Forcing is simply the iterative conception undertaken with multivalued logic

Joel David Hamkins
O'Hara Professor of Logic
University of Notre Dame
VRF, Mathematical Institute, Oxford

ForcingFest
Oslo 21 June 2024

Introduction—myths of forcing

People sometimes describe forcing as involving:

- Forcing is about augmenting a given model of set theory with an "ideal" object, like a field extension

Introduction—myths of forcing

People sometimes describe forcing as involving:

- Forcing is about augmenting a given model of set theory with an "ideal" object, like a field extension
- Forcing is about genericity, a filter meeting dense sets

Introduction—myths of forcing

People sometimes describe forcing as involving:

- Forcing is about augmenting a given model of set theory with an “ideal” object, like a field extension
- Forcing is about genericity, a filter meeting dense sets
- The forcing extension is defined via $\mathrm{val}(\tau, G)$, the value of a name τ by a generic filter G.

Introduction—myths of forcing

People sometimes describe forcing as involving:

- Forcing is about augmenting a given model of set theory with an “ideal” object, like a field extension
- Forcing is about genericity, a filter meeting dense sets
- The forcing extension is defined via $\mathrm{val}(\tau, G)$, the value of a name τ by a generic filter G.
- Forcing is about countable transitive models of ZFC or a fragment of ZFC.

Introduction—myths of forcing

People sometimes describe forcing as involving:

- Forcing is about augmenting a given model of set theory with an “ideal” object, like a field extension
- Forcing is about genericity, a filter meeting dense sets
- The forcing extension is defined via $\mathrm{val}(\tau, G)$, the value of a name τ by a generic filter G.
- Forcing is about countable transitive models of ZFC or a fragment of ZFC.

Nevertheless, I shall give an account of forcing that is about none of these things.

Introduction—myths of forcing

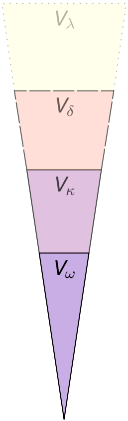
People sometimes describe forcing as involving:

- Forcing is about augmenting a given model of set theory with an "ideal" object, like a field extension
- Forcing is about genericity, a filter meeting dense sets
- The forcing extension is defined via $\mathrm{val}(\tau, G)$, the value of a name τ by a generic filter G.
- Forcing is about countable transitive models of ZFC or a fragment of ZFC.

Nevertheless, I shall give an account of forcing that is about none of these things.

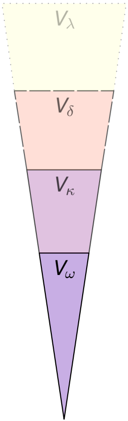
Rather, in this alternative account, forcing arises naturally from the iterative conception, but undertaken in a multi-valued logical setting.

Iterative conception and the cumulative hierarchy



The set-theoretic universe is often described as an iterative cumulative hierarchy, stratified by levels.

Iterative conception and the cumulative hierarchy

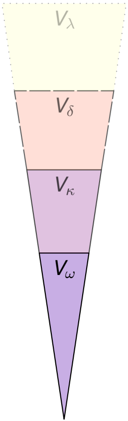


The set-theoretic universe is often described as an iterative cumulative hierarchy, stratified by levels.

We begin at the bottom layer with nothing, the empty set.

$$V_0 = \varnothing.$$

Iterative conception and the cumulative hierarchy



The set-theoretic universe is often described as an iterative cumulative hierarchy, stratified by levels.

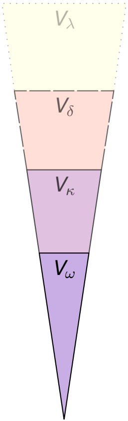
We begin at the bottom layer with nothing, the empty set.

$$V_0 = \varnothing.$$

Subsequent layers consist of all subsets of earlier levels.

$$V_{\alpha+1} = P(V_\alpha) = \{\, X \mid X \subseteq V_\alpha \,\}$$

Iterative conception and the cumulative hierarchy



The set-theoretic universe is often described as an iterative cumulative hierarchy, stratified by levels.

We begin at the bottom layer with nothing, the empty set.

$$V_0 = \varnothing.$$

Subsequent layers consist of all subsets of earlier levels.

$$V_{\alpha+1} = P(V_\alpha) = \{\, X \mid X \subseteq V_\alpha \,\}$$

At limit stages, we accumulate everything so far:

$$V_\lambda = \bigcup_{\alpha<\alpha} V_\alpha, \qquad \text{for limit ordinals } \lambda.$$

Thus the entire set-theoretic universe V emerges as the *cumulative hierarchy*.

Set theory via hereditary functions

Can we develop the central ideas of set theory using functions instead of sets?

Set theory via hereditary functions

Can we develop the central ideas of set theory using functions instead of sets?

Instead of having a set A with elements $a \in A$, imagine that we have a function

$$f : X \to 2 = \{ 0, 1 \},$$

We compute the value $f(a)$, with $f(a) = 1$ indicating that a is a member.

Set theory via hereditary functions

Can we develop the central ideas of set theory using functions instead of sets?

Instead of having a set A with elements $a \in A$, imagine that we have a function

$$f : X \to 2 = \{0, 1\},$$

We compute the value $f(a)$, with $f(a) = 1$ indicating that a is a member.

But of course, a itself should be a function, not a set.

Set theory via hereditary functions

Can we develop the central ideas of set theory using functions instead of sets?

Instead of having a set A with elements $a \in A$, imagine that we have a function

$$f : X \to 2 = \{ 0, 1 \},$$

We compute the value $f(a)$, with $f(a) = 1$ indicating that a is a member.

But of course, a itself should be a function, not a set.

Indeed, hereditarily.

Set theory via hereditary functions

Can we develop the central ideas of set theory using functions instead of sets?

Instead of having a set A with elements $a \in A$, imagine that we have a function

$$f : X \to \mathbb{2} = \{0, 1\},$$

We compute the value $f(a)$, with $f(a) = 1$ indicating that a is a member.

But of course, a itself should be a function, not a set.

Indeed, hereditarily. Functions all the way down.

Hereditary functions hierarchy

We will use characteristic functions. $2 = \{0, 1\}$.

- $f(g) = 1$ indicates that g is an element of f
- $f(g) = 0$ lacks that indication

Hereditary functions hierarchy

We will use characteristic functions. $\mathbb{2} = \{ 0, 1 \}$.

- $f(g) = 1$ indicates that g is an element of f
- $f(g) = 0$ lacks that indication

Build up the hierarchy in stages, beginning with nothing.

$$V_0^{\mathbb{2}} = \varnothing.$$

Hereditary functions hierarchy

We will use characteristic functions. $\mathbb{2} = \{0, 1\}$.

- $f(g) = 1$ indicates that g is an element of f
- $f(g) = 0$ lacks that indication

Build up the hierarchy in stages, beginning with nothing.

$$V_0^{\mathbb{2}} = \varnothing.$$

Subsequent layers have all functions defined on earlier levels .

$$V_{\alpha+1}^{\mathbb{2}} \quad = \quad \{ f \mid f : V_\alpha^{\mathbb{2}} \to \mathbb{2} \}$$

Hereditary functions hierarchy

We will use characteristic functions. $\mathbb{2} = \{0, 1\}$.

- $f(g) = 1$ indicates that g is an element of f
- $f(g) = 0$ lacks that indication

Build up the hierarchy in stages, beginning with nothing.

$$V_0^{\mathbb{2}} = \varnothing.$$

Subsequent layers have all functions defined on earlier levels .

$$V_{\alpha+1}^{\mathbb{2}} \quad = \quad \{ f \mid f \colon V_\alpha^{\mathbb{2}} \to \mathbb{2} \}$$

At limit stages, we accumulate everything so far:

$$V_\lambda^{\mathbb{2}} = \bigcup_{\alpha<\alpha} V_\alpha^{\mathbb{2}}, \qquad \text{for limit ordinals } \lambda.$$

Thus the hereditary function universe $V^{\mathbb{2}}$ emerges in a cumulative hierarchy. It's functions all the way down.

Hereditary functions as sets

We want to view the hereditary functions as sets.

Hereditary functions as sets

We want to view the hereditary functions as sets.

Naive idea

The idea seems to be to define $g \in f$ if and only if $f(g) = 1$.

Hereditary functions as sets

We want to view the hereditary functions as sets.

Naive idea

The idea seems to be to define $g \in f$ if and only if $f(g) = 1$.

But that doesn't quite work—there is a problem.

Hereditary functions as sets

We want to view the hereditary functions as sets.

Naive idea

The idea seems to be to define $g \in f$ if and only if $f(g) = 1$.

But that doesn't quite work—there is a problem.

Not well defined. It could be that g and g' agree on all the value 1 items, hence "equivalent," but we might have $f(g') = 0$, even though $f(g) = 1$.

Hereditary functions as sets

We want to view the hereditary functions as sets.

Naive idea

The idea seems to be to define $g \in f$ if and only if $f(g) = 1$.

But that doesn't quite work—there is a problem.

Not well defined. It could be that g and g' agree on all the value 1 items, hence "equivalent," but we might have $f(g') = 0$, even though $f(g) = 1$.

Worse: perhaps g and g' agree only on *equivalent* items.

Seems to get complicated. What to do?

Recursive definition of equivalence and membership

We can define equivalence and membership recursively.
Truth values $[\![\varphi]\!]$ are either 0 or 1.

Recursive definition of equivalence and membership

We can define equivalence and membership recursively.
Truth values $[\![\varphi]\!]$ are either 0 or 1.

For hereditary functions $f, g \in V^{2}$, we define:

$$[\![g \in f]\!] \quad = \quad \bigvee_{f(h)=1} [\![g = h]\!]$$

Recursive definition of equivalence and membership

We can define equivalence and membership recursively.
Truth values $[\![\varphi]\!]$ are either 0 or 1.

For hereditary functions $f, g \in V^{\mathbb{2}}$, we define:

$$[\![g \in f]\!] = \bigvee_{f(h)=1} [\![g = h]\!]$$

$$[\![g = f]\!] = [\![g \subseteq f]\!] \wedge [\![f \subseteq g]\!]$$

$$[\![g \subseteq f]\!] = \bigwedge_{g(h)=1} [\![h \in f]\!]$$

Recursive definition of equivalence and membership

We can define equivalence and membership recursively.
Truth values $[\![\varphi]\!]$ are either 0 or 1.

For hereditary functions $f, g \in V^{\mathbb{2}}$, we define:

$$[\![g \in f]\!] = \bigvee_{f(h)=1} [\![g = h]\!]$$

$$[\![g = f]\!] = [\![g \subseteq f]\!] \wedge [\![f \subseteq g]\!]$$

$$[\![g \subseteq f]\!] = \bigwedge_{g(h)=1} [\![h \in f]\!]$$

Defined by recursion on the hereditary function hierarchy.

The hereditary function quotient

Define the quotient relations

$$f =_1 g \quad \text{if and only if} \quad [\![f = g]\!] = 1$$

This is a congruence with respect to $\in_1$.

The hereditary function quotient

Define the quotient relations

$$f =_1 g \quad \text{if and only if} \quad [\![f = g]\!] = 1$$

This is a congruence with respect to $\in_1$.

$$g \in_1 f \quad \text{if and only if} \quad [\![g \in f]\!] = 1$$

Not the same as merely $f(g) = 1$.

The hereditary function quotient

Define the quotient relations

$$f =_1 g \quad \text{if and only if} \quad [\![f = g]\!] = 1$$

This is a congruence with respect to $\in_1$.

$$g \in_1 f \quad \text{if and only if} \quad [\![g \in f]\!] = 1$$

Not the same as merely $f(g) = 1$.

Gives a quotient structure on the equivalence classes $[f]_{=_1}$.

$$\langle V^{\mathbb{2}}, \in_1 \rangle / =_1$$

This is now a model of ZFC.

Inserting original universe into new universe

We build the function universe $V^{\mathbb{2}}$ inside ZFC universe $\langle V, \in \rangle$.

Inserting original universe into new universe

We build the function universe $V^{\mathbb{2}}$ inside ZFC universe $\langle V, \in \rangle$.

Define natural translation of sets x to functions $\check{x} : \mathrm{dom}(\check{x}) \to \mathbb{2}$.

Inserting original universe into new universe

We build the function universe $V^{\mathbb{2}}$ inside ZFC universe $\langle V, \in \rangle$.

Define natural translation of sets x to functions $\check{x} : \mathrm{dom}(\check{x}) \to \mathbb{2}$.

$$\mathrm{dom}(\check{x}) \ = \ \{ \, \check{y} \mid y \in x \, \}, \qquad\qquad \check{x}(\check{y}) = 1$$

Inserting original universe into new universe

We build the function universe $V^{\mathbb{2}}$ inside ZFC universe $\langle V, \in \rangle$.

Define natural translation of sets x to functions $\check{x} : \mathrm{dom}(\check{x}) \to \mathbb{2}$.

$$\mathrm{dom}(\check{x}) \;=\; \{\, \check{y} \mid y \in x \,\}, \qquad \check{x}(\check{y}) = 1$$

Map every set into the hereditary function universe

$$x \quad \mapsto \quad \check{x}$$

Inserting original universe into new universe

We build the function universe $V^{\mathbb{2}}$ inside ZFC universe $\langle V, \in \rangle$.

Define natural translation of sets x to functions $\check{x} : \mathrm{dom}(\check{x}) \to \mathbb{2}$.

$$\mathrm{dom}(\check{x}) \;=\; \{\, \check{y} \mid y \in x \,\}, \qquad\qquad \check{x}(\check{y}) = 1$$

Map every set into the hereditary function universe

$$x \quad \mapsto \quad \check{x}$$

It preserves the membership relation:

$$y \in x \quad \iff \quad \check{y} \in_1 \check{x}$$

Hereditary function universe isomorphic to V

Fact

For hereditary functions, the check map is an isomorphism

$$\langle V, \in\rangle \quad \cong \quad \langle V^{2}, \in_1\rangle / =_1$$

$$x \quad \mapsto \quad [\check{x}]_{=_1}$$

Hereditary function universe isomorphic to V

Fact

For hereditary functions, the check map is an isomorphism

$$\langle V, \in\rangle \quad \cong \quad \langle V^{2}, \in_1\rangle / =_1$$

$$x \quad \mapsto \quad [\check{x}]_{=_1}$$

Every function is equivalent to a check function $\check{x}$.

Hereditary function universe isomorphic to V

Fact

For hereditary functions, the check map is an isomorphism

$$\langle V, \in\rangle \quad \cong \quad \langle V^2, \in_1\rangle / =_1$$

$$x \quad \mapsto \quad [\check{x}]_{=_1}$$

Every function is equivalent to a check function $\check{x}$.

The fact shows the sense in which the hereditary function idea provides an equivalent perspective on set theory.

Multi-valued logical setting

Next, let's try same idea, but with more than two truth values.

Multi-valued logical setting

Next, let's try same idea, but with more than two truth values.

We allow truth values not just in $\mathbb{2} = \{ 0, 1 \}$, but in an arbitrary multi-valued logic $\mathbb{A}$. Works best when $\mathbb{A}$ is complete.

Multi-valued logical setting

Next, let's try same idea, but with more than two truth values.

We allow truth values not just in $2 = \{0, 1\}$, but in an arbitrary multi-valued logic $\mathbb{A}$. Works best when $\mathbb{A}$ is complete.

Want to use hereditary functions with truth values in $\mathbb{A}$.

$$\sigma : X \to \mathbb{A}$$

Functions all the way down.

Multi-valued logical setting

Next, let's try same idea, but with more than two truth values.

We allow truth values not just in $\mathbb{2} = \{0, 1\}$, but in an arbitrary multi-valued logic $\mathbb{A}$. Works best when $\mathbb{A}$ is complete.

Want to use hereditary functions with truth values in $\mathbb{A}$.

$$\sigma : X \to \mathbb{A}$$

Functions all the way down.

Main idea

$\sigma(\tau) = a$ means:

τ is a member of σ, with truth value at least a.

Universe of $\mathbb{A}$-names

The cumulative universe of $\mathbb{A}$-names starts with nothing.

$$V_0^{\mathbb{A}} = \varnothing.$$

Universe of $\mathbb{A}$-names

The cumulative universe of $\mathbb{A}$-names starts with nothing.

$$V_0^{\mathbb{A}} = \varnothing.$$

Subsequent layers have all functions on earlier levels.

Universe of $\mathbb{A}$-names

The cumulative universe of $\mathbb{A}$-names starts with nothing.

$$V_0^{\mathbb{A}} = \varnothing.$$

Subsequent layers have all functions on earlier levels.

$$V_{\alpha+1}^{\mathbb{A}} = \{ \sigma \mid \sigma \vdots V_\alpha^{\mathbb{A}} \to \mathbb{A} \}$$

Universe of $\mathbb{A}$-names

The cumulative universe of $\mathbb{A}$-names starts with nothing.

$$V_0^{\mathbb{A}} = \varnothing.$$

Subsequent layers have all functions on earlier levels.

$$V_{\alpha+1}^{\mathbb{A}} = \{\, \sigma \mid \sigma : V_\alpha^{\mathbb{A}} \to \mathbb{A} \,\}$$

At limit stages λ, we accumulate everything so far:

$$V_\lambda^{\mathbb{A}} = \bigcup_{\alpha<\alpha} V_\alpha^{\mathbb{A}}.$$

Key point—iterative conception in multi-valued logic

The resulting universe $V^{\mathbb{A}}$ is the iterative cumulative hierarchy, the class of all $\mathbb{A}$-names, in multi-valued logic.

Atomic truth values

We define the atomic truth values recursively, as before:

$$[\![\,\tau \in \sigma\,]\!] \quad = \quad \bigvee_{\sigma(\eta)=b} [\![\,\tau = \eta\,]\!] \wedge b$$

Atomic truth values

We define the atomic truth values recursively, as before:

$$[\![\tau \in \sigma]\!] = \bigvee_{\sigma(\eta)=b} [\![\tau = \eta]\!] \wedge b$$

$$[\![\tau = \sigma]\!] = [\![\tau \subseteq \sigma]\!] \wedge [\![\sigma \subseteq \tau]\!]$$

$$[\![\tau \subseteq \sigma]\!] = \bigwedge_{\tau(\eta)=b} (b \implies [\![\eta \in \sigma]\!])$$

Atomic truth values

We define the atomic truth values recursively, as before:

$$[\![\,\tau \in \sigma\,]\!] \quad = \quad \bigvee_{\sigma(\eta)=b} [\![\,\tau = \eta\,]\!] \wedge b$$

$$[\![\,\tau = \sigma\,]\!] \quad = \quad [\![\,\tau \subseteq \sigma\,]\!] \wedge [\![\,\sigma \subseteq \tau\,]\!]$$

$$[\![\,\tau \subseteq \sigma\,]\!] \quad = \quad \bigwedge_{\tau(\eta)=b} (b \implies [\![\,\eta \in \sigma\,]\!])$$

Defined by recursion on the hierarchy of names.

Atomic truth values

We define the atomic truth values recursively, as before:

$$[\![\tau \in \sigma]\!] = \bigvee_{\sigma(\eta)=b} [\![\tau = \eta]\!] \wedge b$$

$$[\![\tau = \sigma]\!] = [\![\tau \subseteq \sigma]\!] \wedge [\![\sigma \subseteq \tau]\!]$$

$$[\![\tau \subseteq \sigma]\!] = \bigwedge_{\tau(\eta)=b} (b \implies [\![\eta \in \sigma]\!])$$

Defined by recursion on the hierarchy of names.

The algebra $\mathbb{A}$ should be complete for the infinite conjunctions/disjunctions to make sense.

$\mathbb{A}$-valued semantics

The semantics extend naturally to all assertions.

$$[\![\, \varphi \wedge \psi \,]\!] = [\![\, \varphi \,]\!] \wedge [\![\, \psi \,]\!]$$

$\mathbb{A}$-valued semantics

The semantics extend naturally to all assertions.

$$[\![\,\varphi \wedge \psi\,]\!] = [\![\,\varphi\,]\!] \wedge [\![\,\psi\,]\!]$$

$$[\![\,\neg\varphi\,]\!] = \neg[\![\,\varphi\,]\!]$$

$\mathbb{A}$-valued semantics

The semantics extend naturally to all assertions.

$$[\![\, \varphi \wedge \psi \,]\!] = [\![\, \varphi \,]\!] \wedge [\![\, \psi \,]\!]$$

$$[\![\, \neg\varphi \,]\!] = \neg[\![\, \varphi \,]\!]$$

$$[\![\, \exists x\, \varphi(x, \vec{s}) \,]\!] = \bigvee_t [\![\, \varphi(t, \vec{s}) \,]\!]$$

$\mathbb{A}$-valued semantics

The semantics extend naturally to all assertions.

$$[\![\, \varphi \wedge \psi \,]\!] = [\![\, \varphi \,]\!] \wedge [\![\, \psi \,]\!]$$

$$[\![\, \neg\varphi \,]\!] = \neg[\![\, \varphi \,]\!]$$

$$[\![\, \exists x\, \varphi(x, \vec{s}) \,]\!] = \bigvee_t [\![\, \varphi(t, \vec{s}) \,]\!]$$

Thus, every assertion gets a $\mathbb{A}$-valued truth value.

The forcing theorem

If one uses a complete boolean algebra $\mathbb{B}$, then:

The forcing theorem

If one uses a complete boolean algebra $\mathbb{B}$, then:

Forcing theorem

Every axiom of ZFC gets truth value 1 in $V^{\mathbb{B}}$.

The forcing theorem

If one uses a complete boolean algebra $\mathbb{B}$, then:

Forcing theorem

Every axiom of ZFC gets truth value 1 in $V^{\mathbb{B}}$.

- For any names σ, τ, can make a name $\dot{p}$ for the pair $\{\sigma, \tau\}$:

$$\dot{p}(\sigma) = 1 \qquad \dot{p}(\tau) = 1$$

The forcing theorem

If one uses a complete boolean algebra $\mathbb{B}$, then:

Forcing theorem

Every axiom of ZFC gets truth value 1 in $V^{\mathbb{B}}$.

- For any names σ, τ, can make a name $\dot{p}$ for the pair $\{\, \sigma, \tau \,\}$:

$$\dot{p}(\sigma) = 1 \qquad \dot{p}(\tau) = 1$$

- For any name σ, there is a name for the power set $\dot{P}$:

$$\dot{P}(\tau) = [\![\, \tau \subseteq \sigma \,]\!] \text{ whenever } \tau : \mathrm{dom}(\sigma) \to \mathbb{A}$$

The forcing theorem

If one uses a complete boolean algebra $\mathbb{B}$, then:

Forcing theorem

Every axiom of ZFC gets truth value 1 in $V^{\mathbb{B}}$.

- For any names σ, τ, can make a name $\dot{p}$ for the pair $\{\, \sigma, \tau \,\}$:

$$\dot{p}(\sigma) = 1 \qquad \dot{p}(\tau) = 1$$

- For any name σ, there is a name for the power set $\dot{P}$:

$$\dot{P}(\tau) = [\![\, \tau \subseteq \sigma \,]\!] \text{ whenever } \tau : \mathrm{dom}(\sigma) \to \mathbb{A}$$

- Similarly build names for $\bigcup \sigma$, etc.

Forcing theorem—generalizations

- Complete Boolean algebra $\mathbb{B}$ gives ZFC with value 1.

Forcing theorem—generalizations

- Complete Boolean algebra $\mathbb{B}$ gives ZFC with value 1.
- Complete Heyting algebra $\mathbb{A}$ gives IZF with value 1.

Forcing theorem—generalizations

- Complete Boolean algebra $\mathbb{B}$ gives ZFC with value 1.
- Complete Heyting algebra $\mathbb{A}$ gives IZF with value 1.
- Paraconsistent algebras $\mathbb{A}$ give various paraconsistent versions of ZF.

Independence via forcing

Further amazing fact

Different choices of Boolean algebra $\mathbb{B}$ lead to different truth values of various central statements in set theory.

Independence via forcing

Further amazing fact

Different choices of Boolean algebra $\mathbb{B}$ lead to different truth values of various central statements in set theory.

1. $\mathbb{B} = \mathbb{2}$ is same as the hereditary function case. $V^{\mathbb{B}}$ is isomorphic to universe V in which name hierarchy is constructed.
2. Every nontrivial $\mathbb{B}$ has $[\![V \neq L]\!] = 1$.

Independence via forcing

Further amazing fact

Different choices of Boolean algebra $\mathbb{B}$ lead to different truth values of various central statements in set theory.

1. $\mathbb{B} = \mathbb{2}$ is same as the hereditary function case. $V^{\mathbb{B}}$ is isomorphic to universe V in which name hierarchy is constructed.
2. Every nontrivial $\mathbb{B}$ has $[\![V \neq L]\!] = 1$.
3. There is $\mathbb{B}$ for which $[\![\neg\mathrm{CH}]\!]^{\mathbb{B}} = 1$.

Independence via forcing

Further amazing fact

Different choices of Boolean algebra $\mathbb{B}$ lead to different truth values of various central statements in set theory.

1. $\mathbb{B} = \mathbb{2}$ is same as the hereditary function case. $V^{\mathbb{B}}$ is isomorphic to universe V in which name hierarchy is constructed.
2. Every nontrivial $\mathbb{B}$ has $[\![V \neq L]\!] = 1$.
3. There is $\mathbb{B}$ for which $[\![\neg\mathrm{CH}]\!]^{\mathbb{B}} = 1$.
4. Another for which $[\![\mathrm{CH}]\!] = 1$.

Independence via forcing

Further amazing fact

Different choices of Boolean algebra $\mathbb{B}$ lead to different truth values of various central statements in set theory.

1. $\mathbb{B} = \mathbb{2}$ is same as the hereditary function case. $V^{\mathbb{B}}$ is isomorphic to universe V in which name hierarchy is constructed.
2. Every nontrivial $\mathbb{B}$ has $[\![V \neq L]\!] = 1$.
3. There is $\mathbb{B}$ for which $[\![\neg\mathrm{CH}]\!]^{\mathbb{B}} = 1$.
4. Another for which $[\![\mathrm{CH}]\!] = 1$.
5. Another for which $[\![\mathrm{MA} + \neg\mathrm{CH}]\!] = 1$.

Independence via forcing

Further amazing fact

Different choices of Boolean algebra $\mathbb{B}$ lead to different truth values of various central statements in set theory.

1. $\mathbb{B} = \mathbb{2}$ is same as the hereditary function case. $V^{\mathbb{B}}$ is isomorphic to universe V in which name hierarchy is constructed.
2. Every nontrivial $\mathbb{B}$ has $[\![V \neq L]\!] = 1$.
3. There is $\mathbb{B}$ for which $[\![\neg\mathrm{CH}]\!]^{\mathbb{B}} = 1$.
4. Another for which $[\![\mathrm{CH}]\!] = 1$.
5. Another for which $[\![\mathrm{MA} + \neg\mathrm{CH}]\!] = 1$.

The hard work of forcing is to find a $\mathbb{B}$ that makes $[\![\sigma]\!] = 1$, where σ is a sentence you are interested in.

Two approaches to forcing

Boolean-valued models

- Clarifies the central metamathematical issues of forcing.
- Provides robust account of forcing semantics
- Realizes forcing as iterative conception in multi-valued logic.

Two approaches to forcing

Boolean-valued models

- Clarifies the central metamathematical issues of forcing.
- Provides robust account of forcing semantics
- Realizes forcing as iterative conception in multi-valued logic.
- Good for conceptualizing how forcing works in principle

Two approaches to forcing

Boolean-valued models

- Clarifies the central metamathematical issues of forcing.
- Provides robust account of forcing semantics
- Realizes forcing as iterative conception in multi-valued logic.
- Good for conceptualizing how forcing works in principle

Partial orders

- For actual use, when trying to force specific sentences.
- Design conditions as small pieces of desired generic object.
- Posets can have better combinatorics than Boolean algebras.
- Any poset is completed to a Boolean algebra.

Two approaches to forcing

Boolean-valued models

- Clarifies the central metamathematical issues of forcing.
- Provides robust account of forcing semantics
- Realizes forcing as iterative conception in multi-valued logic.
- Good for conceptualizing how forcing works in principle

Partial orders

- For actual use, when trying to force specific sentences.
- Design conditions as small pieces of desired generic object.
- Posets can have better combinatorics than Boolean algebras.
- Any poset is completed to a Boolean algebra.
- Good for actually using forcing for a specific purpose.

Two approaches to forcing

Boolean-valued models

- Clarifies the central metamathematical issues of forcing.
- Provides robust account of forcing semantics
- Realizes forcing as iterative conception in multi-valued logic.
- Good for conceptualizing how forcing works in principle

Partial orders

- For actual use, when trying to force specific sentences.
- Design conditions as small pieces of desired generic object.
- Posets can have better combinatorics than Boolean algebras.
- Any poset is completed to a Boolean algebra.
- Good for actually using forcing for a specific purpose.

Two approaches to forcing

Boolean-valued models

- Clarifies the central metamathematical issues of forcing.
- Provides robust account of forcing semantics
- Realizes forcing as iterative conception in multi-valued logic.
- Good for conceptualizing how forcing works in principle

Partial orders

- For actual use, when trying to force specific sentences.
- Design conditions as small pieces of desired generic object.
- Posets can have better combinatorics than Boolean algebras.
- Any poset is completed to a Boolean algebra.
- Good for actually using forcing for a specific purpose.

$\mathbb{B}$-valued $\to$ classical

Every $\mathbb{B}$-valued model easily transforms to a 2-valued model.

Simply use an ultrafilter $U \subseteq \mathbb{B}$ and define

$$\tau =_U \sigma \quad \iff \quad [\![\tau = \sigma]\!] \in U$$

$\mathbb{B}$-valued $\to$ classical

Every $\mathbb{B}$-valued model easily transforms to a 2-valued model.

Simply use an ultrafilter $U \subseteq \mathbb{B}$ and define

$$\tau =_U \sigma \quad \iff \quad [\![\tau = \sigma]\!] \in U$$

$$\tau \in_U \sigma \quad \iff \quad [\![\tau \in \sigma]\!] \in U$$

$\mathbb{B}$-valued $\to$ classical

Every $\mathbb{B}$-valued model easily transforms to a 2-valued model.

Simply use an ultrafilter $U \subseteq \mathbb{B}$ and define

$$\tau =_U \sigma \quad \iff \quad [\![\, \tau = \sigma \,]\!] \in U$$

$$\tau \in_U \sigma \quad \iff \quad [\![\, \tau \in \sigma \,]\!] \in U$$

Łoś lemma

For any complete Boolean algebra $\mathbb{B}$ and ultrafilter $U \subseteq \mathbb{B}$,

$$V^{\mathbb{B}}/U \models \varphi[\,[\tau_0]_U, \ldots, [\tau_n]_U] \quad \text{iff} \quad [\![\, \varphi(\tau_0, \ldots, \tau_n) \,]\!] \in U.$$

Proved just like the Łoś theorem for power-set ultraproducts.

Ultrapowers as an example

Consider structures M_i for $i \in I$,

Ultrapowers as an example

Consider structures M_i for $i \in I$, and form the product

$$\prod_i M_i = \{\, f \mid \mathrm{dom}(f) = I, \quad f(i) \in M_i \,\}.$$

Ultrapowers as an example

Consider structures M_i for $i \in I$, and form the product

$$\prod_i M_i = \{\, f \mid \mathrm{dom}(f) = I, \quad f(i) \in M_i \,\}.$$

Define atomic truth values and extend recursively

$$[\![\, \varphi(f) \,]\!] = \{\, i \in I \mid M_i \models \varphi[f(i)] \,\}.$$

Ultrapowers as an example

Consider structures M_i for $i \in I$, and form the product

$$\prod_i M_i = \{\, f \mid \mathrm{dom}(f) = I, \quad f(i) \in M_i \,\}.$$

Define atomic truth values and extend recursively

$$[\![\, \varphi(f) \,]\!] = \{\, i \in I \mid M_i \models \varphi[f(i)] \,\}.$$

This is a $\mathbb{B}$-valued model, where $\mathbb{B} = P(I)$, power set algebra.

Ultrapowers as an example

Consider structures M_i for $i \in I$, and form the product

$$\prod_i M_i = \{\, f \mid \mathrm{dom}(f) = I, \quad f(i) \in M_i \,\}.$$

Define atomic truth values and extend recursively

$$[\![\, \varphi(f) \,]\!] = \{\, i \in I \mid M_i \models \varphi[f(i)] \,\}.$$

This is a $\mathbb{B}$-valued model, where $\mathbb{B} = P(I)$, power set algebra.

Can quotient by an ultrafilter $U \subseteq \mathbb{B}$. Ultraproduct $\prod_i M_i / U$.

Ultrapowers as an example

Consider structures M_i for $i \in I$, and form the product

$$\prod_i M_i = \{\, f \mid \mathrm{dom}(f) = I, \quad f(i) \in M_i \,\}.$$

Define atomic truth values and extend recursively

$$[\![\, \varphi(f) \,]\!] = \{\, i \in I \mid M_i \models \varphi[f(i)] \,\}.$$

This is a $\mathbb{B}$-valued model, where $\mathbb{B} = P(I)$, power set algebra.

Can quotient by an ultrafilter $U \subseteq \mathbb{B}$. Ultraproduct $\prod_i M_i/U$.

Łoś theorem

$\prod_i M_i/U \models \varphi(f) \quad \iff \quad [\![\, \varphi(f) \,]\!] \in U.$

Original model embeds

We formed $V^{\mathbb{B}}$ in $\langle V, \in \rangle$.

Original model embeds

We formed $V^{\mathbb{B}}$ in $\langle V, \in \rangle$.

Can define check names

$$\check{x} = \{ \langle \check{y}, 1 \rangle \mid y \in x \}.$$

Original model embeds

We formed $V^{\mathbb{B}}$ in $\langle V, \in \rangle$.

Can define check names

$$\check{x} = \{ \langle \check{y}, 1 \rangle \mid y \in x \}.$$

Define the $\check{V}$ predicate

$$[\![\sigma \in \check{V}]\!] \quad = \quad \bigvee_{x \in V} [\![\sigma = \check{x}]\!]$$

Original model embeds

We formed $V^{\mathbb{B}}$ in $\langle V, \in \rangle$.

Can define check names

$$\check{x} = \{ \langle \check{y}, 1 \rangle \mid y \in x \}.$$

Define the $\check{V}$ predicate

$$[\![\sigma \in \check{V}]\!] \quad = \quad \bigvee_{x \in V} [\![\sigma = \check{x}]\!]$$

Elementary embedding induced by $x \mapsto \check{x}$. Boolean ultrapower.

$$\langle V, \in \rangle \models \varphi(x) \quad \text{iff} \quad [\![\varphi^{\check{V}}(\check{x})]\!]$$

Genericity

Can define the canonical name of the generic object.

$$\dot{G} = \{ \langle \check{b}, b \rangle \mid b \in \mathbb{B} \}$$

Genericity

Can define the canonical name of the generic object.

$$\dot{G} = \{\, \langle \check{b}, b \rangle \mid b \in \mathbb{B} \,\}$$

It turns out that

$$[\![\, \dot{\mathbb{G}} \subseteq \check{\mathbb{B}} \text{ is an ultrafilter }]\!] = 1.$$

Genericity

Can define the canonical name of the generic object.

$$\dot{G} = \{ \langle \check{b}, b \rangle \mid b \in \mathbb{B} \}$$

It turns out that

$$[\![\dot{\mathbb{G}} \subseteq \check{\mathbb{B}} \text{ is an ultrafilter }]\!] = 1.$$

Moreover,

$$[\![\dot{G} \subseteq \check{\mathbb{B}} \text{ is } \check{V}\text{-generic}]\!] = 1.$$

Genericity

Can define the canonical name of the generic object.

$$\dot{G} = \{\, \langle \check{b}, b \rangle \mid b \in \mathbb{B} \,\}$$

It turns out that

$$[\![\, \dot{\mathbb{G}} \subseteq \check{\mathbb{B}} \text{ is an ultrafilter }]\!] = 1.$$

Moreover,

$$[\![\, \dot{G} \subseteq \check{\mathbb{B}} \text{ is } \check{V}\text{-generic} \,]\!] = 1.$$

Can also show that the value recursion works with value 1.

$$[\![\, \tau = \mathrm{val}(\check{\tau}, \dot{G}) \,]\!] = 1$$

Genericity

Can define the canonical name of the generic object.

$$\dot{G} = \{ \langle \check{b}, b \rangle \mid b \in \mathbb{B} \}$$

It turns out that

$$[\![\dot{\mathbb{G}} \subseteq \check{\mathbb{B}} \text{ is an ultrafilter }]\!] = 1.$$

Moreover,

$$[\![\dot{G} \subseteq \check{\mathbb{B}} \text{ is } \check{V}\text{-generic}]\!] = 1.$$

Can also show that the value recursion works with value 1.

$$[\![\tau = \operatorname{val}(\check{\tau}, \dot{G})]\!] = 1$$

In other words, with truth value 1, the universe is $\check{V}[\dot{G}]$.

Genericity is an afterthought

I want to emphasize:

All discussion of $\dot{G}$ enters *after the fact*.

Genericity is an afterthought

I want to emphasize:

All discussion of $\dot{G}$ enters *after the fact*.

We had already constructed the universe $V^{\mathbb{B}}/U$ and defined truth there, including the copy of V via $\check{V}/U$, without any mention of genericity or $\dot{G}$.

Genericity is an afterthought

I want to emphasize:

All discussion of $\dot{G}$ enters *after the fact*.

We had already constructed the universe $V^{\mathbb{B}}/U$ and defined truth there, including the copy of V via $\check{V}/U$, without any mention of genericity or $\dot{G}$.

This is the sense in which one can understand forcing without any talk of genericity.

Genericity is an afterthought

I want to emphasize:

All discussion of $\dot{G}$ enters *after the fact*.

We had already constructed the universe $V^{\mathbb{B}}/U$ and defined truth there, including the copy of V via $\check{V}/U$, without any mention of genericity or $\dot{G}$.

This is the sense in which one can understand forcing without any talk of genericity.

We did not form $V^{\mathbb{B}}/U$ by augmenting the ground model $\check{V}$ with an ideal object. Rather, we built it by implementing the iterative conception in $\mathbb{B}$-valued logic to get $V^{\mathbb{B}}$. We used the ultrafilter U to quotient this to a classical model.

Reasoning in Boolean logic

Dana Scott had mentioned, in the earliest days of forcing, that part of the difficulty of forcing would be learning how to reason under the Boolean brackets

$$[\![\varphi]\!].$$

Reasoning in Boolean logic

Dana Scott had mentioned, in the earliest days of forcing, that part of the difficulty of forcing would be learning how to reason under the Boolean brackets

$$[\![\varphi]\!].$$

This turns out not to be true.

Reasoning in Boolean logic

Dana Scott had mentioned, in the earliest days of forcing, that part of the difficulty of forcing would be learning how to reason under the Boolean brackets

$$[\![\varphi]\!].$$

This turns out not to be true.

Reasoning inside the Boolean brackets $[\![\cdots]\!]$ is largely identical to classical reasoning, since classical logic gets value 1.

Reasoning in Boolean logic

Dana Scott had mentioned, in the earliest days of forcing, that part of the difficulty of forcing would be learning how to reason under the Boolean brackets

$$[\![\, \varphi \,]\!].$$

This turns out not to be true.

Reasoning inside the Boolean brackets $[\![\cdots]\!]$ is largely identical to classical reasoning, since classical logic gets value 1.

If premises φ etc. hold with Boolean value at least *b*, and those premises imply ψ in classical logic, then ψ holds with value at least *b*.

Reasoning in Boolean logic

Dana Scott had mentioned, in the earliest days of forcing, that part of the difficulty of forcing would be learning how to reason under the Boolean brackets

$$[\![\, \varphi \,]\!].$$

This turns out not to be true.

Reasoning inside the Boolean brackets $[\![\, \cdots \,]\!]$ is largely identical to classical reasoning, since classical logic gets value 1.

If premises φ etc. hold with Boolean value at least *b*, and those premises imply ψ in classical logic, then ψ holds with value at least *b*.

So it is actually quite easy to reason under the Boolean brackets.

Naturalist account of forcing

Given a set theoretic universe V, we can write down the theory of what it would be like to live in the forcing extension $V[G]$ for a V-generic ultrafilter $G \subseteq \mathbb{B}$ for some forcing notion $\mathbb{B}$.

Naturalist account of forcing

Given a set theoretic universe V, we can write down the theory of what it would be like to live in the forcing extension $V[G]$ for a V-generic ultrafilter $G \subseteq \mathbb{B}$ for some forcing notion $\mathbb{B}$.

1. Introduce a predicate symbol $\check{V}$ for the ground model, constants $\check{x}$ for every $x \in V$, constant $\dot{G}$ for the generic.

Naturalist account of forcing

Given a set theoretic universe V, we can write down the theory of what it would be like to live in the forcing extension $V[G]$ for a V-generic ultrafilter $G \subseteq \mathbb{B}$ for some forcing notion $\mathbb{B}$.

1. Introduce a predicate symbol $\check{V}$ for the ground model, constants $\check{x}$ for every $x \in V$, constant $\dot{G}$ for the generic.
2. Assert $\check{V}$ is a transitive class containing all ordinals.

Naturalist account of forcing

Given a set theoretic universe V, we can write down the theory of what it would be like to live in the forcing extension $V[G]$ for a V-generic ultrafilter $G \subseteq \mathbb{B}$ for some forcing notion $\mathbb{B}$.

1. Introduce a predicate symbol $\check{V}$ for the ground model, constants $\check{x}$ for every $x \in V$, constant $\dot{G}$ for the generic.
2. Assert $\check{V}$ is a transitive class containing all ordinals.
3. Assert $\dot{G} \subseteq \check{\mathbb{B}}$ is $\check{V}$-generic.

Naturalist account of forcing

Given a set theoretic universe V, we can write down the theory of what it would be like to live in the forcing extension $V[G]$ for a V-generic ultrafilter $G \subseteq \mathbb{B}$ for some forcing notion $\mathbb{B}$.

1. Introduce a predicate symbol $\check{V}$ for the ground model, constants $\check{x}$ for every $x \in V$, constant $\dot{G}$ for the generic.
2. Assert $\check{V}$ is a transitive class containing all ordinals.
3. Assert $\dot{G} \subseteq \check{\mathbb{B}}$ is $\check{V}$-generic.
4. Assert that the universe is $\check{V}[\dot{G}]$.

Naturalist account of forcing

Given a set theoretic universe V, we can write down the theory of what it would be like to live in the forcing extension $V[G]$ for a V-generic ultrafilter $G \subseteq \mathbb{B}$ for some forcing notion $\mathbb{B}$.

1. Introduce a predicate symbol $\check{V}$ for the ground model, constants $\check{x}$ for every $x \in V$, constant $\dot{G}$ for the generic.
2. Assert $\check{V}$ is a transitive class containing all ordinals.
3. Assert $\dot{G} \subseteq \check{\mathbb{B}}$ is $\check{V}$-generic.
4. Assert that the universe is $\check{V}[\dot{G}]$.
5. For each truth $\varphi(x)$ of V, assert $\varphi^{\check{V}}(\check{x})$, that is, φ holds in ground model.

Naturalist account of forcing

Given a set theoretic universe V, we can write down the theory of what it would be like to live in the forcing extension $V[G]$ for a V-generic ultrafilter $G \subseteq \mathbb{B}$ for some forcing notion $\mathbb{B}$.

1. Introduce a predicate symbol $\check{V}$ for the ground model, constants $\check{x}$ for every $x \in V$, constant $\dot{G}$ for the generic.
2. Assert $\check{V}$ is a transitive class containing all ordinals.
3. Assert $\dot{G} \subseteq \check{\mathbb{B}}$ is $\check{V}$-generic.
4. Assert that the universe is $\check{V}[\dot{G}]$.
5. For each truth $\varphi(x)$ of V, assert $\varphi^{\check{V}}(\check{x})$, that is, φ holds in ground model.
6. Follows that all statements forced over V hold in the theory.

Naturalist account of forcing

Given a set theoretic universe V, we can write down the theory of what it would be like to live in the forcing extension $V[G]$ for a V-generic ultrafilter $G \subseteq \mathbb{B}$ for some forcing notion $\mathbb{B}$.

1. Introduce a predicate symbol $\check{V}$ for the ground model, constants $\check{x}$ for every $x \in V$, constant $\dot{G}$ for the generic.
2. Assert $\check{V}$ is a transitive class containing all ordinals.
3. Assert $\dot{G} \subseteq \check{\mathbb{B}}$ is $\check{V}$-generic.
4. Assert that the universe is $\check{V}[\dot{G}]$.
5. For each truth $\varphi(x)$ of V, assert $\varphi^{\check{V}}(\check{x})$, that is, φ holds in ground model.
6. Follows that all statements forced over V hold in the theory.

The theory expresses what it would be like to live in $V[G]$.

Let us show it is consistent.

Naturalist account is consistent

Naturalist account

For any model of ZFC, the theory expressed by the naturalist account of forcing over that model is consistent.

Naturalist account is consistent

Naturalist account

For any model of ZFC, the theory expressed by the naturalist account of forcing over that model is consistent.

This is exactly what the $\mathbb{B}$-valued approach to forcing establishes. The naturalist account theory is true in $V^{\mathbb{B}}$ with Boolean value 1.

Naturalist account is consistent

Naturalist account

For any model of ZFC, the theory expressed by the naturalist account of forcing over that model is consistent.

This is exactly what the $\mathbb{B}$-valued approach to forcing establishes. The naturalist account theory is true in $V^{\mathbb{B}}$ with Boolean value 1.

So there is no need to have actual generic filters. One can act as though you have them, everything you want, since this is what the theory of the naturalist account of forcing expresses.

Naturalist account is consistent

Naturalist account

For any model of ZFC, the theory expressed by the naturalist account of forcing over that model is consistent.

This is exactly what the $\mathbb{B}$-valued approach to forcing establishes. The naturalist account theory is true in $V^{\mathbb{B}}$ with Boolean value 1.

So there is no need to have actual generic filters. One can act as though you have them, everything you want, since this is what the theory of the naturalist account of forcing expresses.

To say, "Let G be V-generic; work in $V[G]$" is exactly to make this move, to adopt the naturalist account of forcing.

Pervasive mutual interpretability phenomenon

Interpretation via inner models

Can often interpret a theory T by finding an inner model of T.

Pervasive mutual interpretability phenomenon

Interpretation via inner models

Can often interpret a theory T by finding an inner model of T.

- ZFC $+\,\mathrm{CH}$ is interpretable in ZFC.
- Indeed, GCH, $V = L$ is interpretable in ZFC. And more.

Pervasive mutual interpretability phenomenon

Interpretation via inner models

Can often interpret a theory T by finding an inner model of T.

- $\text{ZFC} + \text{CH}$ is interpretable in ZFC.
- Indeed, GCH, $V = L$ is interpretable in ZFC. And more.

Interpretation via forcing

Forcing also provides interpretations via $V^{\mathbb{B}}/U$.

- $\text{ZFC} + \neg\text{CH}$ is interpretable in ZFC.
- $\text{ZFC} + \text{MA}_{\omega_1}$ is interpretable in ZFC.
- Every forceable theory is interpretable in ZFC.

Pervasive mutual interpretability phenomenon

Interpretation via inner models

Can often interpret a theory T by finding an inner model of T.

- $\mathrm{ZFC} + \mathrm{CH}$ is interpretable in ZFC.
- Indeed, GCH, $V = L$ is interpretable in ZFC. And more.

Interpretation via forcing

Forcing also provides interpretations via $V^{\mathbb{B}}/U$.

- $\mathrm{ZFC} + \neg\mathrm{CH}$ is interpretable in ZFC.
- $\mathrm{ZFC} + \mathrm{MA}_{\omega_1}$ is interpretable in ZFC.
- Every forceable theory is interpretable in ZFC.

Abstractly, the naturalist account of forcing sets up an interpretation of the forced theory.

But no bi-interpretation phenomenon

So we have numerous instances of mutual interpretation of set theories extending ZFC in diverse ways.

But no bi-interpretation phenomenon

So we have numerous instances of mutual interpretation of set theories extending ZFC in diverse ways.

Meanwhile, there is no bi-interpretation phenomenon.

Theorem (Enayat)

No two extensions of ZFC are bi-interpretable, unless equal.

But no bi-interpretation phenomenon

So we have numerous instances of mutual interpretation of set theories extending ZFC in diverse ways.

Meanwhile, there is no bi-interpretation phenomenon.

Theorem (Enayat)

No two extensions of ZFC are bi-interpretable, unless equal.

There are various edge cases with weakenings to Zermelo set theory ZC, and to ZFC^-.

Forcing potentialism

We can actually define the forcing modal operators in ZFC.

Forcing potentialism

We can actually define the forcing modal operators in ZFC.

- $\Diamond\varphi$ means $\exists\mathbb{B}\,[\![\,\varphi\,]\!]^{\mathbb{B}} = 1$.

Forcing potentialism

We can actually define the forcing modal operators in ZFC.

- $\Diamond\varphi$ means $\exists\mathbb{B}\,[\![\varphi]\!]^{\mathbb{B}} = 1$.
- $\Box\varphi$ means $\forall\mathbb{B}\,[\![\varphi]\!]^{\mathbb{B}} = 1$.

Forcing potentialism

We can actually define the forcing modal operators in ZFC.

- $\Diamond\varphi$ means $\exists\mathbb{B}\ [\![\varphi]\!]^{\mathbb{B}} = 1$.
- $\Box\varphi$ means $\forall\mathbb{B}\ [\![\varphi]\!]^{\mathbb{B}} = 1$.

That is, we can express the modalities in the modality-free base language.

Forcing potentialism

We can actually define the forcing modal operators in ZFC.

- $\Diamond\varphi$ means $\exists\mathbb{B}\ [\![\varphi]\!]^{\mathbb{B}} = 1$.
- $\Box\varphi$ means $\forall\mathbb{B}\ [\![\varphi]\!]^{\mathbb{B}} = 1$.

That is, we can express the modalities in the modality-free base language.

In this sense, the forcing modalities are internal concepts of set theory—not metaphysical or metamathematical.

Forcing potentialism

We can actually define the forcing modal operators in ZFC.

- $\Diamond\varphi$ means $\exists\mathbb{B}\ [\![\varphi]\!]^{\mathbb{B}} = 1$.
- $\Box\varphi$ means $\forall\mathbb{B}\ [\![\varphi]\!]^{\mathbb{B}} = 1$.

That is, we can express the modalities in the modality-free base language.

In this sense, the forcing modalities are internal concepts of set theory—not metaphysical or metamathematical.

In particular, one can speak of forceability over any model of ZFC. (No need to restrict to countable transitive models.)

Modality deflationism

Forcing modalities $\Diamond\varphi$, $\Box\varphi$ are expressible in the non-modal language of set theory.

Modality deflationism

Forcing modalities $\Diamond\varphi$, $\Box\varphi$ are expressible in the non-modal language of set theory.

This is quite different from many other notions of potentialism.

- Not true in set-theoretic rank-potentialism.

Modality deflationism

Forcing modalities $\Diamond\varphi$, $\Box\varphi$ are expressible in the non-modal language of set theory.

This is quite different from many other notions of potentialism.

- Not true in set-theoretic rank-potentialism.
- Not true in Aristotelian arithmetic potentialism.

Modality deflationism

Forcing modalities $\Diamond\varphi$, $\Box\varphi$ are expressible in the non-modal language of set theory.

This is quite different from many other notions of potentialism.

- Not true in set-theoretic rank-potentialism.
- Not true in Aristotelian arithmetic potentialism.
- Not true in models-of-arithmetic end-extensional potentialism. (but almost...)

Modality deflationism

Forcing modalities $\Diamond\varphi$, $\Box\varphi$ are expressible in the non-modal language of set theory.

This is quite different from many other notions of potentialism.

- Not true in set-theoretic rank-potentialism.
- Not true in Aristotelian arithmetic potentialism.
- Not true in models-of-arithmetic end-extensional potentialism. (but almost...)

Zermelo potentialism

Models of $ZFCU_2$. Two modalities:

- $\Diamond\varphi$, increase the height.

Modality deflationism

Forcing modalities $\Diamond\varphi$, $\Box\varphi$ are expressible in the non-modal language of set theory.

This is quite different from many other notions of potentialism.

- Not true in set-theoretic rank-potentialism.
- Not true in Aristotelian arithmetic potentialism.
- Not true in models-of-arithmetic end-extensional potentialism. (but almost...)

Zermelo potentialism

Models of $ZFCU_2$. Two modalities:

- $\Diamond\varphi$, increase the height.
- $\Diamond\varphi$, add urelements.

The upward modality $\Diamond\varphi$ is not expressible in set theory, but the adding-more-urelements modality $\Diamond\varphi$ IS expressible.

Forcing validities

A modal assertion $\varphi(p_1, \ldots, p_n)$ is *valid for forcing*, if all substitution instances $\varphi(\psi_1, \ldots, \psi_n)$ hold in every model of set theory, for any set-theoretic assertions ψ_i.

Forcing validities

A modal assertion $\varphi(p_1, \ldots, p_n)$ is *valid for forcing*, if all substitution instances $\varphi(\psi_1, \ldots, \psi_n)$ hold in every model of set theory, for any set-theoretic assertions ψ_i.

Theorem (Hamkins,Löwe)

The ZFC-provably valid principles of forcing are exactly the assertions of S4.2.

Summary

I have described a certain way of thinking about forcing.

- not about augmenting a model with an “ideal” object

Summary

I have described a certain way of thinking about forcing.

- not about augmenting a model with an “ideal” object
- not about genericity or meeting dense sets

Summary

I have described a certain way of thinking about forcing.

- not about augmenting a model with an “ideal” object
- not about genericity or meeting dense sets
- not about countable models

Summary

I have described a certain way of thinking about forcing.

- not about augmenting a model with an “ideal” object
- not about genericity or meeting dense sets
- not about countable models
- not necessarily metatheoretic

Summary

I have described a certain way of thinking about forcing.

- not about augmenting a model with an “ideal” object
- not about genericity or meeting dense sets
- not about countable models
- not necessarily metatheoretic

Instead, the Boolean-valued approach to forcing

- Reveals forcing simply as the iterative conception, undertaken in multivalued logic

Summary

I have described a certain way of thinking about forcing.

- not about augmenting a model with an “ideal” object
- not about genericity or meeting dense sets
- not about countable models
- not necessarily metatheoretic

Instead, the Boolean-valued approach to forcing

- Reveals forcing simply as the iterative conception, undertaken in multivalued logic
- Is not metatheoretic, but makes sense internally to ZFC

Summary

I have described a certain way of thinking about forcing.

- not about augmenting a model with an “ideal” object
- not about genericity or meeting dense sets
- not about countable models
- not necessarily metatheoretic

Instead, the Boolean-valued approach to forcing

- Reveals forcing simply as the iterative conception, undertaken in multivalued logic
- Is not metatheoretic, but makes sense internally to ZFC
- Therefore makes sense over any model of ZFC

Summary

I have described a certain way of thinking about forcing.

- not about augmenting a model with an "ideal" object
- not about genericity or meeting dense sets
- not about countable models
- not necessarily metatheoretic

Instead, the Boolean-valued approach to forcing

- Reveals forcing simply as the iterative conception, undertaken in multivalued logic
- Is not metatheoretic, but makes sense internally to ZFC
- Therefore makes sense over any model of ZFC
- Leads consequently to a rich mutual interpretation phenomenon

Thank you.

Slides and articles available on http://jdh.hamkins.org.

Joel David Hamkins
O'Hara Professor of Logic
University of Notre Dame

VRF Mathematical Intitute
University of Oxford