Gödel's SpeedupTheorem (GST)

James Oswald & 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

IFLAI2
Sep 24 2023

R A I R
Rensselaer AI and Reasoning Lab

Gödel's Speedup Theorem
(GST)

James Oswald & 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

IFLAI2
Sep 24 2023

Gödel's Speedup Theorem (GST)

James Oswald & 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

IFLAI2
Sep 24 2023

Gödel's SpeedupTheorem (GST)

James Oswald & 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

IFLAI2
Sep 24 2023

RAIR
Rensselaer AI and Reasoning Lab

OXFORD UNIVERSITY PRESS

Gödel's Speedup Theorem
(GST)

James Oswald & 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

IFLAI2
Sep 24 2023

Note: This is a version designed for those who have had at least one serious, proof-intensive university-level course in formal logic.

Background Context …

Gödel's Great Theorems (OUP)

by Selmer Bringsjord

- Introduction ("The Wager")
- Brief Preliminaries (e.g. the propositional calculus & FOL)
- The Completeness Theorem
- The First Incompleteness Theorem
- The Second Incompleteness Theorem
- The Speedup Theorem
- The Continuum-Hypothesis Theorem
- The Time-Travel Theorem
- Gödel's "God Theorem"
- Could a Finite Machine Match Gödel's Greatness?

STOP & REVIEW IF NEEDED!

Gödel's Great Theorems (OUP)

by Selmer Bringsjord

- Introduction ("The Wager")
- Brief Preliminaries (e.g. the propositional calculus & FOL)
- The Completeness Theorem
- The First Incompleteness Theorem
- The Second Incompleteness Theorem
- The Speedup Theorem
- The Continuum-Hypothesis Theorem
- The Time-Travel Theorem
- Gödel's "God Theorem"
- Could a Finite Machine Match Gödel's Greatness?

STOP & REVIEW IF NEEDED!

Gödel's Great Theorems (OUP)

by Selmer Bringsjord

√
- Introduction ("The Wager")
- Brief Preliminaries (e.g. the propositional calculus & FOL)
- The Completeness Theorem
- The First Incompleteness Theorem
- The Second Incompleteness Theorem
- The Speedup Theorem
- The Continuum-Hypothesis Theorem
- The Time-Travel Theorem
- Gödel's "God Theorem"
- Could a Finite Machine Match Gödel's Greatness?

STOP & REVIEW IF NEEDED!

Gödel's Great Theorems (OUP)

by Selmer Bringsjord

- √ Introduction ("The Wager")
- √ Brief Preliminaries (e.g. the propositional calculus & FOL)
- The Completeness Theorem
- The First Incompleteness Theorem
- The Second Incompleteness Theorem
- The Speedup Theorem
- The Continuum-Hypothesis Theorem
- The Time-Travel Theorem
- Gödel's "God Theorem"
- Could a Finite Machine Match Gödel's Greatness?

STOP & REVIEW IF NEEDED!

Gödel's Great Theorems (OUP)

by Selmer Bringsjord

- √ Introduction ("The Wager")
- √ Brief Preliminaries (e.g. the propositional calculus & FOL)
- √ The Completeness Theorem
- The First Incompleteness Theorem
- The Second Incompleteness Theorem
- The Speedup Theorem
- The Continuum-Hypothesis Theorem
- The Time-Travel Theorem
- Gödel's "God Theorem"
- Could a Finite Machine Match Gödel's Greatness?

STOP & REVIEW IF NEEDED!

Gödel's Great Theorems (OUP)

by Selmer Bringsjord

- √ Introduction ("The Wager")
- √ Brief Preliminaries (e.g. the propositional calculus & FOL)
- √ The Completeness Theorem
- √ The First Incompleteness Theorem
- The Second Incompleteness Theorem
- The Speedup Theorem
- The Continuum-Hypothesis Theorem
- The Time-Travel Theorem
- Gödel's "God Theorem"
- Could a Finite Machine Match Gödel's Greatness?

STOP & REVIEW IF NEEDED!

Gödel's Great Theorems (OUP)

by Selmer Bringsjord

√ • Introduction ("The Wager")
√ • Brief Preliminaries (e.g. the propositional calculus & FOL)
√ • The Completeness Theorem
√ • The First Incompleteness Theorem
√ • The Second Incompleteness Theorem
- The Speedup Theorem
- The Continuum-Hypothesis Theorem
- The Time-Travel Theorem
- Gödel's "God Theorem"
- Could a Finite Machine Match Gödel's Greatness?

STOP & REVIEW IF NEEDED!

Gödel's Great Theorems (OUP)

by Selmer Bringsjord

- √ Introduction ("The Wager")
- √ Brief Preliminaries (e.g. the propositional calculus & FOL)
- √ The Completeness Theorem
- √ The First Incompleteness Theorem
- √ The Second Incompleteness Theorem
- → The Speedup Theorem
- The Continuum-Hypothesis Theorem
- The Time-Travel Theorem
- Gödel's "God Theorem"
- Could a Finite Machine Match Gödel's Greatness?

Switching to more expressive logics can produce a level of speedup beyond the reaching of standard computation.

STOP & REVIEW IF NEEDED!

Gödel's Great Theorems (OUP)

by Selmer Bringsjord

- √ Introduction ("The Wager")
- √ Brief Preliminaries (e.g. the propositional calculus & FOL)
- √ The Completeness Theorem
- √ The First Incompleteness Theorem
- √ The Second Incompleteness Theorem
- → The Speedup Theorem
- → The Continuum-Hypothesis Theorem
- The Time-Travel Theorem
- Gödel's "God Theorem"
- Could a Finite Machine Match Gödel's Greatness?

Switching to more expressive logics can produce a level of speedup beyond the reaching of standard computation.

By far the greatest of GGT; Selm's analysis based Sherlock Holmes' mystery "Silver Blaze."

Ascending Acceleration

Ascending Acceleration

2 sec: 60 mph 5.5 sec: 100 mph 7.5 sec: 150 mph

Ascending Acceleration

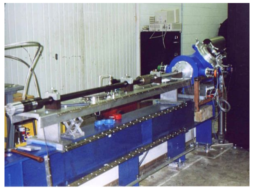
2 sec: 60 mph 5.5 sec: 100 mph 7.5 sec: 150 mph

20 sec: 268 mph 520 sec: 17,000 mph

Ascending Acceleration

2 sec: 60 mph   5.5 sec: 100 mph   7.5 sec: 150 mph

20 sec: 268 mph   520 sec: 17,000 mph



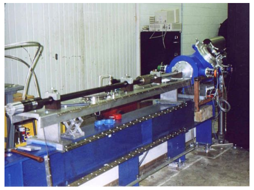
1 sec: 20,000 mph

light-gas gun

Ascending Acceleration

2 sec: 60 mph   5.5 sec: 100 mph   7.5 sec: 150 mph

20 sec: 268 mph   520 sec: 17,000 mph



1 sec: 20,000 mph

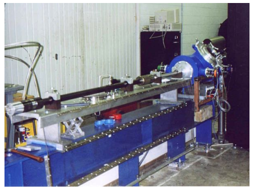
light-gas gun

Primitive Recursion: $h(x,0) = f(x); h(x, y') = g(x, y, h(x, y))$

Ascending Acceleration

2 sec: 60 mph 5.5 sec: 100 mph 7.5 sec: 150 mph

20 sec: 268 mph 520 sec: 17,000 mph



1 sec: 20,000 mph

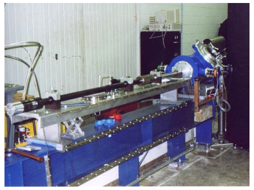
light-gas gun

Primitive Recursion: $h(x,0) = f(x); h(x, y') = g(x, y, h(x, y))$

Ascending Acceleration

2 sec: 60 mph   5.5 sec: 100 mph   7.5 sec: 150 mph

20 sec: 268 mph   520 sec: 17,000 mph



light-gas gun

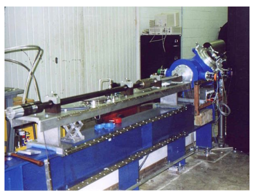
1 sec: 20,000 mph

Primitive Recursion: $h(x,0) = f(x); h(x, y') = g(x, y, h(x, y))$

Ascending Acceleration

2 sec: 60 mph   5.5 sec: 100 mph   7.5 sec: 150 mph

20 sec: 268 mph   520 sec: 17,000 mph



light-gas gun

1 sec: 20,000 mph

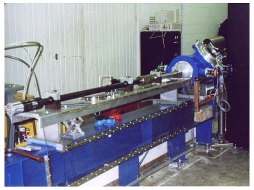
Primitive Recursion: $h(x,0) = f(x); h(x, y') = g(x, y, h(x, y))$

exponentiation: $x^y = x \cdot x \cdot \ldots \cdot x$ (row of y xs)

Ascending Acceleration

2 sec: 60 mph 5.5 sec: 100 mph 7.5 sec: 150 mph

20 sec: 268 mph 520 sec: 17,000 mph



1 sec: 20,000 mph

light-gas gun

Primitive Recursion: $h(x,0) = f(x); h(x,y') = g(x,y,h(x,y))$

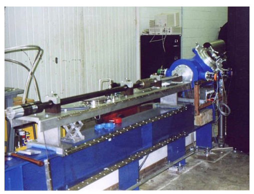
exponentiation: $x^y = x \cdot x \cdot \ldots \cdot x$ (row of y xs)

super-exponentiation (tetration): $x \uparrow (x \uparrow (x \uparrow \ldots \uparrow x))$ (y xs)

Ascending Acceleration

2 sec: 60 mph   5.5 sec: 100 mph   7.5 sec: 150 mph

20 sec: 268 mph   520 sec: 17,000 mph



1 sec: 20,000 mph

light-gas gun

Primitive Recursion: $h(x,0) = f(x); h(x,y') = g(x,y,h(x,y))$

exponentiation: $x^y = x \cdot x \cdot \ldots \cdot x$ (row of y xs)

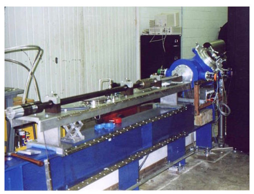
super-exponentiation (tetration): $x \uparrow (x \uparrow (x \uparrow \ldots \uparrow x))$ (y xs)

Ackermann Function

Ascending Acceleration

2 sec: 60 mph 5.5 sec: 100 mph 7.5 sec: 150 mph

20 sec: 268 mph 520 sec: 17,000 mph



light-gas gun

1 sec: 20,000 mph

Primitive Recursion: $h(x,0) = f(x); h(x, y') = g(x, y, h(x, y))$

exponentiation: $x^y = x \cdot x \cdot \ldots \cdot x$ (row of y xs)

super-exponentiation (tetration): $x \uparrow (x \uparrow (x \uparrow \ldots \uparrow x))$ (y xs)

Ackermann Function

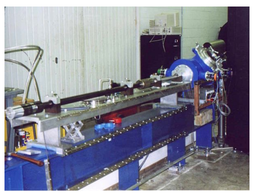
$\alpha(x, y, z) = x\langle y\rangle z$ and $\gamma(x) = \alpha(x, x, x)$; then:

$$\gamma(0) = 0 + 0 = 0$$
$$\gamma(1) = 1 \cdot 1 = 0$$
$$\gamma(2) = 2^2 = 4$$
$$\gamma(3) = 3^{3^3} = 3 \uparrow\uparrow 3 = 7,625,597,484,987$$
$$\gamma(4) = 4 \uparrow\uparrow 4 => 10^{1000} (\text{note: } 10^{100} \text{ is googol})$$

Ascending Acceleration

2 sec: 60 mph   5.5 sec: 100 mph   7.5 sec: 150 mph

20 sec: 268 mph   520 sec: 17,000 mph



light-gas gun

1 sec: 20,000 mph

Primitive Recursion: $h(x,0) = f(x); h(x, y') = g(x, y, h(x, y))$

---

exponentiation: $x^y = x \cdot x \cdot \ldots \cdot x$ (row of y xs)

super-exponentiation (tetration): $x \uparrow (x \uparrow (x \uparrow \ldots \uparrow x))$ (y xs)

Ackermann Function

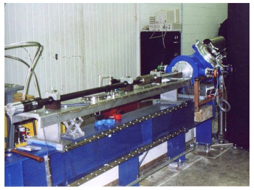
$\alpha(x, y, z) = x\langle y\rangle z$ and $\gamma(x) = \alpha(x, x, x)$; then:

$\gamma(0) = 0 + 0 = 0$
$\gamma(1) = 1 \cdot 1 = 0$
$\gamma(2) = 2^2 = 4$
$\gamma(3) = 3^{3^3} = 3 \uparrow\uparrow 3 = 7,625,597,484,987$
$\gamma(4) = 4 \uparrow\uparrow 4 => 10^{1000}$(note: 10^{100} is googol)

Ascending Acceleration

2 sec: 60 mph 5.5 sec: 100 mph 7.5 sec: 150 mph

20 sec: 268 mph 520 sec: 17,000 mph



1 sec: 20,000 mph

light-gas gun

Primitive Recursion: $h(x,0) = f(x); h(x, y') = g(x, y, h(x, y))$

---

exponentiation: $x^y = x \cdot x \cdot \ldots \cdot x$ (row of y xs)

super-exponentiation (tetration): $x \uparrow (x \uparrow (x \uparrow \ldots \uparrow x))$ (y xs)

$\alpha(x, y, z) = x\langle y\rangle z$ and $\gamma(x) = \alpha(x, x, x)$; then:

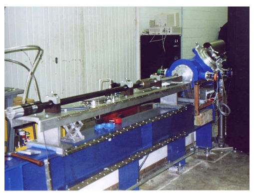
Ackermann Function

$$\gamma(0) = 0 + 0 = 0$$
$$\gamma(1) = 1 \cdot 1 = 0$$
$$\gamma(2) = 2^2 = 4$$
$$\gamma(3) = 3^{3^3} = 3 \uparrow\uparrow 3 = 7,625,597,484,987$$
$$\gamma(4) = 4 \uparrow\uparrow 4 => 10^{1000} (\text{note: } 10^{100} \text{ is googol})$$

Ascending Acceleration

2 sec: 60 mph   5.5 sec: 100 mph   7.5 sec: 150 mph

20 sec: 268 mph   520 sec: 17,000 mph



light-gas gun

1 sec: 20,000 mph

Primitive Recursion: $h(x,0) = f(x); h(x, y') = g(x, y, h(x, y))$

---

exponentiation: $x^y = x \cdot x \cdot \ldots \cdot x$ (row of y xs)

super-exponentiation (tetration): $x \uparrow (x \uparrow (x \uparrow \ldots \uparrow x))$ (y xs)

$\alpha(x, y, z) = x\langle y \rangle z$ and $\gamma(x) = \alpha(x, x, x)$; then:

Ackermann Function

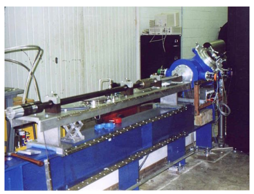
$\gamma(0) = 0 + 0 = 0$
$\gamma(1) = 1 \cdot 1 = 0$
$\gamma(2) = 2^2 = 4$
$\gamma(3) = 3^{3^3} = 3 \uparrow\uparrow 3 = 7,625,597,484,987$
$\gamma(4) = 4 \uparrow\uparrow 4 => 10^{1000}$(note: 10^{100} is googol)

$\Sigma : \mathbb{Z}^+ \mapsto \mathbb{Z}^+$ where $\Sigma(k) =$ max productivity of a k-state TM

Ascending Acceleration

2 sec: 60 mph   5.5 sec: 100 mph   7.5 sec: 150 mph

20 sec: 268 mph   520 sec: 17,000 mph



light-gas gun

1 sec: 20,000 mph

Primitive Recursion: $h(x,0) = f(x); h(x, y') = g(x, y, h(x, y))$

exponentiation: $x^y = x \cdot x \cdot \ldots \cdot x$ (row of y xs)

super-exponentiation (tetration): $x \uparrow (x \uparrow (x \uparrow \ldots \uparrow x))$ (y xs)

Ackermann Function

$\alpha(x, y, z) = x\langle y\rangle z$ and $\gamma(x) = \alpha(x, x, x)$; then:

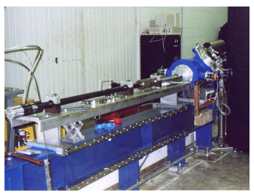
$\gamma(0) = 0 + 0 = 0$
$\gamma(1) = 1 \cdot 1 = 0$
$\gamma(2) = 2^2 = 4$
$\gamma(3) = 3^{3^3} = 3 \uparrow\uparrow 3 = 7,625,597,484,987$
$\gamma(4) = 4 \uparrow\uparrow 4 => 10^{1000}$(note: 10^{100} is googol)

$\Sigma : \mathbb{Z}^+ \mapsto \mathbb{Z}^+$ where $\Sigma(k) =$ max productivity of a k-state TM

Ascending Acceleration

2 sec: 60 mph 5.5 sec: 100 mph 7.5 sec: 150 mph

20 sec: 268 mph 520 sec: 17,000 mph



light-gas gun

1 sec: 20,000 mph

Primitive Recursion: $h(x,0) = f(x); h(x, y') = g(x, y, h(x, y))$

exponentiation: $x^y = x \cdot x \cdot \ldots \cdot x$ (row of y xs)

super-exponentiation (tetration): $x \uparrow (x \uparrow (x \uparrow \ldots \uparrow x))$ (y xs)

Ackermann Function

$\alpha(x, y, z) = x\langle y\rangle z$ and $\gamma(x) = \alpha(x, x, x)$; then:

$$\gamma(0) = 0 + 0 = 0$$
$$\gamma(1) = 1 \cdot 1 = 0$$
$$\gamma(2) = 2^2 = 4$$
$$\gamma(3) = 3^{3^3} = 3 \uparrow\uparrow 3 = 7,625,597,484,987$$
$$\gamma(4) = 4 \uparrow\uparrow 4 => 10^{1000} (\text{note: } 10^{100} \text{ is googol})$$

$\Sigma : \mathbb{Z}^+ \mapsto \mathbb{Z}^+$ where $\Sigma(k) =$ max productivity of a k-state TM

Climbing Again the *k*-order Ladder

Climbing Again 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 Again 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 Again the *k*-order Ladder

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

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

Climbing Again 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 Again 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 Again 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 Again the *k*-order Ladder

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

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 Again the *k*-order Ladder

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 Again the *k*-order Ladder

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

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

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

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

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

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

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

Climbing Again the *k*-order Ladder

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

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 Again the *k*-order Ladder

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

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 Again 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.

Climbing Again 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 $\mathscr{L}_1$

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

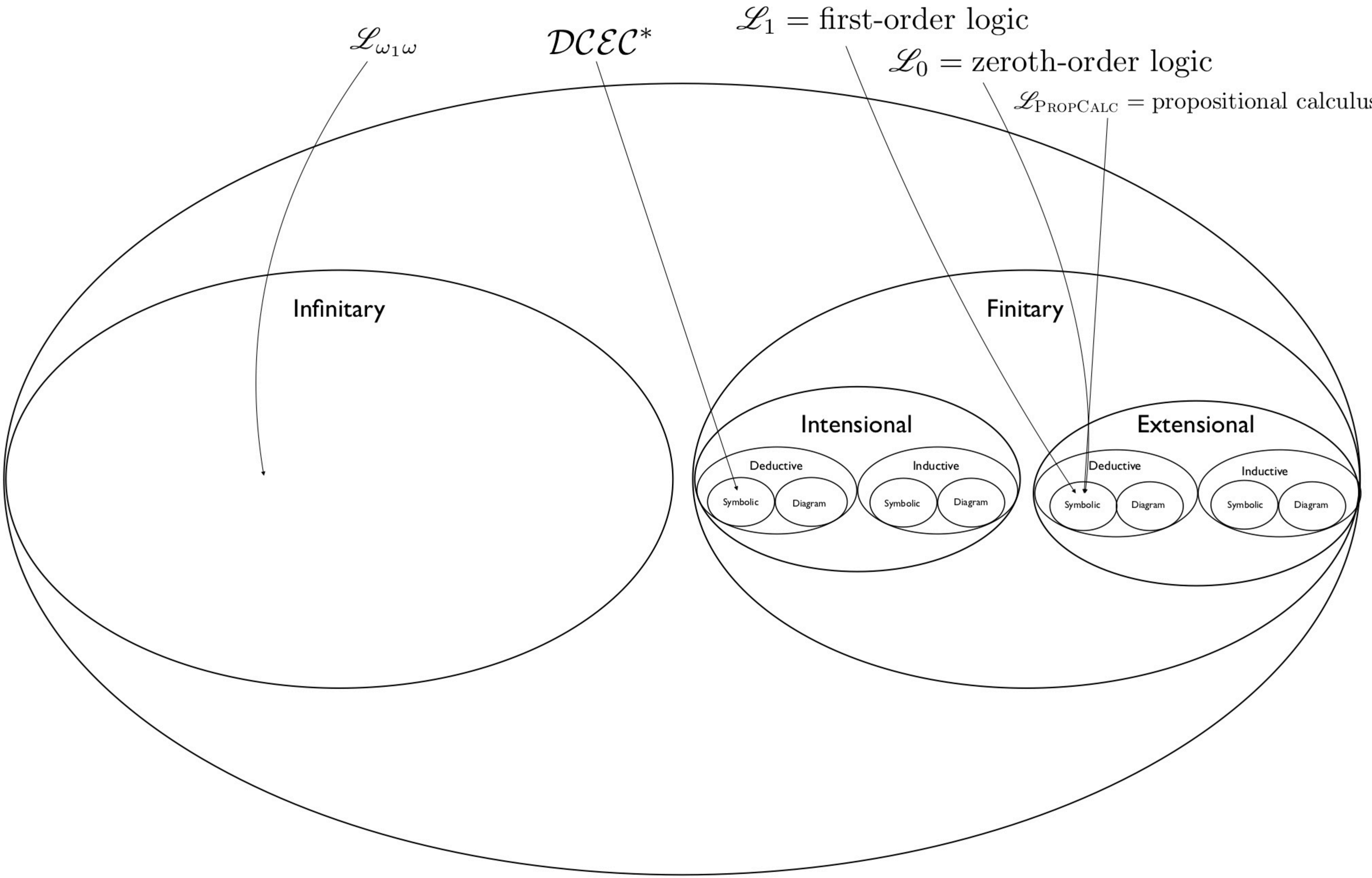
There's some thing which is a llama and likes *b* (which is also a llama), and whose father is a llama too.

ZOL $\mathscr{L}_0$

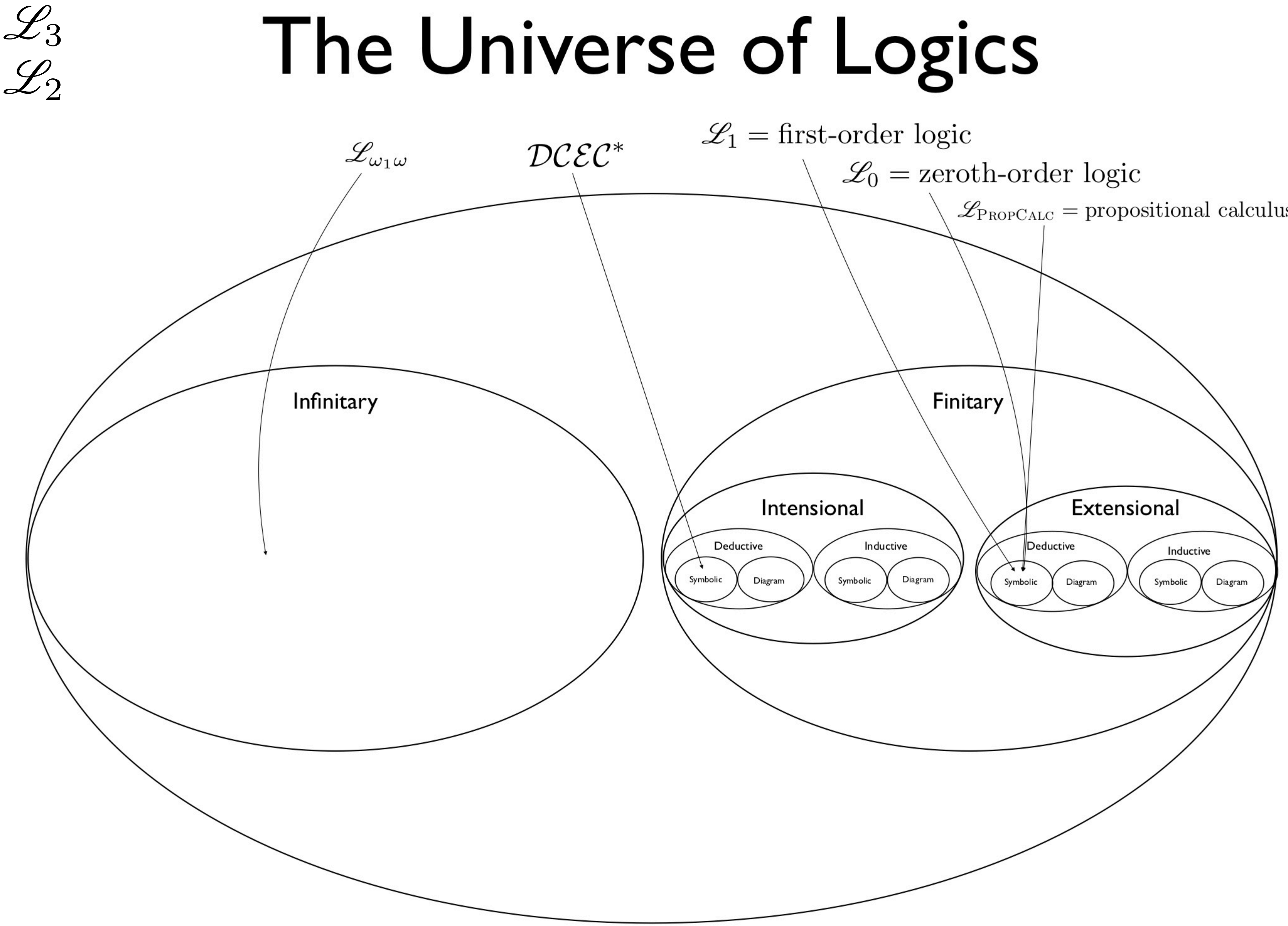
$$Llama(a) \wedge Llama(b) \wedge Likes(a,b) \wedge Llama(fatherOf(a))$$

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

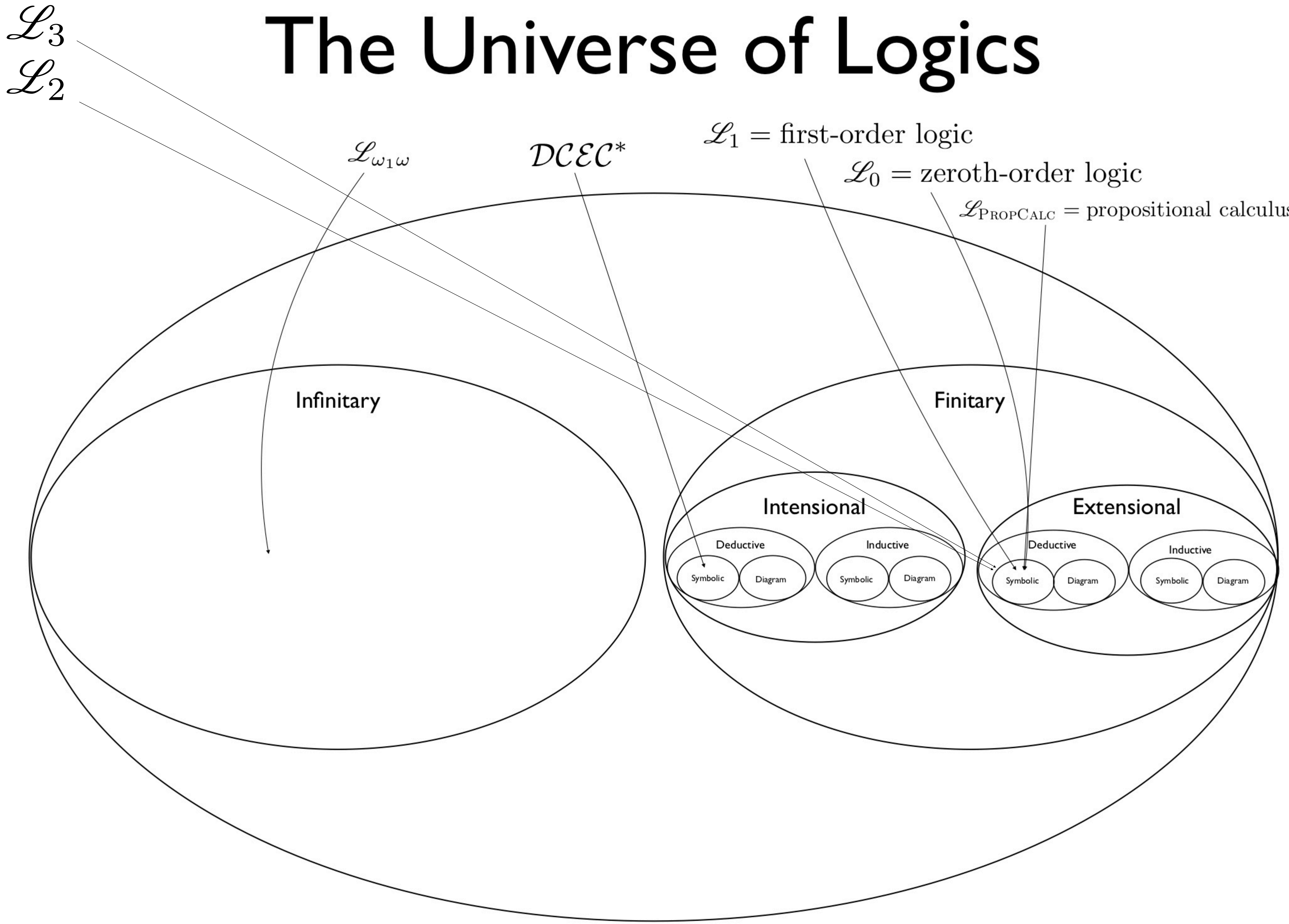
The Universe of Logics



The Universe of Logics



The Universe of Logics



Climbing the *k*-order Ladder

⋮

TOL $\mathscr{L}_3$

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

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.

Gödel's Speedup Theorem

Gödel's Speedup Theorem

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

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:

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 $Z_i \nvdash^{f(k) \text{ symbols}} \phi.$

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 $Z_i \nvdash^{f(k) \text{ symbols}} \phi$.

!

Ascending Acceleration

Ascending Acceleration

2 sec: 60 mph   5.5 sec: 100 mph   7.5 sec: 150 mph

Ascending Acceleration

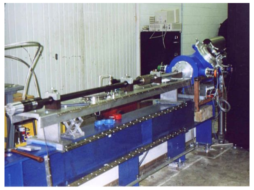
2 sec: 60 mph   5.5 sec: 100 mph   7.5 sec: 150 mph

20 sec: 268 mph   520 sec: 17,000 mph

Ascending Acceleration

2 sec: 60 mph   5.5 sec: 100 mph   7.5 sec: 150 mph

20 sec: 268 mph   520 sec: 17,000 mph



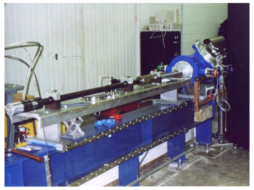
light-gas gun

1 sec: 20,000 mph

Ascending Acceleration

2 sec: 60 mph   5.5 sec: 100 mph   7.5 sec: 150 mph

20 sec: 268 mph   520 sec: 17,000 mph



1 sec: 20,000 mph

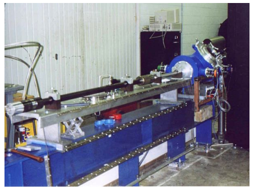
light-gas gun

Primitive Recursion: $h(x,0) = f(x); h(x,y') = g(x,y,h(x,y))$

Ascending Acceleration

2 sec: 60 mph 5.5 sec: 100 mph 7.5 sec: 150 mph

20 sec: 268 mph 520 sec: 17,000 mph



1 sec: 20,000 mph

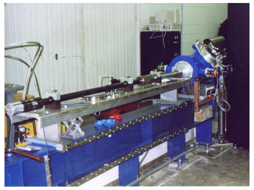
light-gas gun

Primitive Recursion: $h(x,0) = f(x); h(x, y') = g(x, y, h(x, y))$

Ascending Acceleration

2 sec: 60 mph 5.5 sec: 100 mph 7.5 sec: 150 mph

20 sec: 268 mph 520 sec: 17,000 mph



1 sec: 20,000 mph

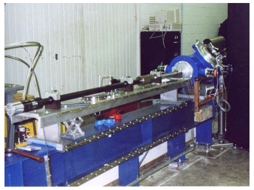
light-gas gun

Primitive Recursion: $h(x,0) = f(x); h(x, y') = g(x, y, h(x, y))$

Ascending Acceleration

2 sec: 60 mph 5.5 sec: 100 mph 7.5 sec: 150 mph

20 sec: 268 mph 520 sec: 17,000 mph



1 sec: 20,000 mph

light-gas gun

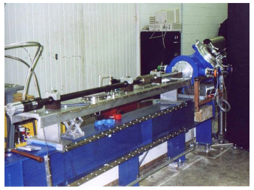
Primitive Recursion: $h(x,0) = f(x); h(x, y') = g(x, y, h(x, y))$

exponentiation: $x^y = x \cdot x \cdot \ldots \cdot x$ (row of y xs)

Ascending Acceleration

2 sec: 60 mph   5.5 sec: 100 mph   7.5 sec: 150 mph

20 sec: 268 mph   520 sec: 17,000 mph



1 sec: 20,000 mph

light-gas gun

Primitive Recursion: $h(x,0) = f(x); h(x,y') = g(x,y,h(x,y))$

---

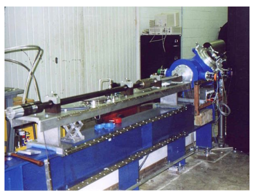
exponentiation: $x^y = x \cdot x \cdot \ldots \cdot x$ (row of y xs)

super-exponentiation (tetration): $x \uparrow (x \uparrow (x \uparrow \ldots \uparrow x))$ (y xs)

Ascending Acceleration

2 sec: 60 mph   5.5 sec: 100 mph   7.5 sec: 150 mph

20 sec: 268 mph   520 sec: 17,000 mph



1 sec: 20,000 mph

light-gas gun

Primitive Recursion: $h(x,0) = f(x); h(x,y') = g(x,y,h(x,y))$

exponentiation: $x^y = x \cdot x \cdot \ldots \cdot x$ (row of y xs)

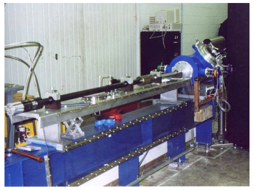
super-exponentiation (tetration): $x \uparrow (x \uparrow (x \uparrow \ldots \uparrow x))$ (y xs)

Ackermann Function

Ascending Acceleration

2 sec: 60 mph   5.5 sec: 100 mph   7.5 sec: 150 mph

20 sec: 268 mph   520 sec: 17,000 mph



light-gas gun

1 sec: 20,000 mph

Primitive Recursion: $h(x,0) = f(x); h(x, y') = g(x, y, h(x, y))$

---

exponentiation: $x^y = x \cdot x \cdot \ldots \cdot x$ (row of y xs)

super-exponentiation (tetration): $x \uparrow (x \uparrow (x \uparrow \ldots \uparrow x))$ (y xs)

Ackermann Function

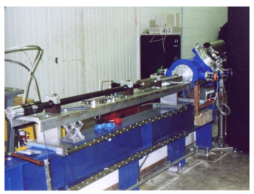
$\alpha(x, y, z) = x\langle y\rangle z$ and $\gamma(x) = \alpha(x, x, x)$; then:

$\gamma(0) = 0 + 0 = 0$
$\gamma(1) = 1 \cdot 1 = 0$
$\gamma(2) = 2^2 = 4$
$\gamma(3) = 3^{3^3} = 3 \uparrow\uparrow 3 = 7,625,597,484,987$
$\gamma(4) = 4 \uparrow\uparrow 4 => 10^{1000}$(note: 10^{100} is googol)

Ascending Acceleration

2 sec: 60 mph 5.5 sec: 100 mph 7.5 sec: 150 mph

20 sec: 268 mph 520 sec: 17,000 mph



1 sec: 20,000 mph

light-gas gun

Primitive Recursion: $h(x,0) = f(x); h(x,y') = g(x,y,h(x,y))$

exponentiation: $x^y = x \cdot x \cdot \ldots \cdot x$ (row of y xs)

super-exponentiation (tetration): $x \uparrow (x \uparrow (x \uparrow \ldots \uparrow x))$ (y xs)

$\alpha(x,y,z) = x\langle y\rangle z$ and $\gamma(x) = \alpha(x,x,x)$; then:

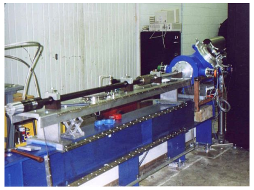
Ackermann Function

$$\gamma(0) = 0 + 0 = 0$$
$$\gamma(1) = 1 \cdot 1 = 0$$
$$\gamma(2) = 2^2 = 4$$
$$\gamma(3) = 3^{3^3} = 3 \uparrow\uparrow 3 = 7,625,597,484,987$$
$$\gamma(4) = 4 \uparrow\uparrow 4 => 10^{1000} (\text{note: } 10^{100} \text{ is googol})$$

Ascending Acceleration

2 sec: 60 mph   5.5 sec: 100 mph   7.5 sec: 150 mph

20 sec: 268 mph   520 sec: 17,000 mph



1 sec: 20,000 mph

light-gas gun

Primitive Recursion: $h(x,0) = f(x); h(x,y') = g(x,y,h(x,y))$

exponentiation: $x^y = x \cdot x \cdot \ldots \cdot x$ (row of y xs)

super-exponentiation (tetration): $x \uparrow (x \uparrow (x \uparrow \ldots \uparrow x))$ (y xs)

Ackermann Function

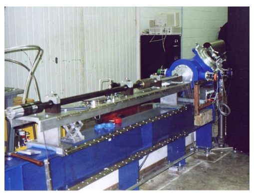
$\alpha(x,y,z) = x\langle y\rangle z$ and $\gamma(x) = \alpha(x,x,x)$; then:

$$\gamma(0) = 0 + 0 = 0$$
$$\gamma(1) = 1 \cdot 1 = 0$$
$$\gamma(2) = 2^2 = 4$$
$$\gamma(3) = 3^{3^3} = 3 \uparrow\uparrow 3 = 7,625,597,484,987$$
$$\gamma(4) = 4 \uparrow\uparrow 4 => 10^{1000} (\text{note: } 10^{100} \text{ is googol})$$

Ascending Acceleration

2 sec: 60 mph 5.5 sec: 100 mph 7.5 sec: 150 mph

20 sec: 268 mph 520 sec: 17,000 mph



light-gas gun

1 sec: 20,000 mph

Primitive Recursion: $h(x,0) = f(x); h(x, y') = g(x, y, h(x, y))$

exponentiation: $x^y = x \cdot x \cdot \ldots \cdot x$ (row of y xs)

super-exponentiation (tetration): $x \uparrow (x \uparrow (x \uparrow \ldots \uparrow x))$ (y xs)

$\alpha(x, y, z) = x\langle y\rangle z$ and $\gamma(x) = \alpha(x, x, x)$; then:

Ackermann Function

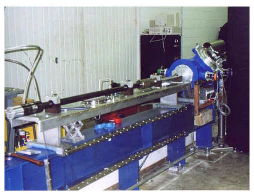
$\gamma(0) = 0 + 0 = 0$
$\gamma(1) = 1 \cdot 1 = 0$
$\gamma(2) = 2^2 = 4$
$\gamma(3) = 3^{3^3} = 3 \uparrow\uparrow 3 = 7,625,597,484,987$
$\gamma(4) = 4 \uparrow\uparrow 4 => 10^{1000}$(note: 10^{100} is googol)

$\Sigma : \mathbb{Z}^+ \mapsto \mathbb{Z}^+$ where $\Sigma(k) =$ max productivity of a k-state TM

Ascending Acceleration

2 sec: 60 mph   5.5 sec: 100 mph   7.5 sec: 150 mph

20 sec: 268 mph   520 sec: 17,000 mph



1 sec: 20,000 mph

light-gas gun

Primitive Recursion: $h(x,0) = f(x); h(x, y') = g(x, y, h(x, y))$

---

exponentiation: $x^y = x \cdot x \cdot \ldots \cdot x$ (row of y xs)

super-exponentiation (tetration): $x \uparrow (x \uparrow (x \uparrow \ldots \uparrow x))$ (y xs)

$\alpha(x, y, z) = x\langle y\rangle z$ and $\gamma(x) = \alpha(x, x, x)$; then:

Ackermann Function

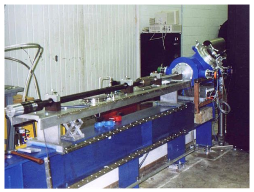
$\gamma(0) = 0 + 0 = 0$
$\gamma(1) = 1 \cdot 1 = 0$
$\gamma(2) = 2^2 = 4$
$\gamma(3) = 3^{3^3} = 3 \uparrow\uparrow 3 = 7,625,597,484,987$
$\gamma(4) = 4 \uparrow\uparrow 4 => 10^{1000}$(note: 10^{100} is googol)

$\Sigma : \mathbb{Z}^+ \mapsto \mathbb{Z}^+$ where $\Sigma(k) =$ max productivity of a k-state TM

Ascending Acceleration

2 sec: 60 mph   5.5 sec: 100 mph   7.5 sec: 150 mph

20 sec: 268 mph   520 sec: 17,000 mph



1 sec: 20,000 mph

light-gas gun

Primitive Recursion: $h(x,0) = f(x); h(x,y') = g(x,y,h(x,y))$

---

exponentiation: $x^y = x \cdot x \cdot \ldots \cdot x$ (row of y xs)

super-exponentiation (tetration): $x \uparrow (x \uparrow (x \uparrow \ldots \uparrow x))$ (y xs)

$\alpha(x,y,z) = x\langle y\rangle z$ and $\gamma(x) = \alpha(x,x,x)$; then:

Ackermann Function

$\gamma(0) = 0 + 0 = 0$
$\gamma(1) = 1 \cdot 1 = 0$
$\gamma(2) = 2^2 = 4$
$\gamma(3) = 3^{3^3} = 3 \uparrow\uparrow 3 = 7,625,597,484,987$
$\gamma(4) = 4 \uparrow\uparrow 4 => 10^{1000}$(note: 10^{100} is googol)

$\Sigma : \mathbb{Z}^+ \mapsto \mathbb{Z}^+$ where $\Sigma(k) =$ max productivity of a k-state TM

The Received View in AI

Expressiveness and tractability in knowledge representation and reasoning[1]

HECTOR J. LEVESQUE[2]
Department of Computer Science, University of Toronto, Toronto, Ont., Canada M5S 1A4

AND

RONALD J. BRACHMAN
AT&T Bell Laboratories, 600 Mountain Avenue, 3C-439, Murray Hill, NJ 07974, U.S.A.

Received November 3, 1986
Revision accepted April 8, 1987

A fundamental computational limit on automated reasoning and its effect on knowledge representation is examined. Basically, the problem is that it can be more difficult to reason correctly with one representational language than with another and, moreover, that this difficulty increases dramatically as the expressive power of the language increases. This leads to a tradeoff between the expressiveness of a representational language and its computational tractability. Here we show that this tradeoff can be seen to underlie the differences among a number of existing representational formalisms, in addition to motivating many of the current research issues in knowledge representation.

Key words: knowledge representation, description subsumption, complexity of reasoning, first-order logic, frames, semantic networks, databases.

Cet article étudie une limitation computationnelle fondamentale du raisonnement automatique et examine ses effets sur la représentation de connaissances. A la base le problème tient en ce qu'il peut être plus difficile de raisonner avec un langage de représentation qu'avec un autre et que cette difficulté augmente considérablement à mesure que croît le pouvoir expressif du langage. Ceci donne lieu à un compromis entre le pouvoir expressif d'un langage de représentation et sa tractibilité computationnelle. Nous montrons que ce compromis peut être vu comme l'une des causes fondamentales de la différence qui existe entre nombre de formalismes de représentation existants et peut motiver plusieurs recherches courantes en représentation de connaissances.

Mots clés : représentation de connaissances, complexité du raisonnement, logique du premier ordre, schémas, réseaux sémantiques, bases de données.

[Traduit par la revue]

Comput. Intell. 3, 78–93 (1987)

1. Introduction

This paper examines from a general point of view a basic computational limit on automated reasoning, and the effect that it has on knowledge representation (KR). The problem is essentially that it can be more difficult to reason correctly with one representational language than with another and, moreover, that this difficulty increases as the expressive power of the language increases. There is a tradeoff between the expressiveness of a representational language and its computational tractability. What we attempt to show is that this tradeoff underlies differences among a number of representational formalisms (such as first-order logic, databases, semantic networks, and frames) and motivates many current research issues in KR (such as the role of analogues, syntactic encodings, and defaults, as well as systems of limited inference and hybrid reasoning).

To deal with such a broad range of representational phenomena we must, of necessity, take a considerably simplified and incomplete view of KR. In particular, we focus on its computational and logical aspects, more or less ignoring its history and relevance in the areas of psychology, linguistics, and philosophy. The area of KR is still very disconnected today and the role of logic remains quite controversial, despite what this paper may suggest. We do believe, however, that the tradeoff discussed here is fundamental. As long as we are dealing with computational systems that reason automatically (without any special intervention or advice) and correctly (once we define what *that* means), we will be able to locate where they stand on the tradeoff: They will either be limited in what knowledge they can represent or unlimited in the reasoning effort they might require.

Our computational focus will not lead us to investigate specific algorithms and data structures for KR and reasoning, however. What we discuss is something much stronger, namely, whether or not algorithms of a certain kind can exist at all. The analysis here is at the *knowledge level* (Newell 1981) where we look at the content of what is represented (in terms of what it says about the world) and not the symbolic structures used to represent that knowledge. Indeed, we examine specific representation schemes in terms of what knowledge they can represent, rather than in terms of how they might actually represent it.

In the next section, we discuss what a KR system is for and what it could mean to reason correctly. Next, we investigate how a KR service might be realized using theorem proving in

[1] This is a revised and substantially augmented version of "A Fundamental Tradeoff in Knowledge Representation and Reasoning," by Hector J. Levesque, which appeared in the Proceedings of the Canadian Society for Computational Studies of Intelligence Conference, London, Ontario, May 1984. It includes portions of two other conference papers: "The Tractability of Subsumption in Frame-Based Description Languages," by Ronald J. Brachman and Hector J. Levesque, which appeared in the Proceedings of the American Association for Artificial Intelligence Conference, Austin, Texas, August 1984; and "What Makes a Knowledge Base Knowledgeable? A View of Databases from the Knowledge Level," by the same authors, which appeared in the Proceedings of the First International Workshop on Expert Database Systems, Kiawah Island, South Carolina, October 1984. Much of this paper appeared as a chapter in *Readings in Knowledge Representation* (Morgan Kaufmann Publishers Inc., 1985), edited by the authors.

[2] Fellow of the Canadian Institute for Advanced Research.

The Received View in AI

1987

Expressiveness and tractability in knowledge representation and reasoning[1]

HECTOR J. LEVESQUE[2]
Department of Computer Science, University of Toronto, Toronto, Ont., Canada M5S 1A4

AND

RONALD J. BRACHMAN
AT&T Bell Laboratories, 600 Mountain Avenue, 3C-439, Murray Hill, NJ 07974, U.S.A.

Received November 3, 1986
Revision accepted April 8, 1987

A fundamental computational limit on automated reasoning and its effect on knowledge representation is examined. Basically, the problem is that it can be more difficult to reason correctly with one representational language than with another and, moreover, that this difficulty increases dramatically as the expressive power of the language increases. This leads to a tradeoff between the expressiveness of a representational language and its computational tractability. Here we show that this tradeoff can be seen to underlie the differences among a number of existing representational formalisms, in addition to motivating many of the current research issues in knowledge representation.

Key words: knowledge representation, description subsumption, complexity of reasoning, first-order logic, frames, semantic networks, databases.

Cet article étudie une limitation computationnelle fondamentale du raisonnement automatique et examine ses effets sur la représentation de connaissances. A la base le problème tient en ce qu'il peut être plus difficile de raisonner avec un langage de représentation qu'avec un autre et que cette difficulté augmente considérablement à mesure que croît le pouvoir expressif du langage. Ceci donne lieu à un compromis entre le pouvoir expressif d'un langage de représentation et sa tractibilité computationnelle. Nous montrons que ce compromis peut être vu comme l'une des causes fondamentales de la différence qui existe entre nombre de formalismes de représentation existants et peut motiver plusieurs recherches courantes en représentation de connaissances.

Mots clés : représentation de connaissances, complexité du raisonnement, logique du premier ordre, schémas, réseaux sémantiques, bases de données.

[Traduit par la revue]

Comput. Intell. 3, 78–93 (1987)

1. Introduction

This paper examines from a general point of view a basic computational limit on automated reasoning, and the effect that it has on knowledge representation (KR). The problem is essentially that it can be more difficult to reason correctly with one representational language than with another and, moreover, that this difficulty increases as the expressive power of the language increases. There is a tradeoff between the expressiveness of a representational language and its computational tractability. What we attempt to show is that this tradeoff underlies differences among a number of representational formalisms (such as first-order logic, databases, semantic networks, and frames) and motivates many current research issues in KR (such as the role of analogues, syntactic encodings, and defaults, as well as systems of limited inference and hybrid reasoning).

To deal with such a broad range of representational phenomena we must, of necessity, take a considerably simplified and incomplete view of KR. In particular, we focus on its computational and logical aspects, more or less ignoring its history and relevance in the areas of psychology, linguistics, and philosophy. The area of KR is still very disconnected today and the role of logic remains quite controversial, despite what this paper may suggest. We do believe, however, that the tradeoff discussed here is fundamental. As long as we are dealing with computational systems that reason automatically (without any special intervention or advice) and correctly (once we define what *that* means), we will be able to locate where they stand on the tradeoff: They will either be limited in what knowledge they can represent or unlimited in the reasoning effort they might require.

Our computational focus will not lead us to investigate specific algorithms and data structures for KR and reasoning, however. What we discuss is something much stronger, namely, whether or not algorithms of a certain kind can exist at all. The analysis here is at the *knowledge level* (Newell 1981) where we look at the content of what is represented (in terms of what it says about the world) and not the symbolic structures used to represent that knowledge. Indeed, we examine specific representation schemes in terms of what knowledge they can represent, rather than in terms of how they might actually represent it.

In the next section, we discuss what a KR system is for and what it could mean to reason correctly. Next, we investigate how a KR service might be realized using theorem proving in

[1] This is a revised and substantially augmented version of "A Fundamental Tradeoff in Knowledge Representation and Reasoning," by Hector J. Levesque, which appeared in the Proceedings of the Canadian Society for Computational Studies of Intelligence Conference, London, Ontario, May 1984. It includes portions of two other conference papers: "The Tractability of Subsumption in Frame-Based Description Languages," by Ronald J. Brachman and Hector J. Levesque, which appeared in the Proceedings of the American Association for Artificial Intelligence Conference, Austin, Texas, August 1984; and "What Makes a Knowledge Base Knowledgeable? A View of Databases from the Knowledge Level," by the same authors, which appeared in the Proceedings of the First International Workshop on Expert Database Systems, Kiawah Island, South Carolina, October 1984. Much of this paper appeared as a chapter in *Readings in Knowledge Representation* (Morgan Kaufmann Publishers Inc., 1985), edited by the authors.

[2] Fellow of the Canadian Institute for Advanced Research.

The Received View in AI

GST
1936

1987

78

Expressiveness and tractability in knowledge representation and reasoning[1]

HECTOR J. LEVESQUE[2]
Department of Computer Science, University of Toronto, Toronto, Ont., Canada M5S 1A4

AND

RONALD J. BRACHMAN
AT&T Bell Laboratories, 600 Mountain Avenue, 3C-439, Murray Hill, NJ 07974, U.S.A.

Received November 3, 1986
Revision accepted April 8, 1987

A fundamental computational limit on automated reasoning and its effect on knowledge representation is examined. Basically, the problem is that it can be more difficult to reason correctly with one representational language than with another and, moreover, that this difficulty increases dramatically as the expressive power of the language increases. This leads to a tradeoff between the expressiveness of a representational language and its computational tractability. Here we show that this tradeoff can be seen to underlie the differences among a number of existing representational formalisms, in addition to motivating many of the current research issues in knowledge representation.

Key words: knowledge representation, description subsumption, complexity of reasoning, first-order logic, frames, semantic networks, databases.

Cet article étudie une limitation computationnelle fondamentale du raisonnement automatique et examine ses effets sur la représentation de connaissances. A la base le problème tient en ce qu'il peut être plus difficile de raisonner avec un langage de représentation qu'avec un autre et que cette difficulté augmente considérablement à mesure que croît le pouvoir expressif du langage. Ceci donne lieu à un compromis entre le pouvoir expressif d'un langage de représentation et sa tractibilité computationnelle. Nous montrons que ce compromis peut être vu comme l'une des causes fondamentales de la différence qui existe entre nombre de formalismes de représentation existants et peut motiver plusieurs recherches courantes en représentation de connaissances.

Mots clés : représentation de connaissances, complexité du raisonnement, logique du premier ordre, schémas, réseaux sémantiques, bases de données.

[Traduit par la revue]

Comput. Intell. 3, 78–93 (1987)

1. Introduction

This paper examines from a general point of view a basic computational limit on automated reasoning, and the effect that it has on knowledge representation (KR). The problem is essentially that it can be more difficult to reason correctly with one representational language than with another and, moreover, that this difficulty increases as the expressive power of the language increases. There is a tradeoff between the expressiveness of a representational language and its computational tractability. What we attempt to show is that this tradeoff underlies differences among a number of representational formalisms (such as first-order logic, databases, semantic networks, and frames) and motivates many current research issues in KR (such as the role of analogues, syntactic encodings, and defaults, as well as systems of limited inference and hybrid reasoning).

To deal with such a broad range of representational phenomena we must, of necessity, take a considerably simplified and incomplete view of KR. In particular, we focus on its computational and logical aspects, more or less ignoring its history and relevance in the areas of psychology, linguistics, and philosophy. The area of KR is still very disconnected today and the role of logic remains quite controversial, despite what this paper may suggest. We do believe, however, that the tradeoff discussed here is fundamental. As long as we are dealing with computational systems that reason automatically (without any special intervention or advice) and correctly (once we define what *that* means), we will be able to locate where they stand on the tradeoff: They will either be limited in what knowledge they can represent or unlimited in the reasoning effort they might require.

Our computational focus will not lead us to investigate specific algorithms and data structures for KR and reasoning, however. What we discuss is something much stronger, namely, whether or not algorithms of a certain kind can exist at all. The analysis here is at the *knowledge level* (Newell 1981) where we look at the content of what is represented (in terms of what it says about the world) and not the symbolic structures used to represent that knowledge. Indeed, we examine specific representation schemes in terms of what knowledge they can represent, rather than in terms of how they might actually represent it.

In the next section, we discuss what a KR system is for and what it could mean to reason correctly. Next, we investigate how a KR service might be realized using theorem proving in

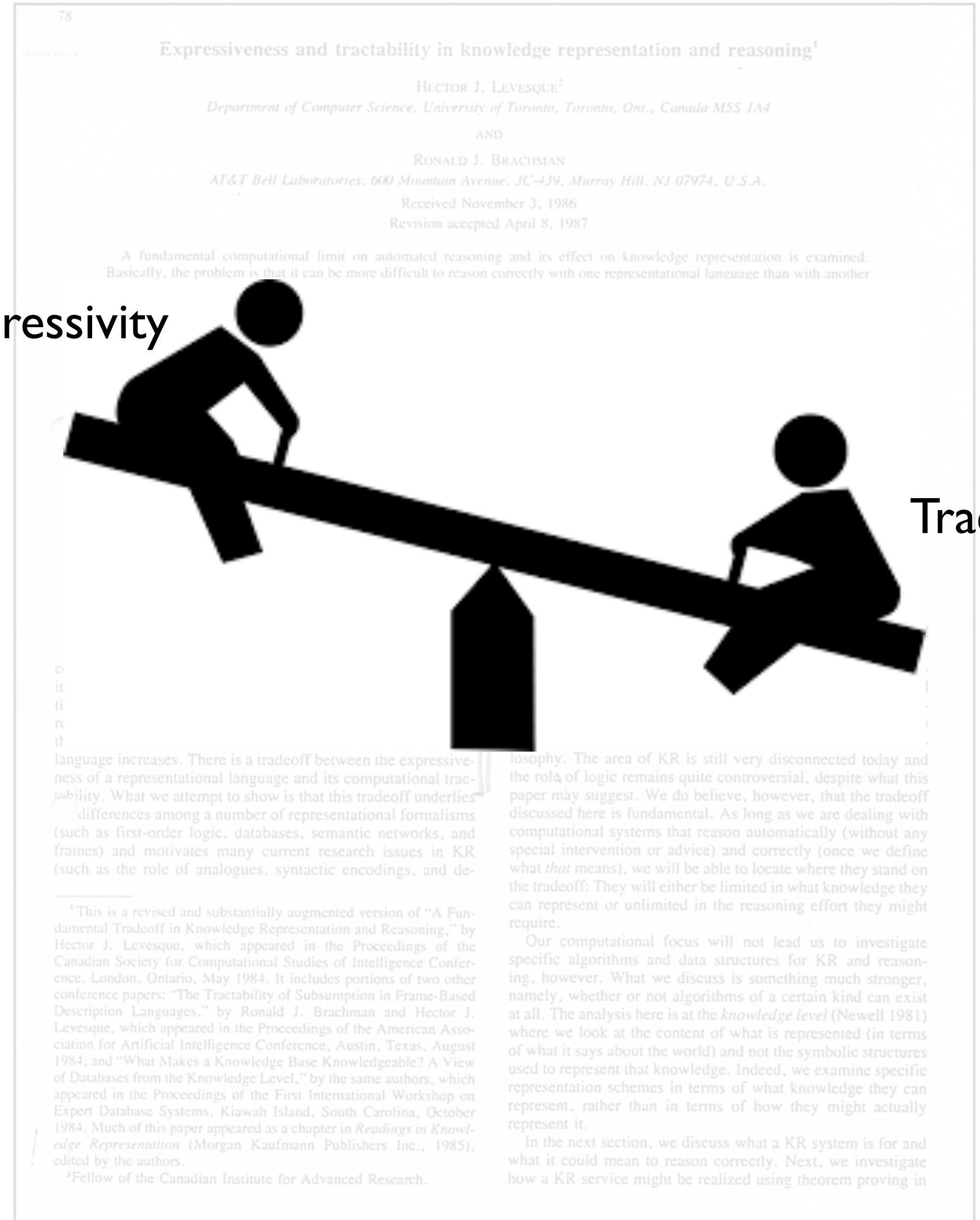
[1] This is a revised and substantially augmented version of "A Fundamental Tradeoff in Knowledge Representation and Reasoning," by Hector J. Levesque, which appeared in the Proceedings of the Canadian Society for Computational Studies of Intelligence Conference, London, Ontario, May 1984. It includes portions of two other conference papers: "The Tractability of Subsumption in Frame-Based Description Languages," by Ronald J. Brachman and Hector J. Levesque, which appeared in the Proceedings of the American Association for Artificial Intelligence Conference, Austin, Texas, August 1984; and "What Makes a Knowledge Base Knowledgeable? A View of Databases from the Knowledge Level," by the same authors, which appeared in the Proceedings of the First International Workshop on Expert Database Systems, Kiawah Island, South Carolina, October 1984. Much of this paper appeared as a chapter in *Readings in Knowledge Representation* (Morgan Kaufmann Publishers Inc., 1985), edited by the authors.

[2] Fellow of the Canadian Institute for Advanced Research.

The Received View in AI

GST
1936

1987



78

Expressiveness and tractability in knowledge representation and reasoning[1]

HECTOR J. LEVESQUE[2]
Department of Computer Science, University of Toronto, Toronto, Ont., Canada M5S 1A4

AND

RONALD J. BRACHMAN
AT&T Bell Laboratories, 600 Mountain Avenue, 3C-439, Murray Hill, NJ 07974, U.S.A.

Received November 3, 1986
Revision accepted April 8, 1987

A fundamental computational limit on automated reasoning and its effect on knowledge representation is examined. Basically, the problem is that it can be more difficult to reason correctly with one representational language than with another

language increases. There is a tradeoff between the expressiveness of a representational language and its computational tractability. What we attempt to show is that this tradeoff underlies differences among a number of representational formalisms (such as first-order logic, databases, semantic networks, and frames) and motivates many current research issues in KR (such as the role of analogues, syntactic encodings, and de-

losophy. The area of KR is still very disconnected today and the role of logic remains quite controversial, despite what this paper may suggest. We do believe, however, that the tradeoff discussed here is fundamental. As long as we are dealing with computational systems that reason automatically (without any special intervention or advice) and correctly (once we define what *that* means), we will be able to locate where they stand on the tradeoff: They will either be limited in what knowledge they can represent or unlimited in the reasoning effort they might require.

Our computational focus will not lead us to investigate specific algorithms and data structures for KR and reasoning, however. What we discuss is something much stronger, namely, whether or not algorithms of a certain kind can exist at all. The analysis here is at the *knowledge level* (Newell 1981) where we look at the content of what is represented (in terms of what it says about the world) and not the symbolic structures used to represent that knowledge. Indeed, we examine specific representation schemes in terms of what knowledge they can represent, rather than in terms of how they might actually represent it.

In the next section, we discuss what a KR system is for and what it could mean to reason correctly. Next, we investigate how a KR service might be realized using theorem proving in

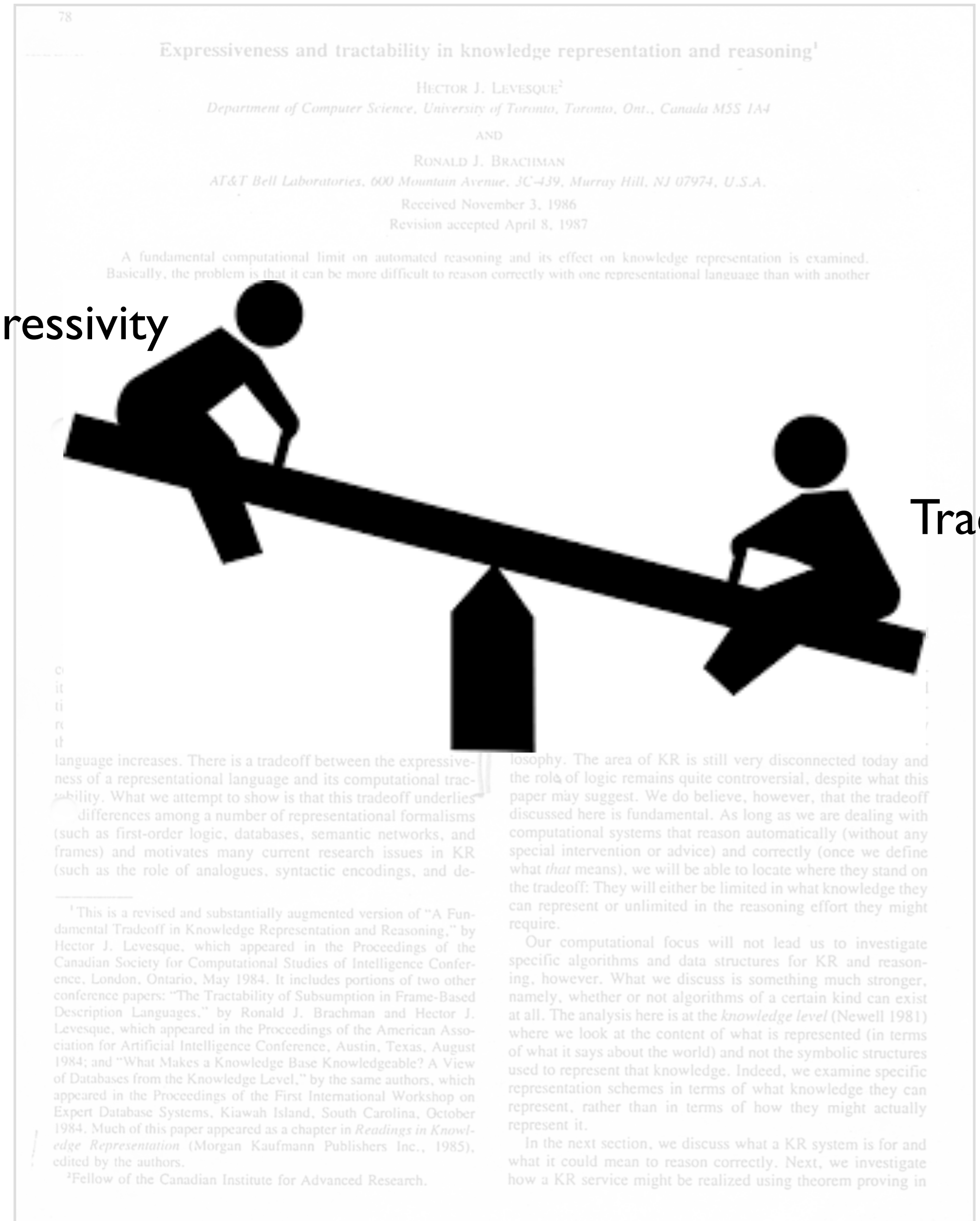
[1] This is a revised and substantially augmented version of "A Fundamental Tradeoff in Knowledge Representation and Reasoning," by Hector J. Levesque, which appeared in the Proceedings of the Canadian Society for Computational Studies of Intelligence Conference, London, Ontario, May 1984. It includes portions of two other conference papers: "The Tractability of Subsumption in Frame-Based Description Languages," by Ronald J. Brachman and Hector J. Levesque, which appeared in the Proceedings of the American Association for Artificial Intelligence Conference, Austin, Texas, August 1984; and "What Makes a Knowledge Base Knowledgeable? A View of Databases from the Knowledge Level," by the same authors, which appeared in the Proceedings of the First International Workshop on Expert Database Systems, Kiawah Island, South Carolina, October 1984. Much of this paper appeared as a chapter in *Readings in Knowledge Representation* (Morgan Kaufmann Publishers Inc., 1985), edited by the authors.

[2] Fellow of the Canadian Institute for Advanced Research.

The Received View in AI

1

GST
1936



78

Expressiveness and tractability in knowledge representation and reasoning[1]

HECTOR J. LEVESQUE[2]
Department of Computer Science, University of Toronto, Toronto, Ont., Canada M5S 1A4
AND
RONALD J. BRACHMAN
AT&T Bell Laboratories, 600 Mountain Avenue, 3C-439, Murray Hill, NJ 07974, U.S.A.

Received November 3, 1986
Revision accepted April 8, 1987

A fundamental computational limit on automated reasoning and its effect on knowledge representation is examined. Basically, the problem is that it can be more difficult to reason correctly with one representational language than with another

language increases. There is a tradeoff between the expressiveness of a representational language and its computational tractability. What we attempt to show is that this tradeoff underlies differences among a number of representational formalisms (such as first-order logic, databases, semantic networks, and frames) and motivates many current research issues in KR (such as the role of analogues, syntactic encodings, and de-

losophy. The area of KR is still very disconnected today and the role of logic remains quite controversial, despite what this paper may suggest. We do believe, however, that the tradeoff discussed here is fundamental. As long as we are dealing with computational systems that reason automatically (without any special intervention or advice) and correctly (once we define what *that* means), we will be able to locate where they stand on the tradeoff: They will either be limited in what knowledge they can represent or unlimited in the reasoning effort they might require.

Our computational focus will not lead us to investigate specific algorithms and data structures for KR and reasoning, however. What we discuss is something much stronger, namely, whether or not algorithms of a certain kind can exist at all. The analysis here is at the *knowledge level* (Newell 1981) where we look at the content of what is represented (in terms of what it says about the world) and not the symbolic structures used to represent that knowledge. Indeed, we examine specific representation schemes in terms of what knowledge they can represent, rather than in terms of how they might actually represent it.

In the next section, we discuss what a KR system is for and what it could mean to reason correctly. Next, we investigate how a KR service might be realized using theorem proving in

[1] This is a revised and substantially augmented version of "A Fundamental Tradeoff in Knowledge Representation and Reasoning," by Hector J. Levesque, which appeared in the Proceedings of the Canadian Society for Computational Studies of Intelligence Conference, London, Ontario, May 1984. It includes portions of two other conference papers: "The Tractability of Subsumption in Frame-Based Description Languages," by Ronald J. Brachman and Hector J. Levesque, which appeared in the Proceedings of the American Association for Artificial Intelligence Conference, Austin, Texas, August 1984; and "What Makes a Knowledge Base Knowledgeable? A View of Databases from the Knowledge Level," by the same authors, which appeared in the Proceedings of the First International Workshop on Expert Database Systems, Kiawah Island, South Carolina, October 1984. Much of this paper appeared as a chapter in *Readings in Knowledge Representation* (Morgan Kaufmann Publishers Inc., 1985), edited by the authors.

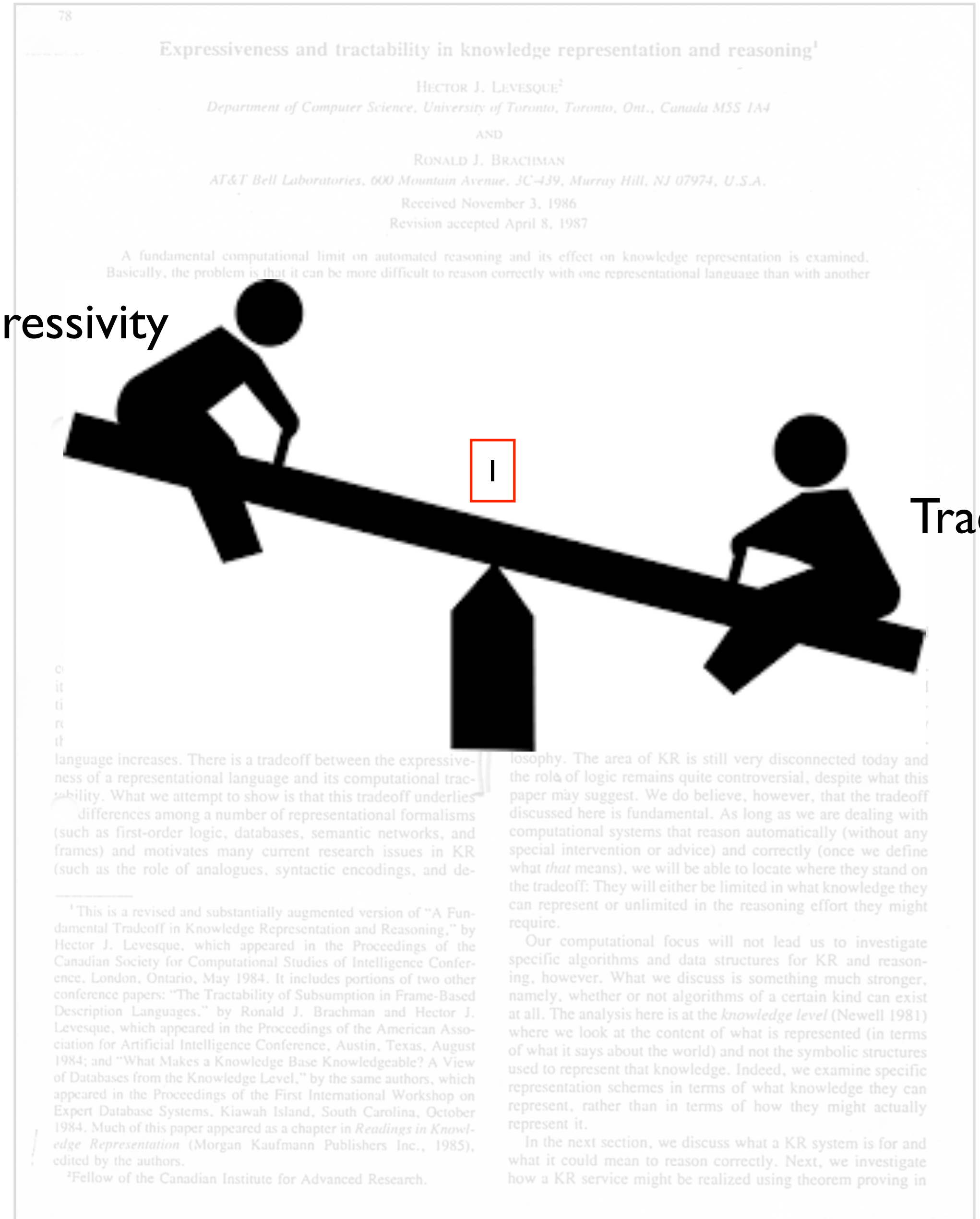
[2] Fellow of the Canadian Institute for Advanced Research.

1987

The Received View in AI

GST
1936

1987



Expressiveness and tractability in knowledge representation and reasoning[1]

Hector J. Levesque[2]
Department of Computer Science, University of Toronto, Toronto, Ont., Canada M5S 1A4
AND
Ronald J. Brachman
AT&T Bell Laboratories, 600 Mountain Avenue, 3C-439, Murray Hill, NJ 07974, U.S.A.

Received November 3, 1986
Revision accepted April 8, 1987

A fundamental computational limit on automated reasoning and its effect on knowledge representation is examined. Basically, the problem is that it can be more difficult to reason correctly with one representational language than with another

language increases. There is a tradeoff between the expressiveness of a representational language and its computational tractability. What we attempt to show is that this tradeoff underlies differences among a number of representational formalisms (such as first-order logic, databases, semantic networks, and frames) and motivates many current research issues in KR (such as the role of analogues, syntactic encodings, and de-

losophy. The area of KR is still very disconnected today and the role of logic remains quite controversial, despite what this paper may suggest. We do believe, however, that the tradeoff discussed here is fundamental. As long as we are dealing with computational systems that reason automatically (without any special intervention or advice) and correctly (once we define what *that* means), we will be able to locate where they stand on the tradeoff: They will either be limited in what knowledge they can represent or unlimited in the reasoning effort they might require.

Our computational focus will not lead us to investigate specific algorithms and data structures for KR and reasoning, however. What we discuss is something much stronger, namely, whether or not algorithms of a certain kind can exist at all. The analysis here is at the *knowledge level* (Newell 1981) where we look at the content of what is represented (in terms of what it says about the world) and not the symbolic structures used to represent that knowledge. Indeed, we examine specific representation schemes in terms of what knowledge they can represent, rather than in terms of how they might actually represent it.

In the next section, we discuss what a KR system is for and what it could mean to reason correctly. Next, we investigate how a KR service might be realized using theorem proving in

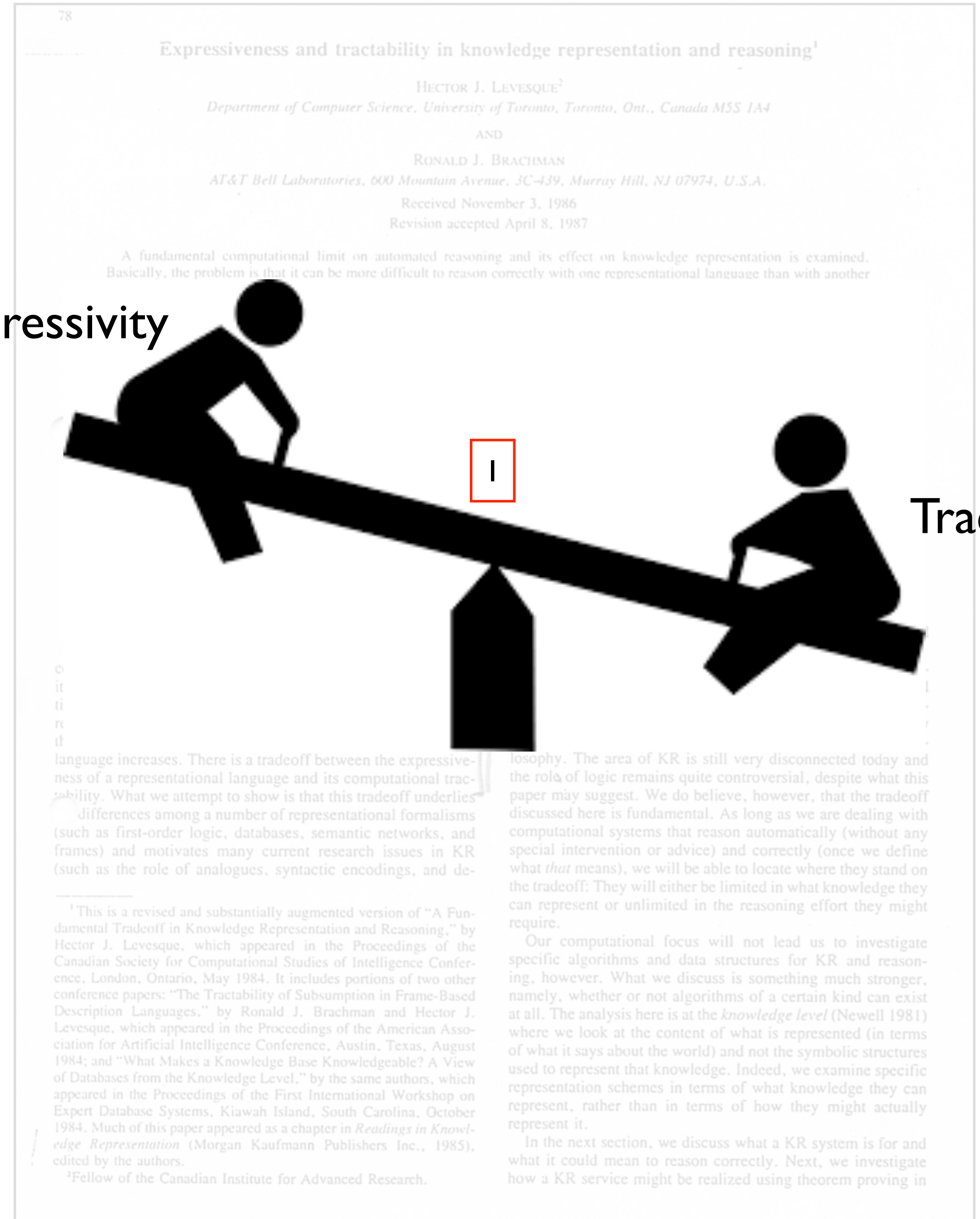
[1] This is a revised and substantially augmented version of "A Fundamental Tradeoff in Knowledge Representation and Reasoning," by Hector J. Levesque, which appeared in the Proceedings of the Canadian Society for Computational Studies of Intelligence Conference, London, Ontario, May 1984. It includes portions of two other conference papers: "The Tractability of Subsumption in Frame-Based Description Languages," by Ronald J. Brachman and Hector J. Levesque, which appeared in the Proceedings of the American Association for Artificial Intelligence Conference, Austin, Texas, August 1984; and "What Makes a Knowledge Base Knowledgeable? A View of Databases from the Knowledge Level," by the same authors, which appeared in the Proceedings of the First International Workshop on Expert Database Systems, Kiawah Island, South Carolina, October 1984. Much of this paper appeared as a chapter in *Readings in Knowledge Representation* (Morgan Kaufmann Publishers Inc., 1985), edited by the authors.

[2] Fellow of the Canadian Institute for Advanced Research.

The Received View in AI

GST
1936

1987



78

Expressiveness and tractability in knowledge representation and reasoning[1]

Hector J. Levesque[2]
Department of Computer Science, University of Toronto, Toronto, Ont., Canada M5S 1A4
AND
Ronald J. Brachman
AT&T Bell Laboratories, 600 Mountain Avenue, 3C-439, Murray Hill, NJ 07974, U.S.A.

Received November 3, 1986
Revision accepted April 8, 1987

A fundamental computational limit on automated reasoning and its effect on knowledge representation is examined. Basically, the problem is that it can be more difficult to reason correctly with one representational language than with another

language increases. There is a tradeoff between the expressiveness of a representational language and its computational tractability. What we attempt to show is that this tradeoff underlies differences among a number of representational formalisms (such as first-order logic, databases, semantic networks, and frames) and motivates many current research issues in KR (such as the role of analogues, syntactic encodings, and de-

losophy. The area of KR is still very disconnected today and the role of logic remains quite controversial, despite what this paper may suggest. We do believe, however, that the tradeoff discussed here is fundamental. As long as we are dealing with computational systems that reason automatically (without any special intervention or advice) and correctly (once we define what *that* means), we will be able to locate where they stand on the tradeoff: They will either be limited in what knowledge they can represent or unlimited in the reasoning effort they might require.

Our computational focus will not lead us to investigate specific algorithms and data structures for KR and reasoning, however. What we discuss is something much stronger, namely, whether or not algorithms of a certain kind can exist at all. The analysis here is at the *knowledge level* (Newell 1981) where we look at the content of what is represented (in terms of what it says about the world) and not the symbolic structures used to represent that knowledge. Indeed, we examine specific representation schemes in terms of what knowledge they can represent, rather than in terms of how they might actually represent it.

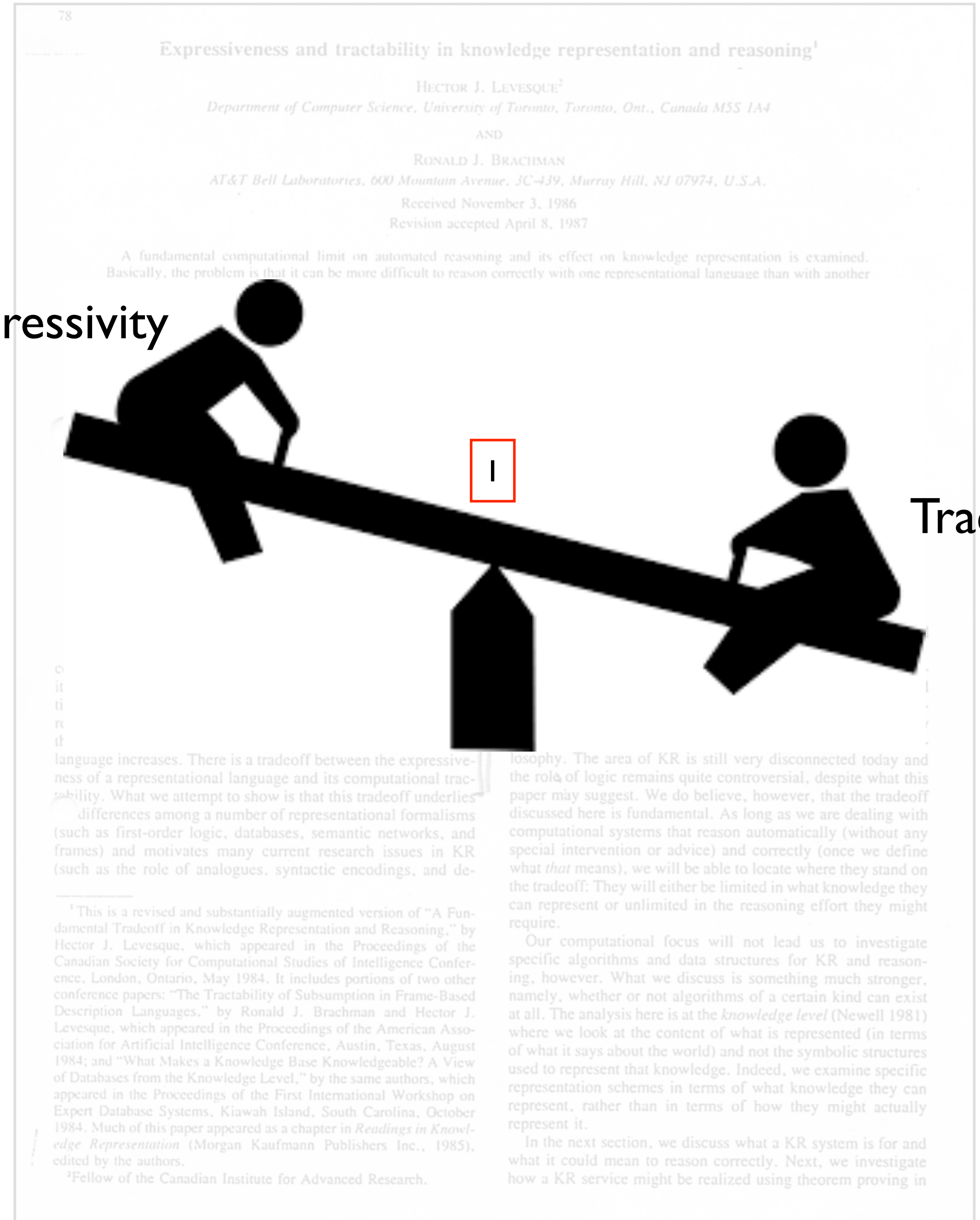
In the next section, we discuss what a KR system is for and what it could mean to reason correctly. Next, we investigate how a KR service might be realized using theorem proving in

[1]This is a revised and substantially augmented version of "A Fundamental Tradeoff in Knowledge Representation and Reasoning," by Hector J. Levesque, which appeared in the Proceedings of the Canadian Society for Computational Studies of Intelligence Conference, London, Ontario, May 1984. It includes portions of two other conference papers: "The Tractability of Subsumption in Frame-Based Description Languages," by Ronald J. Brachman and Hector J. Levesque, which appeared in the Proceedings of the American Association for Artificial Intelligence Conference, Austin, Texas, August 1984; and "What Makes a Knowledge Base Knowledgeable? A View of Databases from the Knowledge Level," by the same authors, which appeared in the Proceedings of the First International Workshop on Expert Database Systems, Kiawah Island, South Carolina, October 1984. Much of this paper appeared as a chapter in *Readings in Knowledge Representation* (Morgan Kaufmann Publishers Inc., 1985), edited by the authors.

[2]Fellow of the Canadian Institute for Advanced Research.

The Received View in AI

GST
1936



78

Expressiveness and tractability in knowledge representation and reasoning[1]

Hector J. Levesque[2]
Department of Computer Science, University of Toronto, Toronto, Ont., Canada M5S 1A4
and
Ronald J. Brachman
AT&T Bell Laboratories, 600 Mountain Avenue, 3C-439, Murray Hill, NJ 07974, U.S.A.

Received November 3, 1986
Revision accepted April 8, 1987

A fundamental computational limit on automated reasoning and its effect on knowledge representation is examined. Basically, the problem is that it can be more difficult to reason correctly with one representational language than with another

language increases. There is a tradeoff between the expressiveness of a representational language and its computational tractability. What we attempt to show is that this tradeoff underlies differences among a number of representational formalisms (such as first-order logic, databases, semantic networks, and frames) and motivates many current research issues in KR (such as the role of analogues, syntactic encodings, and de-

losophy. The area of KR is still very disconnected today and the role of logic remains quite controversial, despite what this paper may suggest. We do believe, however, that the tradeoff discussed here is fundamental. As long as we are dealing with computational systems that reason automatically (without any special intervention or advice) and correctly (once we define what *that* means), we will be able to locate where they stand on the tradeoff: They will either be limited in what knowledge they can represent or unlimited in the reasoning effort they might require.

Our computational focus will not lead us to investigate specific algorithms and data structures for KR and reasoning, however. What we discuss is something much stronger, namely, whether or not algorithms of a certain kind can exist at all. The analysis here is at the *knowledge level* (Newell 1981) where we look at the content of what is represented (in terms of what it says about the world) and not the symbolic structures used to represent that knowledge. Indeed, we examine specific representation schemes in terms of what knowledge they can represent, rather than in terms of how they might actually represent it.

In the next section, we discuss what a KR system is for and what it could mean to reason correctly. Next, we investigate how a KR service might be realized using theorem proving in

[1] This is a revised and substantially augmented version of "A Fundamental Tradeoff in Knowledge Representation and Reasoning," by Hector J. Levesque, which appeared in the Proceedings of the Canadian Society for Computational Studies of Intelligence Conference, London, Ontario, May 1984. It includes portions of two other conference papers: "The Tractability of Subsumption in Frame-Based Description Languages," by Ronald J. Brachman and Hector J. Levesque, which appeared in the Proceedings of the American Association for Artificial Intelligence Conference, Austin, Texas, August 1984; and "What Makes a Knowledge Base Knowledgeable? A View of Databases from the Knowledge Level," by the same authors, which appeared in the Proceedings of the First International Workshop on Expert Database Systems, Kiawah Island, South Carolina, October 1984. Much of this paper appeared as a chapter in *Readings in Knowledge Representation* (Morgan Kaufmann Publishers Inc., 1985), edited by the authors.

[2] Fellow of the Canadian Institute for Advanced Research.

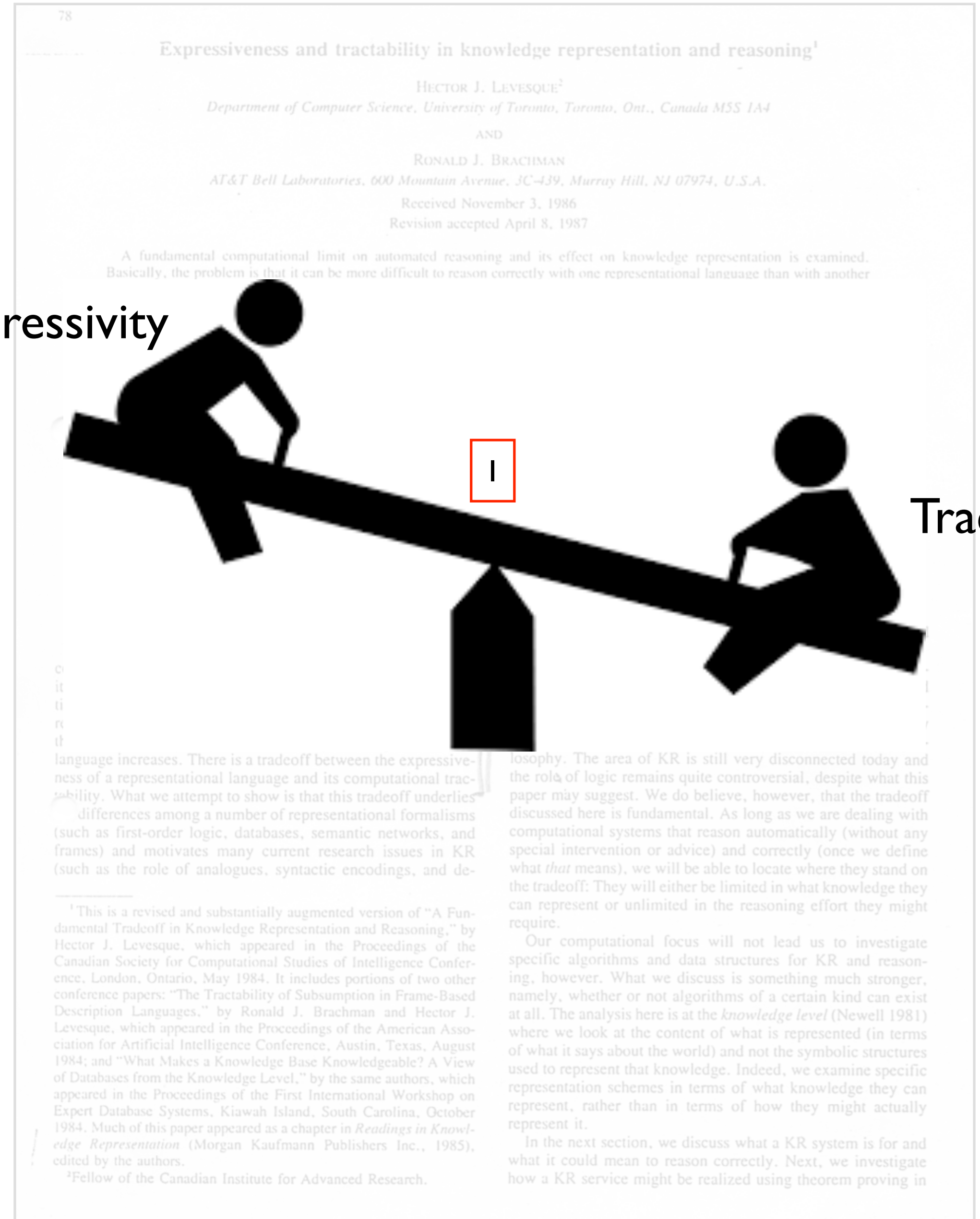
1987

2

The Received View in AI

GST
1936

1987



Expressiveness and tractability in knowledge representation and reasoning[1]

HECTOR J. LEVESQUE[2]
Department of Computer Science, University of Toronto, Toronto, Ont., Canada M5S 1A4
AND
RONALD J. BRACHMAN
AT&T Bell Laboratories, 600 Mountain Avenue, 3C-439, Murray Hill, NJ 07974, U.S.A.

Received November 3, 1986
Revision accepted April 8, 1987

A fundamental computational limit on automated reasoning and its effect on knowledge representation is examined. Basically, the problem is that it can be more difficult to reason correctly with one representational language than with another

language increases. There is a tradeoff between the expressiveness of a representational language and its computational tractability. What we attempt to show is that this tradeoff underlies differences among a number of representational formalisms (such as first-order logic, databases, semantic networks, and frames) and motivates many current research issues in KR (such as the role of analogues, syntactic encodings, and de-

[1] This is a revised and substantially augmented version of "A Fundamental Tradeoff in Knowledge Representation and Reasoning," by Hector J. Levesque, which appeared in the Proceedings of the Canadian Society for Computational Studies of Intelligence Conference, London, Ontario, May 1984. It includes portions of two other conference papers: "The Tractability of Subsumption in Frame-Based Description Languages," by Ronald J. Brachman and Hector J. Levesque, which appeared in the Proceedings of the American Association for Artificial Intelligence Conference, Austin, Texas, August 1984; and "What Makes a Knowledge Base Knowledgeable? A View of Databases from the Knowledge Level," by the same authors, which appeared in the Proceedings of the First International Workshop on Expert Database Systems, Kiawah Island, South Carolina, October 1984. Much of this paper appeared as a chapter in *Readings in Knowledge Representation* (Morgan Kaufmann Publishers Inc., 1985), edited by the authors.

[2] Fellow of the Canadian Institute for Advanced Research.

losophy. The area of KR is still very disconnected today and the role of logic remains quite controversial, despite what this paper may suggest. We do believe, however, that the tradeoff discussed here is fundamental. As long as we are dealing with computational systems that reason automatically (without any special intervention or advice) and correctly (once we define what *that* means), we will be able to locate where they stand on the tradeoff: They will either be limited in what knowledge they can represent or unlimited in the reasoning effort they might require.

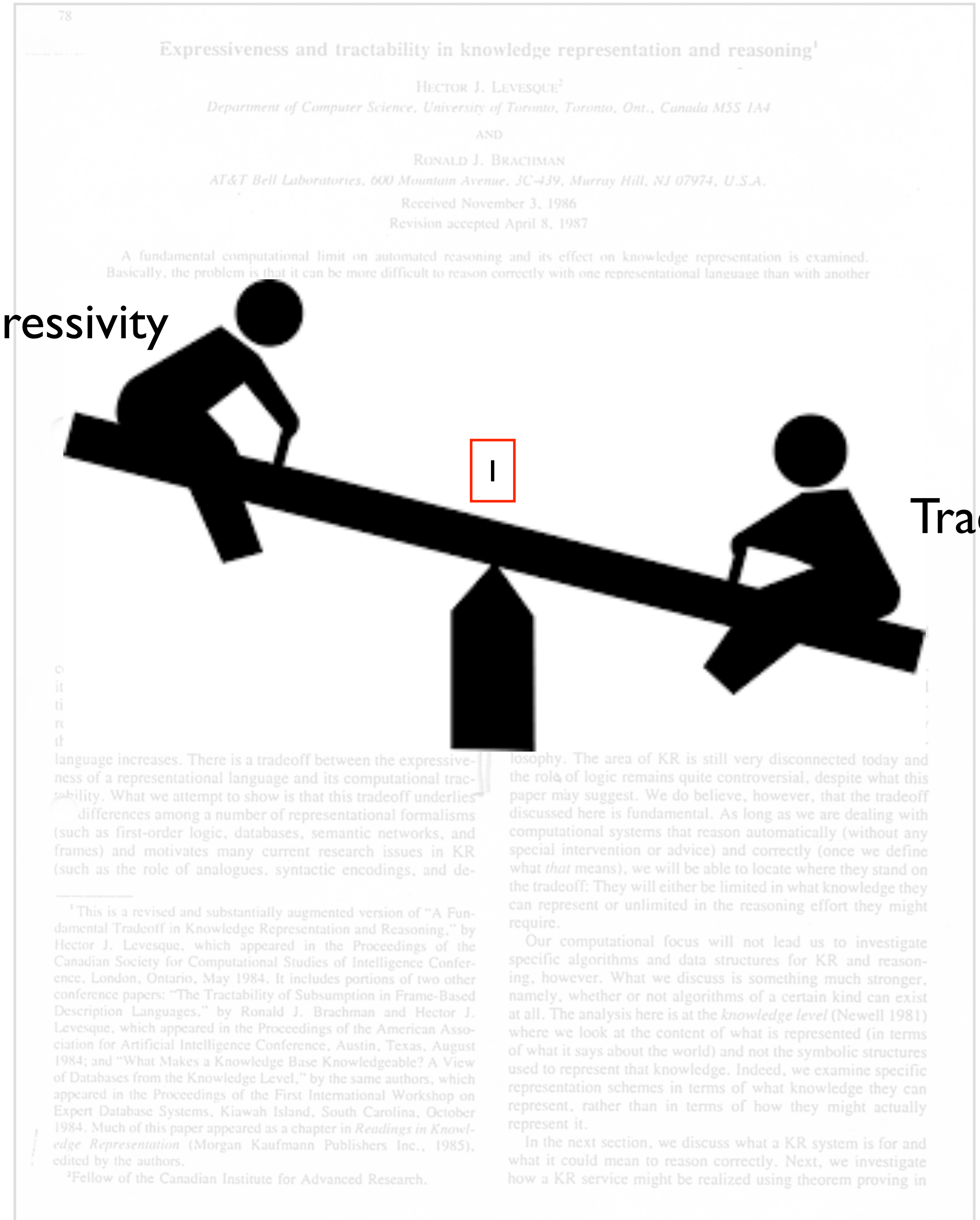
Our computational focus will not lead us to investigate specific algorithms and data structures for KR and reasoning, however. What we discuss is something much stronger, namely, whether or not algorithms of a certain kind can exist at all. The analysis here is at the *knowledge level* (Newell 1981) where we look at the content of what is represented (in terms of what it says about the world) and not the symbolic structures used to represent that knowledge. Indeed, we examine specific representation schemes in terms of what knowledge they can represent, rather than in terms of how they might actually represent it.

In the next section, we discuss what a KR system is for and what it could mean to reason correctly. Next, we investigate how a KR service might be realized using theorem proving in

The Received View in AI

GST
1936

1987



Expressiveness and tractability in knowledge representation and reasoning[1]

HECTOR J. LEVESQUE[2]
Department of Computer Science, University of Toronto, Toronto, Ont., Canada M5S 1A4
AND
RONALD J. BRACHMAN
AT&T Bell Laboratories, 600 Mountain Avenue, 3C-439, Murray Hill, NJ 07974, U.S.A.

Received November 3, 1986
Revision accepted April 8, 1987

A fundamental computational limit on automated reasoning and its effect on knowledge representation is examined. Basically, the problem is that it can be more difficult to reason correctly with one representational language than with another

language increases. There is a tradeoff between the expressiveness of a representational language and its computational tractability. What we attempt to show is that this tradeoff underlies differences among a number of representational formalisms (such as first-order logic, databases, semantic networks, and frames) and motivates many current research issues in KR (such as the role of analogues, syntactic encodings, and de-

losophy. The area of KR is still very disconnected today and the role of logic remains quite controversial, despite what this paper may suggest. We do believe, however, that the tradeoff discussed here is fundamental. As long as we are dealing with computational systems that reason automatically (without any special intervention or advice) and correctly (once we define what *that* means), we will be able to locate where they stand on the tradeoff: They will either be limited in what knowledge they can represent or unlimited in the reasoning effort they might require.

Our computational focus will not lead us to investigate specific algorithms and data structures for KR and reasoning, however. What we discuss is something much stronger, namely, whether or not algorithms of a certain kind can exist at all. The analysis here is at the *knowledge level* (Newell 1981) where we look at the content of what is represented (in terms of what it says about the world) and not the symbolic structures used to represent that knowledge. Indeed, we examine specific representation schemes in terms of what knowledge they can represent, rather than in terms of how they might actually represent it.

In the next section, we discuss what a KR system is for and what it could mean to reason correctly. Next, we investigate how a KR service might be realized using theorem proving in

[1] This is a revised and substantially augmented version of "A Fundamental Tradeoff in Knowledge Representation and Reasoning," by Hector J. Levesque, which appeared in the Proceedings of the Canadian Society for Computational Studies of Intelligence Conference, London, Ontario, May 1984. It includes portions of two other conference papers: "The Tractability of Subsumption in Frame-Based Description Languages," by Ronald J. Brachman and Hector J. Levesque, which appeared in the Proceedings of the American Association for Artificial Intelligence Conference, Austin, Texas, August 1984; and "What Makes a Knowledge Base Knowledgeable? A View of Databases from the Knowledge Level," by the same authors, which appeared in the Proceedings of the First International Workshop on Expert Database Systems, Kiawah Island, South Carolina, October 1984. Much of this paper appeared as a chapter in *Readings in Knowledge Representation* (Morgan Kaufmann Publishers Inc., 1985), edited by the authors.

[2] Fellow of the Canadian Institute for Advanced Research.

Expressivity

1

Tractability

?

2

Bringsjord: Speedup shoots this down, & hence to ignore automated reasoning in highly expressive formats would be foolish for AI.

A Simpler "Speedup" Theorem

A Simpler "Speedup" Theorem

Let f be any recursive function, and again let us refer to $\Phi \supset \mathbf{PA}$. Then there are arithmetic $\mathscr{L}_1$ sentences ϕ s.t. $\Phi \vdash \phi$, where the shortest proof P confirming this has more more than $f(n^\phi)$ symbols.

To prove GST, we shall once again allow ourselves …

Again: The Fixed Point Theorem (FPT)

Assume that Φ is a set of arithmetic sentences such that Repr Φ. There for every arithmetic formula $\psi(x)$ with one free variable x, there is an arithmetic sentence ϕ s.t.

$\Phi \vdash \phi \leftrightarrow \psi(n^{\phi})$.

We can intuitively understand ϕ to be saying:
"I have the property ascribed to me by the formula ψ."

Ok; so let's do it …

Proof: Let f^* be an arbitrary (total) recursive function. We can clearly define a meta-logical relation that expresses the property of having a proof in Φ of ϕ shorter, symbol-wise, than $f(n^\phi)$, for the Gödel number of any formula ϕ. Let us abbreviate this relation as: Prov-sh$_\Phi(\phi, n^\phi)$. By Repr Φ, since a Turing machine can compute this relation, we then have:

(Rep*) = (1) Prov-sh$_\Phi(n^\phi)$ iff $\Phi \vdash \phi$.

Next, we can instantiate the Fixed Point Theorem to yield a formula that declares "There's no proof of me shorter than what f^* applied to me returns!" (And note that we employ a logicization of our meta-logical relation.). More formally, the instantiation will be:

(FPT*) = (2) $\Phi \vdash \bar{\pi}_{sh} \leftrightarrow \neg\mathscr{P}$-sh$_\Phi(n^{\bar{\pi}_{sh}})$

Now what about this self-referential sentence? Can it have a proof shorter than f^* applied to its Gödel number? Suppose for contradiction that it does. Then by left-to-right on (1) it's provable in Φ. But given this, combined with (2), this self-referential sentence is *not* provable by a derivation shorter than f^* applied to it — contradiction! **QED**

Med nok penger, kan logikk løse alle problemer.