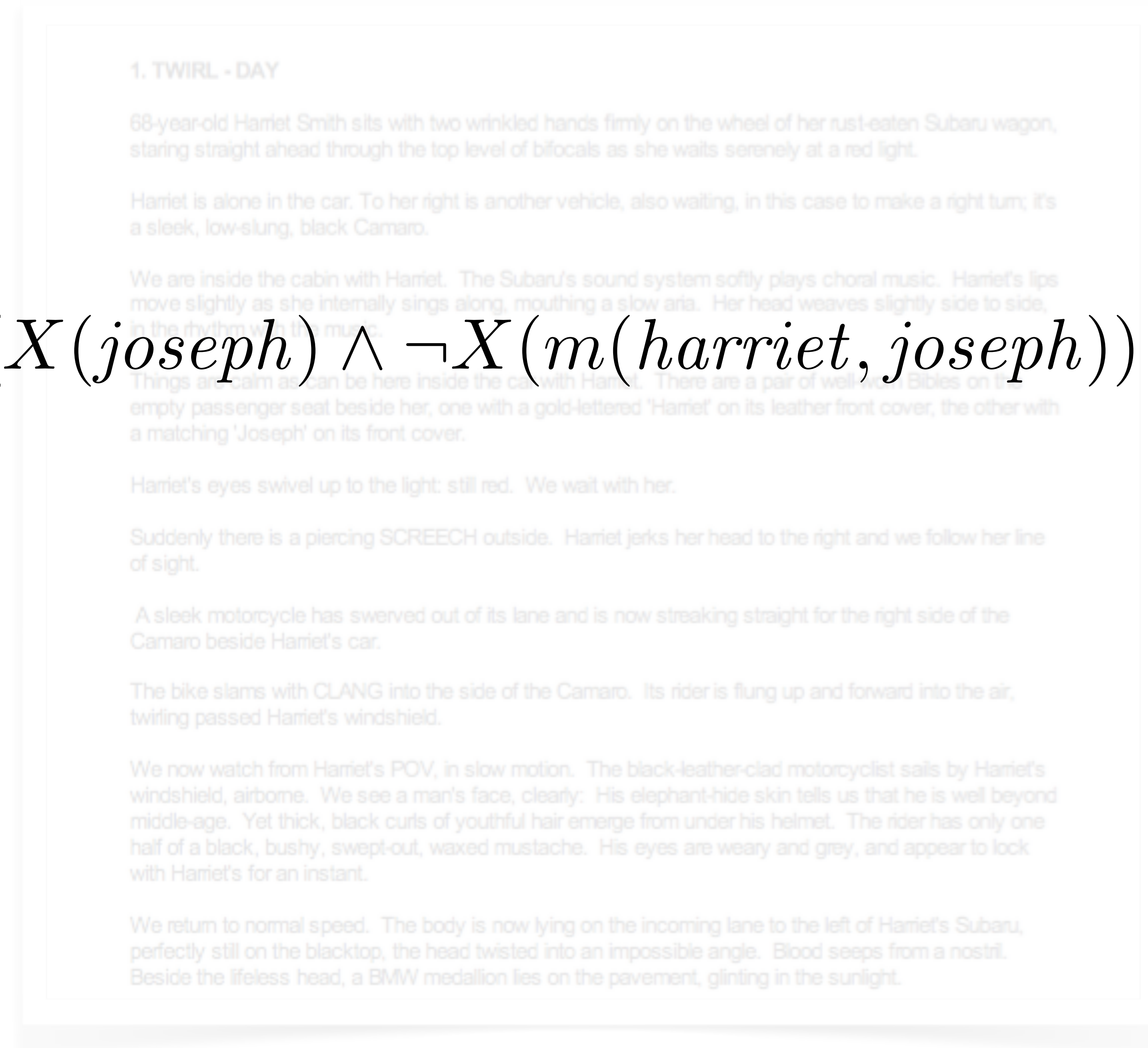
Selmer Bringsjord • John Licato • Alexander Bringsjord

version of 0121161500NY

Contents

*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



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.

$$\checkmark \quad \exists X[X(joseph) \land \neg X(m(harriet, joseph)) \land Sleazy(X)]$$

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

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

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

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

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

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

Climbing the *k*-order Ladder

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

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

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

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

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

Climbing the *k*-order Ladder

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

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

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

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

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

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

Climbing the *k*-order Ladder

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

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

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

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

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

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

Climbing the *k*-order Ladder

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Climbing the *k*-order Ladder

TOL $\exists x, y\ \exists R, R^2[R(x) \land R(y) \land R^2(x,y) \land Positive(R^2) \land R(fatherOf(x))]$

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

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

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

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

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

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

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

TOL $\mathscr{L}_3$

$$\exists x, y\ \exists R, R^2[R(x) \land R(y) \land R^2(x,y) \land Positive(R^2) \land R(fatherOf(x))]$$

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

SOL $\mathscr{L}_2$

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

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

FOL $\mathscr{L}_1$

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

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

ZOL $\mathscr{L}_0$

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

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

$\vdots$

TOL $\mathscr{L}_3$

$$\exists x, y\ \exists R, R^2[R(x) \land R(y) \land R^2(x,y) \land Positive(R^2) \land R(fatherOf(x))]$$

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

SOL $\mathscr{L}_2$

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

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

FOL $\mathscr{L}_1$

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

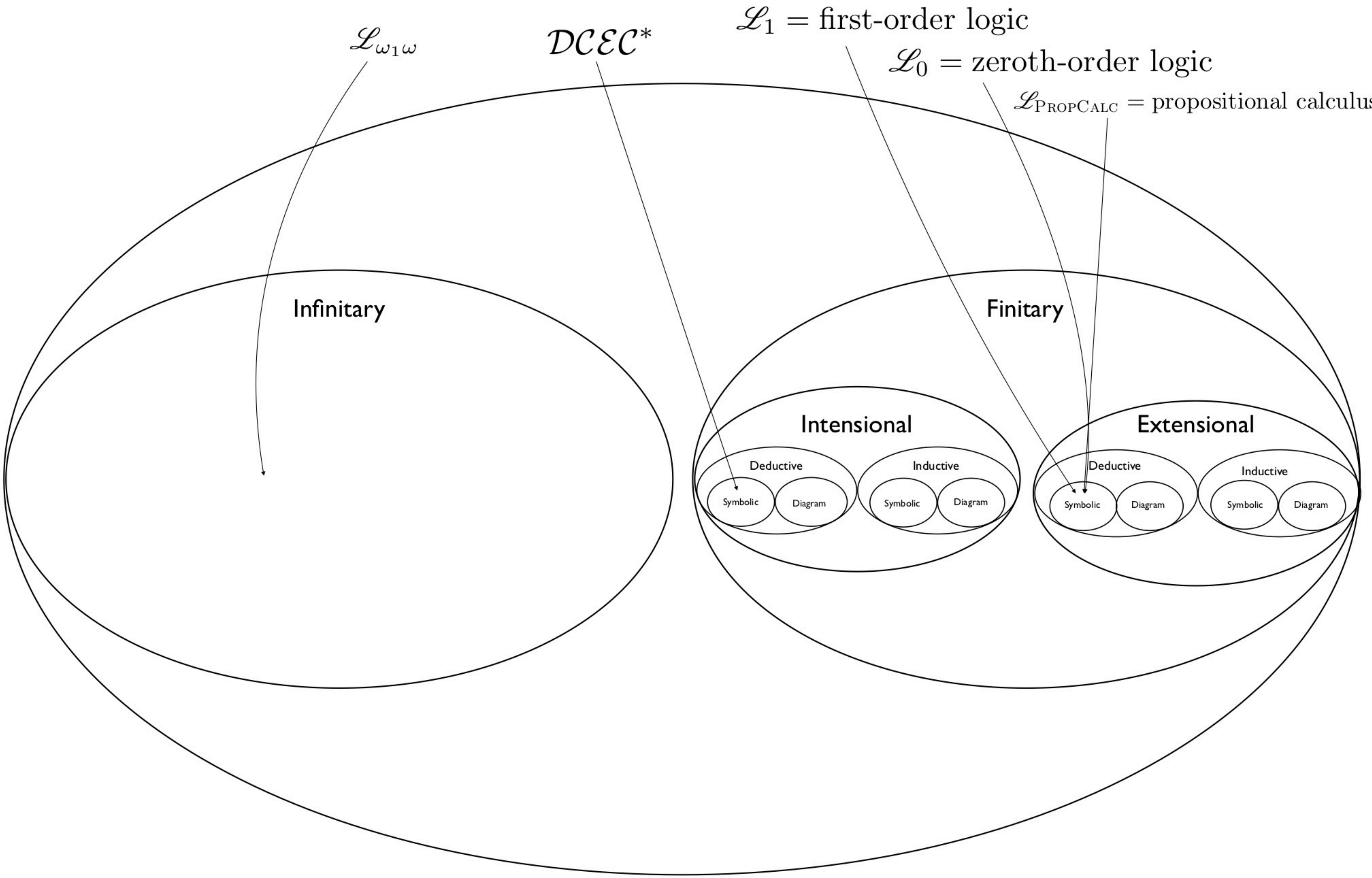
There's some thing which is a llama and likes *b* (which is also a llama), and whose father is a llama too.

ZOL $\mathscr{L}_0$

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

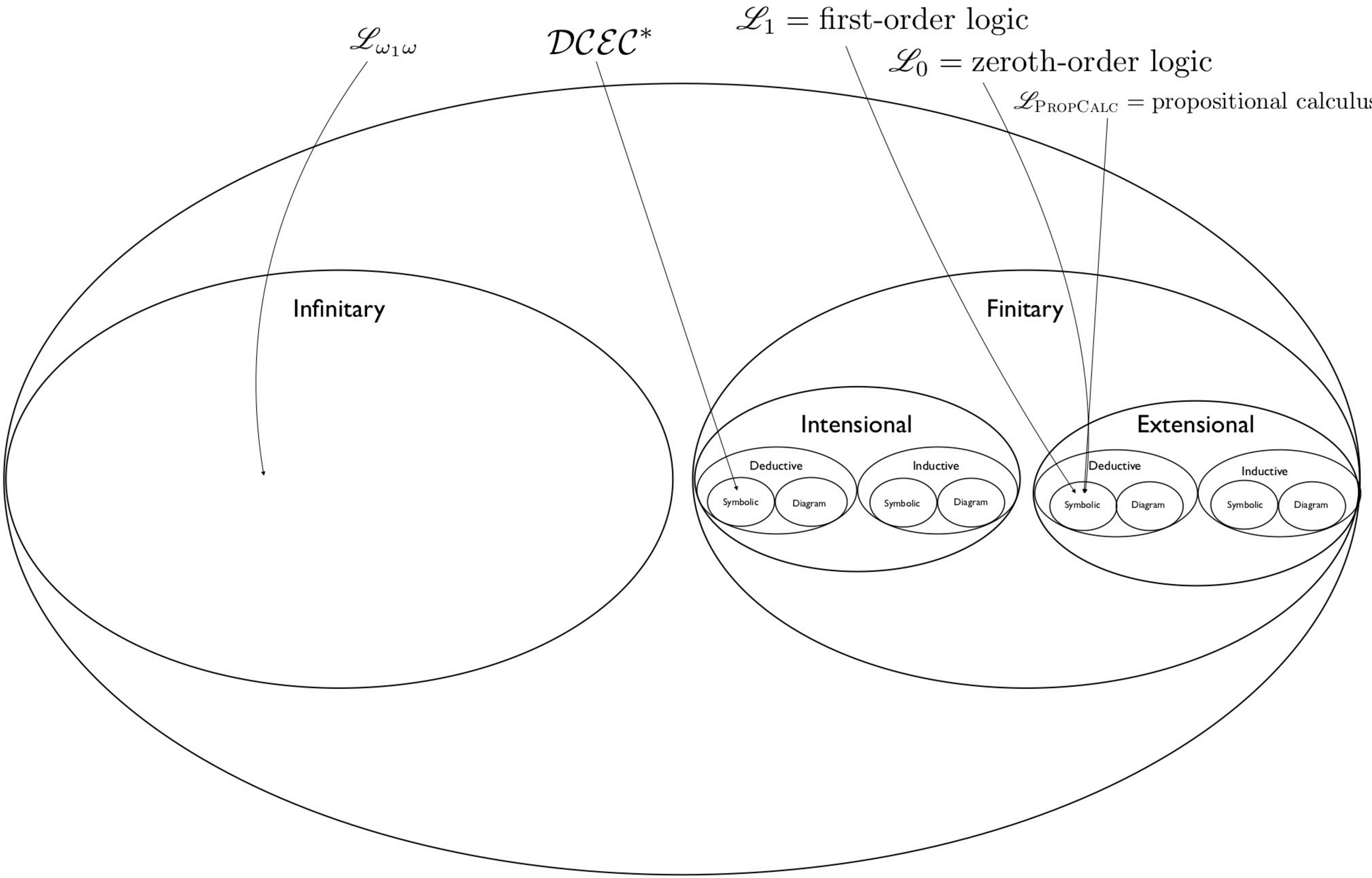
a is a llama, as is *b*, *a* likes *b*, and the father of *a* is a llama as well.

The Universe of Logics

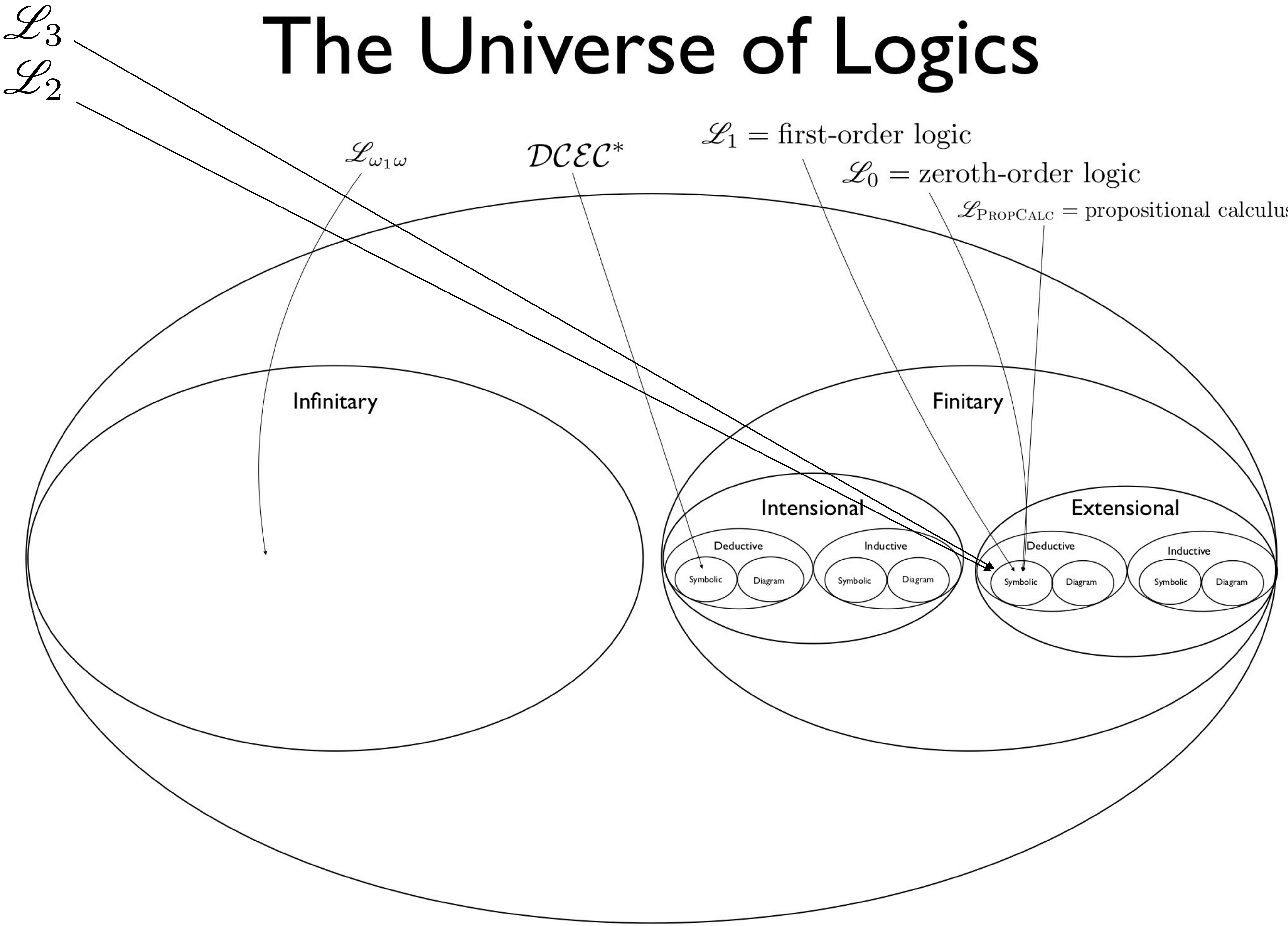


$\mathscr{L}_3$
$\mathscr{L}_2$

The Universe of Logics



The Universe of Logics



Climbing the *k*-order Ladder

⋮

TOL $\mathscr{L}_3$

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

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

SOL $\mathscr{L}_2$

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

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

FOL $\mathscr{L}_1$

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

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

ZOL $\mathscr{L}_0$

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

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

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

Selmer Bringsjord

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

Intro to Logic
3/26/2020

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

Selmer Bringsjord

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

Intro to Logic
3/26/2020

RAIR
Rensselaer AI and Reasoning Lab

“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 *being God* (= G) is an essential property of any thing that has G.	(8), (7), (4), (1)
		Part IV	
	(10)	A thing has necessary existence if and only if all the essences that thing has imply that something exists and has all those essences.	definition
	(10)	A thing has necessary existence if and only if all the essential properties that thing has imply that something exists and has all those essential properties.	definition
	(11)	Necessary existence is a positive property.	premise
∴	(12)	Necessarily, God exists.	(6), (9), (10), (11) **QED**

Gödel's "God Theorem" (formalized, machine verified)

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

Gödel's "God Theorem" (formalized, machine verified)

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

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

Selmer Bringsjord

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

Intro to Logic
3/26/2020

RAIR
Rensselaer AI and Reasoning Lab

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

Selmer Bringsjord

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

Intro to Logic
3/26/2020

RAIR
Rensselaer AI and Reasoning Lab

$PA_I = Z_I$

$$
\begin{array}{ll}
\text{A1} & \forall x(0 \neq s(x)) \\
\text{A2} & \forall x \forall y(s(x) = s(y) \rightarrow x = y) \\
\text{A3} & \forall x(x \neq 0 \rightarrow \exists y(x = s(y)) \\
\text{A4} & \forall x(x + 0 = x) \\
\text{A5} & \forall x \forall y(x + s(y) = s(x + y)) \\
\text{A6} & \forall x(x \times 0 = 0) \\
\text{A7} & \forall x \forall y(x \times s(y) = (x \times y) + x)
\end{array}
$$

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

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

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

$PA_1 = Z_1$

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

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

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

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

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

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

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

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

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

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

$PA_2 = Z_2$

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

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

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

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

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

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

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

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

$PA_2 = Z_2$

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

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

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

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

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

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

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

Induction Axiom $\forall X([X(0) \wedge \forall x(X(x) \rightarrow X(s(x))] \rightarrow \forall x X(x))$

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

$PA_2 = Z_2$

A1	$\forall x(0 \neq s(x))$
A2	$\forall x \forall y(s(x) = s(y) \rightarrow x = y)$
A3	$\forall x(x \neq 0 \rightarrow \exists y(x = s(y)))$
A4	$\forall x(x + 0 = x)$
A5	$\forall x \forall y(x + s(y) = s(x + y))$
A6	$\forall x(x \times 0 = 0)$
A7	$\forall x \forall y(x \times s(y) = (x \times y) + x)$

Induction Axiom $\forall X([X(0) \wedge \forall x(X(x) \rightarrow X(s(x))] \rightarrow \forall x X(x))$

Comprehension Schema $\exists X(\forall x X(x) \leftrightarrow \phi(x))$ where $\phi(x) \in C$

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

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

Selmer Bringsjord

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

Intro to Logic
3/26/2020

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

Selmer Bringsjord

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

Intro to Logic
3/26/2020

Rensselaer AI and Reasoning Lab

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

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

Abstract

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

1 Introduction

Gödel's proofs of his incompleteness theorems are among the greatest intellectual achievements of the 20th century. Even armed with the suggestion that the Liar Paradox (**LP**) might somehow be useful as a guide to proving the incompleteness of Peano Arithmetic (**PA**),[1] the level of creativity and philosophical clarity required to actually tie the two concepts together and produce a valid proof is staggering; it certainly should not be controversial to claim that no computational reasoning system can, at present, achieve this sort of feat without significant human assistance.

1.1 Automating the Proof of G1

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

Likewise, [Amnon, 1993] is an exact parallel of the human-devised proof given by [Kleene, 1996]. Finally, in much more recent and truly impressive work by [Sieg and Field, 2005], there is a move to natural-deduction formats, which we applaud—but the machine essentially begins its processing at a point exceedingly close to where it needs to end up. As 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.[2] In sum, while a lot of commendable work has been done to build the foundation for our prospective work, the daunting formal and engineering challenge of producing a computational system able to produce **G1** without clever 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

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

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

G1

$\mathcal{G}$

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

Naveen Sundar Govindarajulu, John Licato, and Selmer Bringsjord
Department of Computer Science
Department of Cognitive Science
Rensselaer AI & Reasoning Laboratory
govinn@rpi.edu • licatj@rpi.edu • selmer@rpi.edu

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

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

A Context: Infinitary Reasoning, Hypercomputation, and Humble Engineering

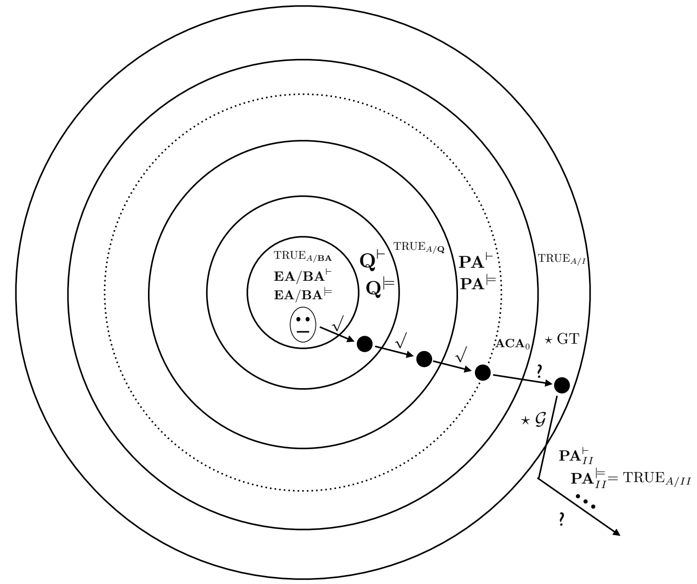
Bringsjord has repeatedly pointed out the obvious fact that the behavior of formal scientists, taken at face value, involve various infinitary structures and reasoning. (We say "at face value" to simply indicate we don't presuppose some view that denies the reality of infinite entities routinely involved in the formal sciences.) For example, in (Bringsjord & van Heuveln 2003), Bringsjord himself operates as such a scientist in presenting an infinitary paradox which to his knowledge has yet to be solved. And he has argued that apparently infinitary behavior constitutes a grave challenge to AI and the Church-Turing Thesis (e.g., see Bringsjord & Arkoudas 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 these 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: Arkoudas & 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 Shilliday 2009).

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

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


Analogico-Deductive Generation of Gödel's First Incompleteness Theorem from the Liar Paradox
John Licato, Naveen Sundar Govindarajulu, Selmer Bringsjord, Michael Pomeranz, Logan Gittelson
Rensselaer Polytechnic Institute
Troy, NY
Abstract
1 Introduction
1.1 Automating the Proof of G1
2 The Analogico-Deductive Approach
2.1 Conjecture Generation

GI


$\mathrm{TRUE}_{A/BA}$
$\mathbf{EA/BA}^{\vdash}$
$\mathbf{EA/BA}^{\models}$
$\mathbf{Q}^{\vdash}$
$\mathbf{Q}^{\models}$
$\mathrm{TRUE}_{A/Q}$
$\mathbf{PA}^{\vdash}$
$\mathbf{PA}^{\models}$
$\mathrm{TRUE}_{A/I}$
ACA_0
$\star$ GT
?
$\star\ \mathcal{G}$
$\mathbf{PA}_{II}^{\vdash}$
$\mathbf{PA}_{II}^{\models} = \mathrm{TRUE}_{A/II}$
?

$\mathcal{G}$


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
A Context: Infinitary Reasoning, Hypercomputation, and Humble Engineering

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

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

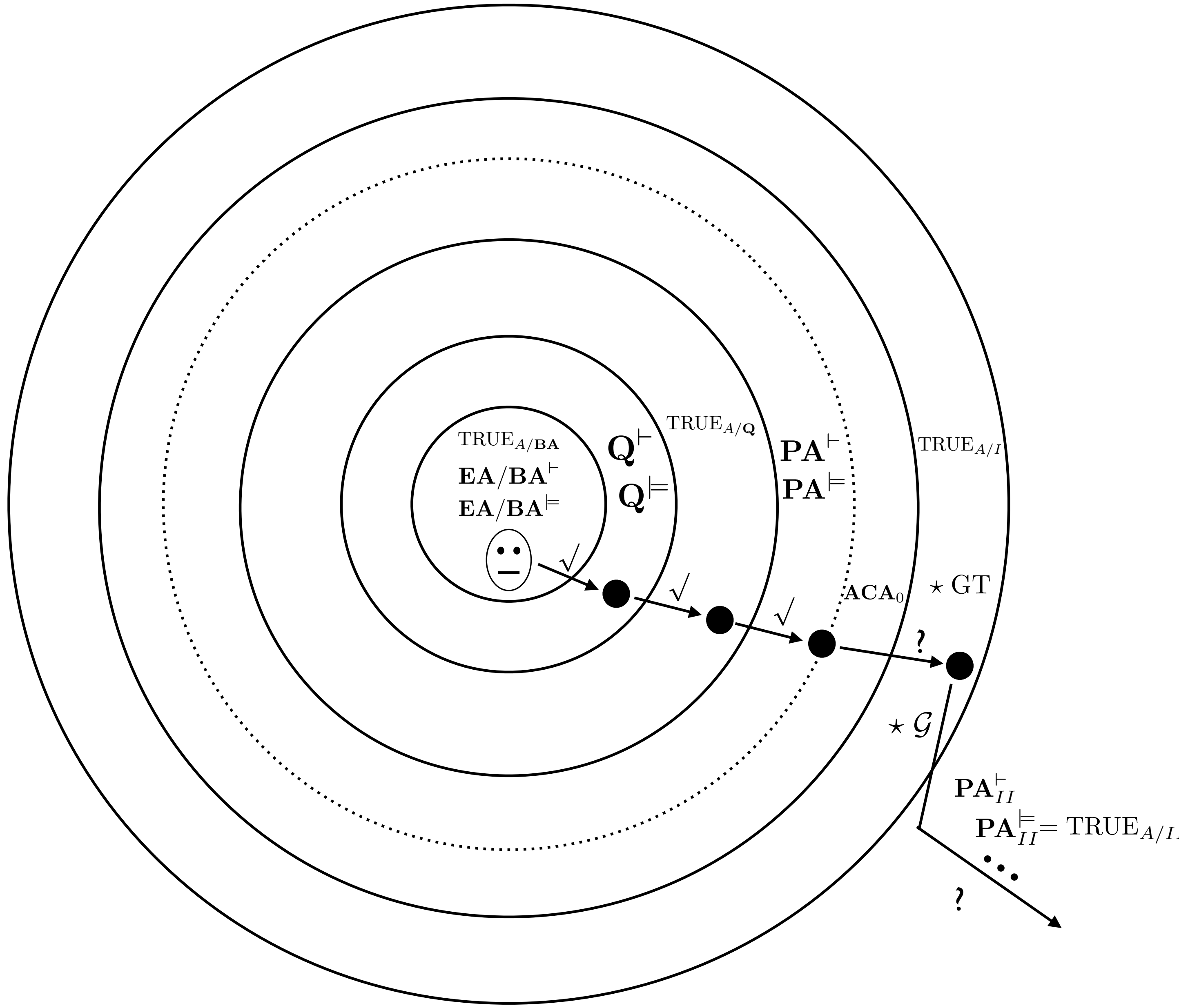
Abstract

1 Introduction

1.1 Automating the Proof of G1

2 The Analogico-Deductive Approach

2.1 Conjecture Generation



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

A Context: Infinitary Reasoning, Hypercomputation, and Humble Engineering

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

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

Abstract

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

1 Introduction

Gödel's proofs of his incompleteness theorems are among the greatest intellectual achievements of the 20th century. Even armed with the suggestion that the Liar Paradox (**LP**) might somehow be useful as a guide to proving the incompleteness of Peano Arithmetic (**PA**),[1] the level of creativity and philosophical clarity required to actually tie the two concepts together and produce a valid proof is staggering; it certainly should not be controversial to claim that no computational reasoning system can, at present, achieve this sort of feat without significant human assistance.

1.1 Automating the Proof of G1

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

Likewise, [Amnon, 1993] is an exact parallel of the human-devised proof given by [Kleene, 1996]. Finally, in much more recent and truly impressive work by [Sieg and Field, 2005], there is a move to natural-deduction formats, which we applaud—but the machine essentially begins its processing at a point exceedingly close to where it needs to end up. As 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.[2] In sum, while a lot of commendable work has been done to build the foundation for our prospective work, the daunting formal and engineering challenge of producing a computational system able to produce **G1** without clever seeding from a human remains entirely unmet.

2 The Analogico-Deductive Approach

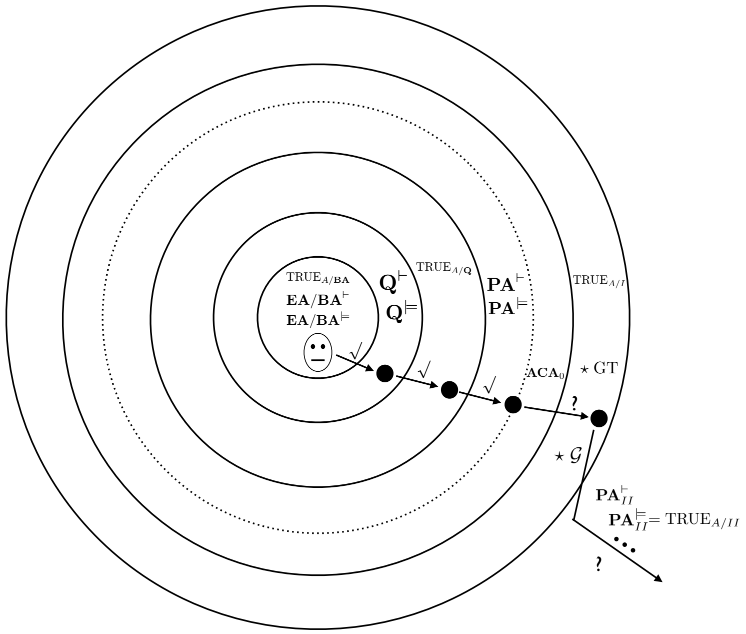
2.1 Conjecture Generation

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

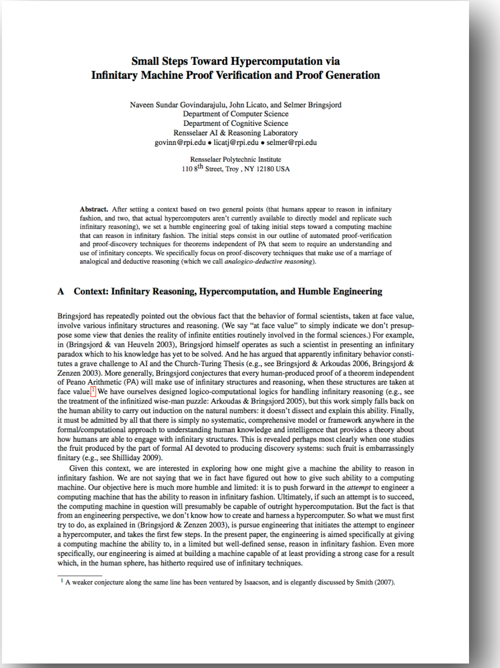
[1] **G1** of course applies to any axiom system meeting the standard conditions (Turing-decidability, representability, consistency), but we tend to refer to **PA** for economization.

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

GI



$\mathcal{G}$



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

Naveen Sundar Govindarajulu, John Licato, and Selmer Bringsjord
Department of Computer Science
Department of Cognitive Science
Rensselaer AI & Reasoning Laboratory
govinn@rpi.edu • licatj@rpi.edu • selmer@rpi.edu

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

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

A Context: Infinitary Reasoning, Hypercomputation, and Humble Engineering

Bringsjord has repeatedly pointed out the obvious fact that the behavior of formal scientists, taken at face value, involve various infinitary structures and reasoning. (We say "at face value" to simply indicate we don't presuppose some view that denies the reality of infinite entities routinely involved in the formal sciences.) For example, in (Bringsjord & van Heuveln 2003), Bringsjord himself operates as such a scientist in presenting an infinitary paradox which to his knowledge has yet to be solved. And he has argued that apparently infinitary behavior constitutes a grave challenge to AI and the Church-Turing Thesis (e.g., see Bringsjord & Arkoudas 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 these 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: Arkoudas & 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 Shilliday 2009).

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

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

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

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

Abstract

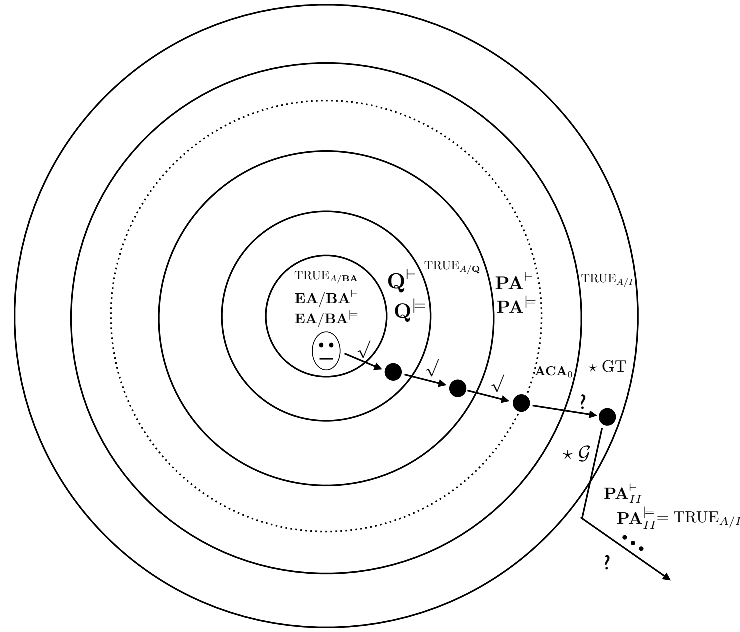
Gödel's proof of his famous first incompleteness theorem (**G1**) has quite understandably long been a tantalizing target for those wanting to engineer impressively intelligent computational systems. After all, in establishing **G1**, Gödel did something that by any metric must be classified as stunningly intelligent. We observe that it has long been under-

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

G1



$\mathcal{G}$

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

Naveen Sundar Govindarajulu, John Licato, and Selmer Bringsjord
Department of Computer Science
Department of Cognitive Science
Rensselaer AI & Reasoning Laboratory
govinn@rpi.edu • licatj@rpi.edu • selmer@rpi.edu

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

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

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 Pomeranz, Logan Gittelson
Rensselaer Polytechnic Institute
Troy, NY
{licatj,govinn,selmer,pomerm,gittel}@rpi.edu

Abstract

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

1 Introduction

Gödel's proofs of his incompleteness theorems are among the greatest intellectual achievements of the 20th century. Even armed with the suggestion that the Liar Paradox (**LP**) might somehow be useful as a guide to proving the incompleteness of Peano Arithmetic (**PA**),[1] the level of creativity and philosophical clarity required to actually tie the two concepts together and produce a valid proof is staggering; it certainly should not be controversial to claim that no computational reasoning system can, at present, achieve this sort of feat without significant human assistance.

1.1 Automating the Proof of G1

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

Likewise, [Amnon, 1993] is an exact parallel of the human-devised proof given by [Kleene, 1996]. Finally, in much more recent and truly impressive work by [Sieg and Field, 2005], there is a move to natural-deduction formats, which we applaud—but the machine essentially begins its processing at a point exceedingly close to where it needs to end up. As 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.[2] In sum, while a lot of commendable work has been done to build the foundation for our prospective work, the daunting formal and engineering challenge of producing a computational system able to produce **G1** without clever seeding from a human remains entirely unmet.

2 The Analogico-Deductive Approach

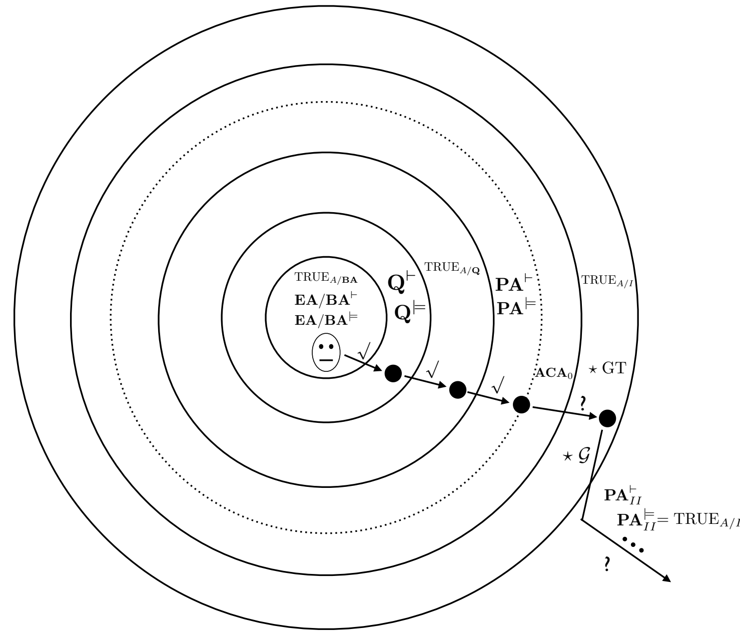
2.1 Conjecture Generation

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

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

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

G1



$\mathcal{G}$

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

Naveen Sundar Govindarajulu, John Licato, and Selmer Bringsjord
Department of Computer Science
Department of Cognitive Science
Rensselaer AI & Reasoning Laboratory
govinn@rpi.edu • licatj@rpi.edu • selmer@rpi.edu

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

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

A Context: Infinitary Reasoning, Hypercomputation, and Humble Engineering

Bringsjord has repeatedly pointed out the obvious fact that the behavior of formal scientists, taken at face value, involve various infinitary structures and reasoning. (We say "at face value" to simply indicate we don't presuppose some view that denies the reality of infinite entities routinely involved in the formal sciences.) For example, in (Bringsjord & van Heuveln 2003), Bringsjord himself operates as such a scientist in presenting an infinitary paradox which to his knowledge has yet to be solved. And he has argued that apparently infinitary behavior constitutes a grave challenge to AI and the Church-Turing Thesis (e.g., see Bringsjord & Arkoudas 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 these 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: Arkoudas & 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 Shilliday 2009).

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

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

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 Gittelson
Rensselaer Polytechnic Institute
Troy, NY

Abstract

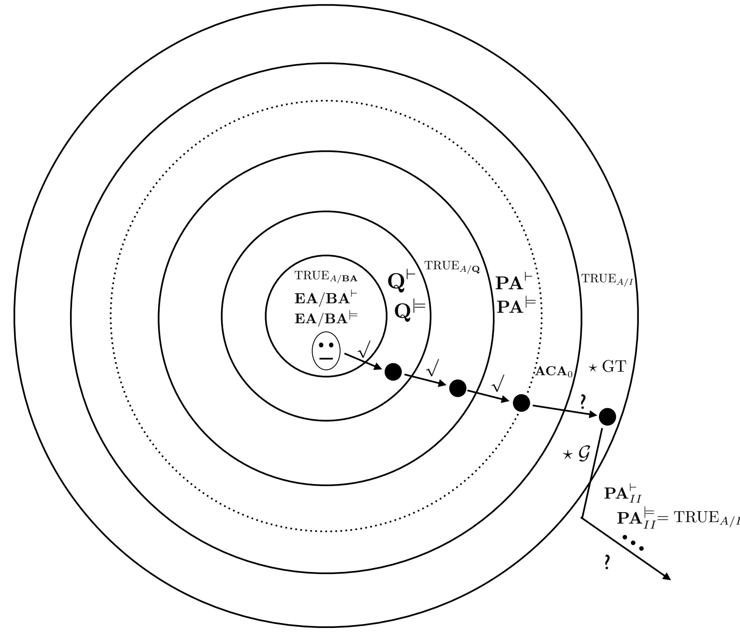
1 Introduction

1.1 Automating the Proof of G1

2 The Analogico-Deductive Approach

2.1 Conjecture Generation

G1



$\mathcal{G}$


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

Naveen Sundar Govindarajulu, John Licato, and Selmer Bringsjord

A Context: Infinitary Reasoning, Hypercomputation, and Humble Engineering

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^0_1 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 Gittelson
Rensselaer Polytechnic Institute
Troy, NY

Abstract

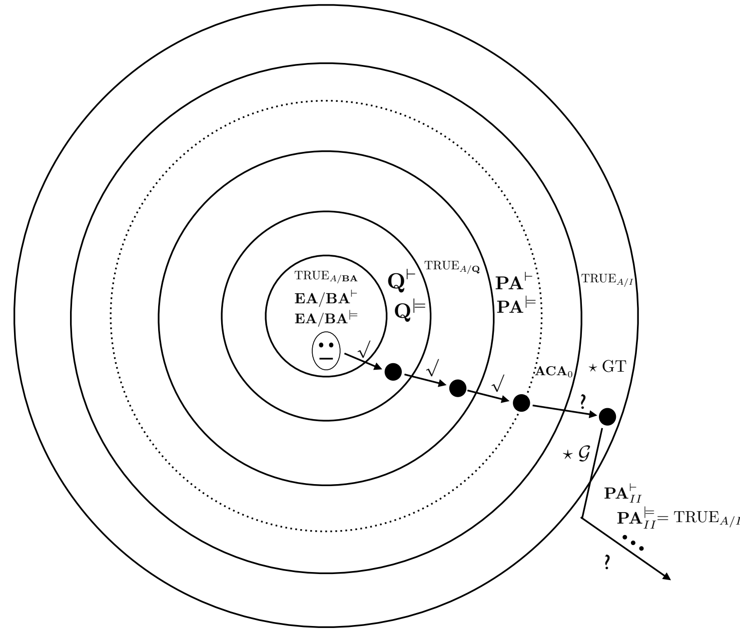
1 Introduction

1.1 Automating the Proof of G1

2 The Analogico-Deductive Approach

2.1 Conjecture Generation

G1



$\mathcal{G}$

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
Rensselaer Polytechnic Institute

A Context: Infinitary Reasoning, Hypercomputation, and Humble Engineering

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^0_1 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 Gittelson
Rensselaer Polytechnic Institute
Troy, NY
{licatj,govinn,selmer,pomerm,gittel}@rpi.edu

Abstract

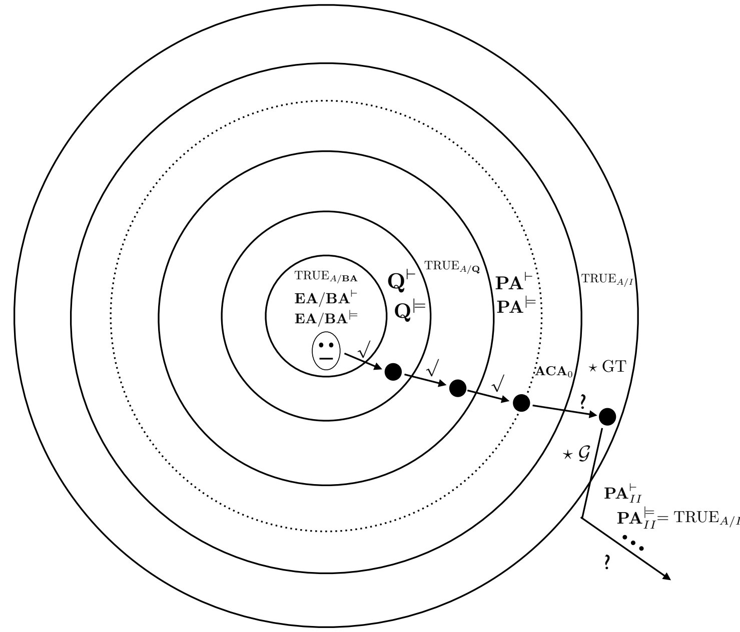
1 Introduction

1.1 Automating the Proof of G1

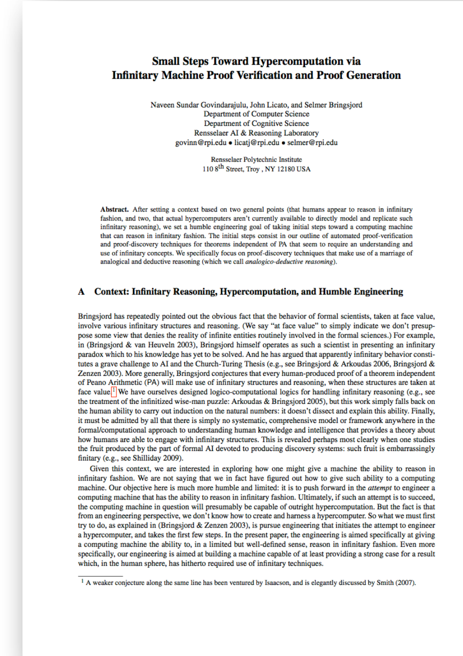
2 The Analogico-Deductive Approach

2.1 Conjecture Generation

G1



$\mathcal{G}$


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

A Context: Infinitary Reasoning, Hypercomputation, and Humble Engineering

Gödel's Speedup Theorem

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

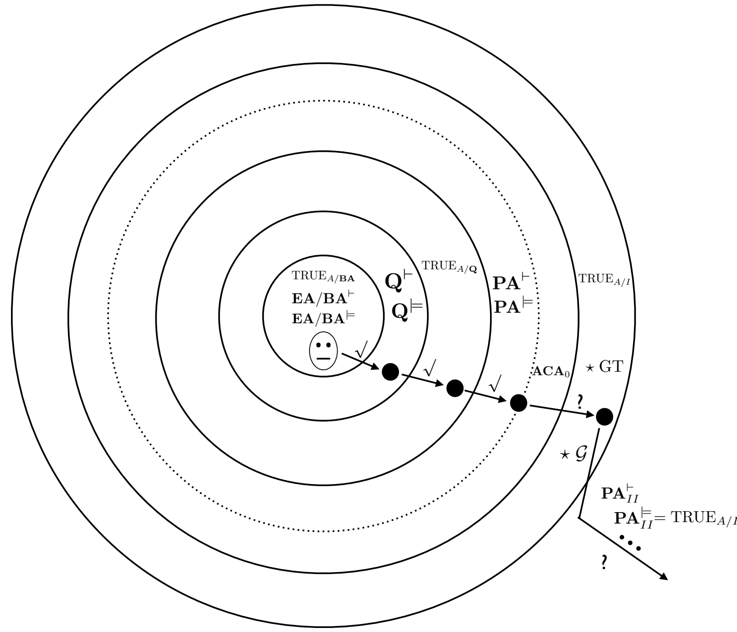
Then there is an infinite family $\mathcal{F}$ of Π^0_1 formulae such that:

1. $\forall \phi \in \mathcal{F}, Z_i \vdash \phi$; and
2. $\forall \phi \in \mathcal{F}$, if k is the least integer s.t. $Z_{i+1} \vdash^{k \text{ symbols}} \phi$, then $Z_i \nvdash^{f(k) \text{ symbols}} \phi$.


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

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

G1



$\mathcal{G}$


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

Naveen Sundar Govindarajulu, John Licato, and Selmer Bringsjord

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^0_1 formulae such that:

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

Can we build an AI that can prove this??

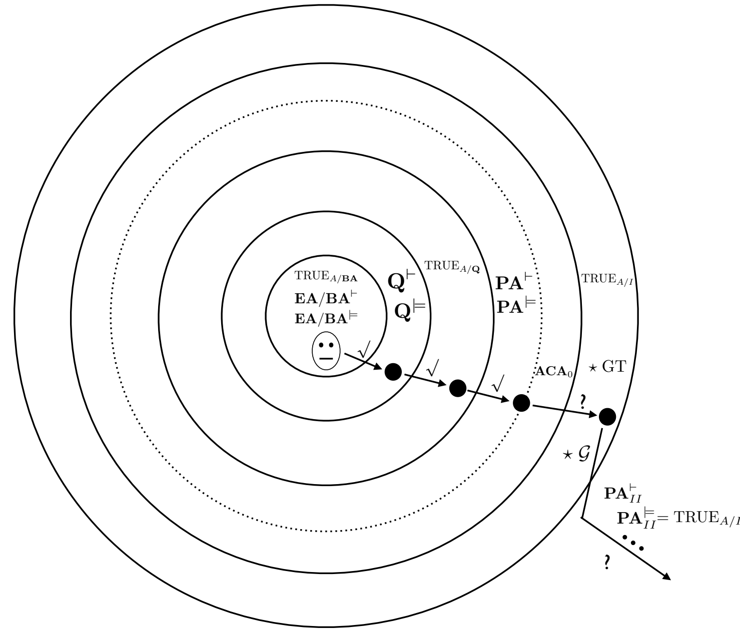
Analogico-Deductive Generation of Gödel's First Incompleteness Theorem from the Liar Paradox

John Licato, Naveen Sundar Govindarajulu, Selmer Bringsjord, Michael Pomeranz, Logan Gittelson

Rensselaer Polytechnic Institute

Troy, NY

G1



$\mathcal{G}$

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

Naveen Sundar Govindarajulu, John Licato, and Selmer Bringsjord

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^0_1 formulae such that:

1. $\forall \phi \in \mathcal{F}, Z_i \vdash \phi$; and
2. $\forall \phi \in \mathcal{F}$, if k is the least integer s.t. $Z_{i+1} \vdash^{k \text{ symbols}} \phi$, then $Z_i \nvdash^{f(k) \text{ symbols}} \phi$.

Can we build an AI that can prove this??

Yes. Somehow …

Supererogatory Reflection

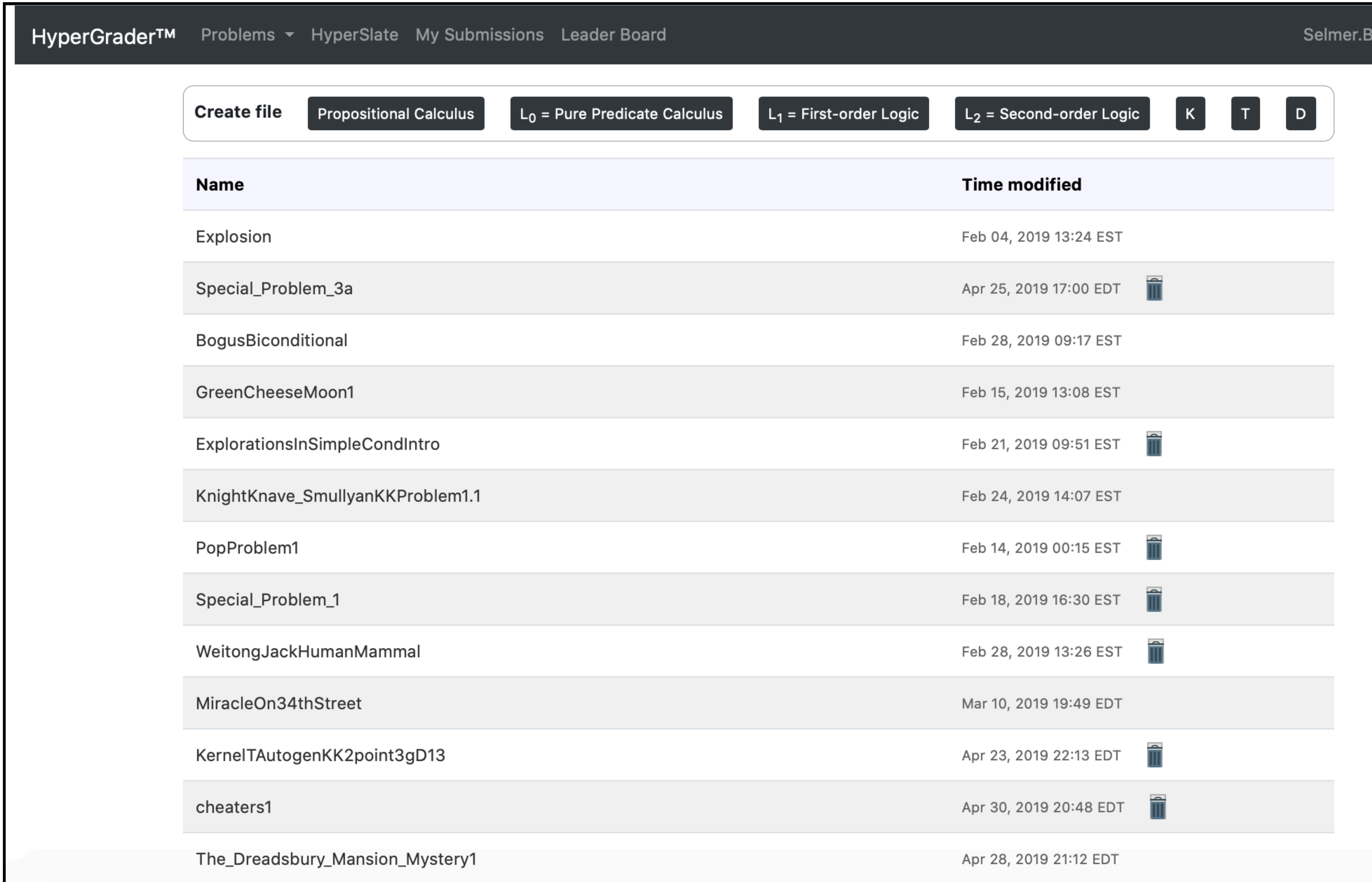
Supererogatory Reflection

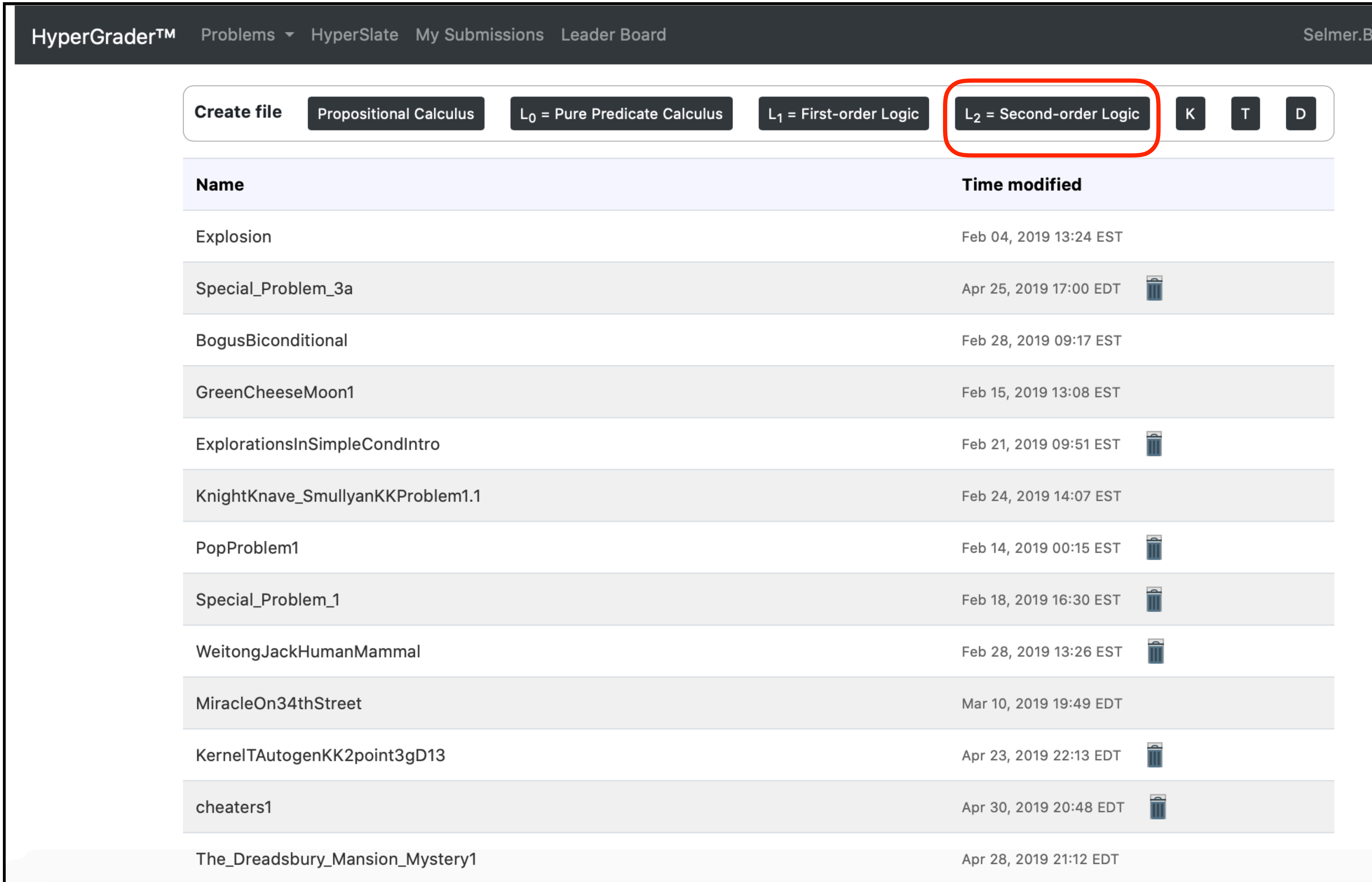
Prove $\vdash_{\mathtt{PA}_1} 1000 \geq 0$; how long is your proof?

Supererogatory Reflection

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

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


HyperGrader™
Problems
HyperSlate
My Submissions
Leader Board
Selmer.Br
Create file
Propositional Calculus
L0 = Pure Predicate Calculus
L1 = First-order Logic
L2 = Second-order Logic
K
T
D
Name
Time modified
Explosion
Feb 04, 2019 13:24 EST
Special_Problem_3a
Apr 25, 2019 17:00 EDT
BogusBiconditional
Feb 28, 2019 09:17 EST
GreenCheeseMoon1
Feb 15, 2019 13:08 EST
ExplorationsInSimpleCondIntro
Feb 21, 2019 09:51 EST
KnightKnave_SmullyanKKProblem1.1
Feb 24, 2019 14:07 EST
PopProblem1
Feb 14, 2019 00:15 EST
Special_Problem_1
Feb 18, 2019 16:30 EST
WeitongJackHumanMammal
Feb 28, 2019 13:26 EST
MiracleOn34thStreet
Mar 10, 2019 19:49 EDT
KernelTAutogenKK2point3gD13
Apr 23, 2019 22:13 EDT
cheaters1
Apr 30, 2019 20:48 EDT
The_Dreadsbury_Mansion_Mystery1
Apr 28, 2019 21:12 EDT


HyperGrader™
Problems
HyperSlate
My Submissions
Leader Board
Selmer.Br
Create file
Propositional Calculus
L_0 = Pure Predicate Calculus
L_1 = First-order Logic
L_2 = Second-order Logic
K
T
D
Name
Time modified
Explosion
Feb 04, 2019 13:24 EST
Special_Problem_3a
Apr 25, 2019 17:00 EDT
BogusBiconditional
Feb 28, 2019 09:17 EST
GreenCheeseMoon1
Feb 15, 2019 13:08 EST
ExplorationsInSimpleCondIntro
Feb 21, 2019 09:51 EST
KnightKnave_SmullyanKKProblem1.1
Feb 24, 2019 14:07 EST
PopProblem1
Feb 14, 2019 00:15 EST
Special_Problem_1
Feb 18, 2019 16:30 EST
WeitongJackHumanMammal
Feb 28, 2019 13:26 EST
MiracleOn34thStreet
Mar 10, 2019 19:49 EDT
KernelTAutogenKK2point3gD13
Apr 23, 2019 22:13 EDT
cheaters1
Apr 30, 2019 20:48 EDT
The_Dreadsbury_Mansion_Mystery1
Apr 28, 2019 21:12 EDT

Slutten