Iterated forcing and Forcing Axioms

Mirna Džamonja, Tutorial 2

26 June, 2019

Forcing and iterated forcing

Forcing is a technique to extend a universe **V** of set theory=ZFC (or ZFC*) to another one, $\mathbf{V}[G]$, so that $\mathbf{V}[G]$

Forcing and iterated forcing

Forcing is a technique to extend a universe **V** of set theory=ZFC (or ZFC*) to another one, $\mathbf{V}[G]$, so that $\mathbf{V}[G]$

- has the same ordinals

Forcing and iterated forcing

Forcing is a technique to extend a universe **V** of set theory=ZFC (or ZFC*) to another one, $\mathbf{V}[G]$, so that $\mathbf{V}[G]$

- has the same ordinals
- (most often) has the same cardinals

Forcing and iterated forcing

Forcing is a technique to extend a universe **V** of set theory=ZFC (or ZFC*) to another one, **V**$[G]$, so that **V**$[G]$

- has the same ordinals
- (most often) has the same cardinals
- satisfies a desired formula ϕ.

Forcing and iterated forcing

Forcing is a technique to extend a universe **V** of set theory=ZFC (or ZFC*) to another one, **V**$[G]$, so that **V**$[G]$

- has the same ordinals
- (most often) has the same cardinals
- satisfies a desired formula ϕ.

For example, ϕ could be the failure of CH, or something more involved such as

Forcing and iterated forcing

Forcing is a technique to extend a universe **V** of set theory=ZFC (or ZFC*) to another one, **V**$[G]$, so that **V**$[G]$

- has the same ordinals
- (most often) has the same cardinals
- satisfies a desired formula ϕ.

For example, ϕ could be the failure of CH, or something more involved such as there are no Suslin trees or

Forcing and iterated forcing

Forcing is a technique to extend a universe **V** of set theory=ZFC (or ZFC*) to another one, **V**$[G]$, so that **V**$[G]$

- has the same ordinals
- (most often) has the same cardinals
- satisfies a desired formula ϕ.

For example, ϕ could be the failure of CH, or something more involved such as there are no Suslin trees or “every ccc Boolean algebra of size $< \mathfrak{c}$ supports a measure”.

Forcing and iterated forcing

Forcing is a technique to extend a universe **V** of set theory=ZFC (or ZFC*) to another one, **V**$[G]$, so that **V**$[G]$

- has the same ordinals
- (most often) has the same cardinals
- satisfies a desired formula ϕ.

For example, ϕ could be the failure of CH, or something more involved such as there are no Suslin trees or "every ccc Boolean algebra of size $< \mathfrak{c}$ supports a measure". For such more involved statements $\neg\exists$ or $\forall\exists$ we need to use iterated forcing.

Forcing and iterated forcing

Forcing is a technique to extend a universe **V** of set theory=ZFC (or ZFC*) to another one, **V**$[G]$, so that **V**$[G]$

- has the same ordinals
- (most often) has the same cardinals
- satisfies a desired formula ϕ.

For example, ϕ could be the failure of CH, or something more involved such as there are no Suslin trees or "every ccc Boolean algebra of size $< \mathfrak{c}$ supports a measure".
For such more involved statements $\neg\exists$ or $\forall\exists$ we need to use iterated forcing.
This can get a little complicated because of two issues: preservation of the axioms and preservation of cardinals.

Iterated forcing

Iteration of forcing is NOT constructing a sequence of extensions $V_0 \subseteq V_1 = V_0[G_0] \subseteq V_1[G_1] \ldots \subseteq V_\alpha[G_\alpha] \ldots$ such that each G_α is P_α-generic for some forcing $\mathbb{P}_\alpha \in V_\alpha$ chosen independently of the previous ones.

Iterated forcing

Iteration of forcing is NOT constructing a sequence of extensions $V_0 \subseteq V_1 = V_0[G_0] \subseteq V_1[G_1] \ldots \subseteq V_\alpha[G_\alpha] \ldots$ such that each G_α is P_α-generic for some forcing $\mathbb{P}_\alpha \in V_\alpha$ chosen independently of the previous ones.

This approach runs into problems already at the stage ω.

Iterated forcing

Iteration of forcing is NOT constructing a sequence of extensions $V_0 \subseteq V_1 = V_0[G_0] \subseteq V_1[G_1] \ldots \subseteq V_\alpha[G_\alpha] \ldots$ such that each G_α is P_α-generic for some forcing $\mathbb{P}_\alpha \in V_\alpha$ chosen independently of the previous ones.

This approach runs into problems already at the stage ω. For example, if we take $V_\omega = \bigcup V_n$, in general this will not even contain $\langle G_n : n < \omega \rangle$.

Iterated forcing

Iteration of forcing is NOT constructing a sequence of extensions $V_0 \subseteq V_1 = V_0[G_0] \subseteq V_1[G_1] \ldots \subseteq V_\alpha[G_\alpha] \ldots$ such that each G_α is P_α-generic for some forcing $\mathbb{P}_\alpha \in V_\alpha$ chosen independently of the previous ones.

This approach runs into problems already at the stage ω. For example, if we take $V_\omega = \bigcup V_n$, in general this will not even contain $\langle G_n : n < \omega \rangle$. See Kunen's book 1st edition.

Iterated forcing

Iteration of forcing is NOT constructing a sequence of extensions $V_0 \subseteq V_1 = V_0[G_0] \subseteq V_1[G_1] \ldots \subseteq V_\alpha[G_\alpha] \ldots$ such that each G_α is P_α-generic for some forcing $\mathbb{P}_\alpha \in V_\alpha$ chosen independently of the previous ones.

This approach runs into problems already at the stage ω. For example, if we take $V_\omega = \bigcup V_n$, in general this will not even contain $\langle G_n : n < \omega \rangle$. See Kunen's book 1st edition.

Instead we need to deal with a forcing which is entirely in V_0 and consists of **names**.

Names

Definition

Names are defined recursively. A $\mathbb{P}$-name is a set of pairs $(p, \underset{\sim}{\sigma})$ where $p \in \mathbb{P}$ and $\underset{\sim}{\sigma}$ is a $\mathbb{P}$-name.

Names

Definition

Names are defined recursively. A $\mathbb{P}$-name is a set of pairs $(p, \underset{\sim}{\sigma})$ where $p \in \mathbb{P}$ and $\underset{\sim}{\sigma}$ is a $\mathbb{P}$-name.

Names, say $\underset{\sim}{\tau}$, are in V and are going to be calculated into objects, $\underset{\sim}{\tau}_G$ in $V[G]$. In fact, $V[G]$ consists entirely of such calculated objects.

Names

Definition

Names are defined recursively. A $\mathbb{P}$-name is a set of pairs $(p, \underset{\sim}{\sigma})$ where $p \in \mathbb{P}$ and $\underset{\sim}{\sigma}$ is a $\mathbb{P}$-name.

Names, say $\underset{\sim}{\tau}$, are in V and are going to be calculated into objects, $\underset{\sim}{\tau}_G$ in $V[G]$. In fact, $V[G]$ consists entirely of such calculated objects.

Definition

(1) The values of $\underset{\sim}{\tau}_G$ are defined recursively. If G is a filter on $\mathbb{P}$ and $\underset{\sim}{\tau}$ a $\mathbb{P}$-name, then

$$\underset{\sim}{\tau}_G = \{\underset{\sim}{\sigma}_G : (\exists p \in G)(p, \underset{\sim}{\sigma}) \in \underset{\sim}{\tau}\}.$$

Names

Definition

Names are defined recursively. A $\mathbb{P}$-name is a set of pairs $(p, \underset{\sim}{\sigma})$ where $p \in \mathbb{P}$ and $\underset{\sim}{\sigma}$ is a $\mathbb{P}$-name.

Names, say $\underset{\sim}{\tau}$, are in V and are going to be calculated into objects, $\underset{\sim}{\tau}_G$ in $V[G]$. In fact, $V[G]$ consists entirely of such calculated objects.

Definition

(1) The values of $\underset{\sim}{\tau}_G$ are defined recursively. If G is a filter on $\mathbb{P}$ and $\underset{\sim}{\tau}$ a $\mathbb{P}$-name, then

$$\underset{\sim}{\tau}_G = \{\underset{\sim}{\sigma}_G : (\exists p \in G)(p, \underset{\sim}{\sigma}) \in \underset{\sim}{\tau}\}.$$

(2) $V[G] = \{\underset{\sim}{\tau}_G : \underset{\sim}{\tau} \in V\}$.

The forcing relation

Definition

Suppose that $\varphi(x_0, x_1, \ldots x_{n-1})$ is a formula in the language of set theory $\mathcal{L} = \{\in\}$ and $\underset{\sim}{\tau}^0, \underset{\sim}{\tau}^1, \ldots \underset{\sim}{\tau}^{n-1}$ are $\mathbb{P}$-names, while $p \in \mathbb{P}$.

The forcing relation

Definition

Suppose that $\varphi(x_0, x_1, \dots x_{n-1})$ is a formula in the language of set theory $\mathcal{L} = \{\in\}$ and $\underset{\sim}{\tau}^0, \underset{\sim}{\tau}^1, \dots \underset{\sim}{\tau}^{n-1}$ are $\mathbb{P}$-names, while $p \in \mathbb{P}$. We say that $p \Vdash \varphi(\underset{\sim}{\tau}_0, \underset{\sim}{\tau}_1, \dots \underset{\sim}{\tau}_{n-1})$ iff

The forcing relation

Definition

Suppose that $\varphi(x_0, x_1, \ldots x_{n-1})$ is a formula in the language of set theory $\mathcal{L} = \{\in\}$ and $\underset{\sim}{\tau}^0, \underset{\sim}{\tau}^1, \ldots \underset{\sim}{\tau}^{n-1}$ are $\mathbb{P}$-names, while $p \in \mathbb{P}$. We say that $p \Vdash \varphi(\underset{\sim}{\tau}_0, \underset{\sim}{\tau}_1, \ldots \underset{\sim}{\tau}_{n-1})$ iff for every relevant V, for every $\mathbb{P}$-generic G over V, we have $V[G] \models \varphi(\underset{\sim}{\tau}^0_G, \underset{\sim}{\tau}^1_G, \ldots \underset{\sim}{\tau}^{n-1}_G)$.

The forcing relation

Definition

Suppose that $\varphi(x_0, x_1, \dots x_{n-1})$ is a formula in the language of set theory $\mathcal{L} = \{\in\}$ and $\underset{\sim}{\tau}^0, \underset{\sim}{\tau}^1, \dots \underset{\sim}{\tau}^{n-1}$ are $\mathbb{P}$-names, while $p \in \mathbb{P}$. We say that $p \Vdash \varphi(\underset{\sim}{\tau}_0, \underset{\sim}{\tau}_1, \dots \underset{\sim}{\tau}_{n-1})$ iff for every relevant V, for every $\mathbb{P}$-generic G over V, we have $V[G] \models \varphi(\underset{\sim}{\tau}^0_G, \underset{\sim}{\tau}^1_G, \dots \underset{\sim}{\tau}^{n-1}_G)$.

A good definition but seems too hard to satisfy.

The forcing relation

Definition

Suppose that $\varphi(x_0, x_1, \ldots x_{n-1})$ is a formula in the language of set theory $\mathcal{L} = \{\in\}$ and $\underset{\sim}{\tau}^0, \underset{\sim}{\tau}^1, \ldots \underset{\sim}{\tau}^{n-1}$ are $\mathbb{P}$-names, while $p \in \mathbb{P}$. We say that $p \Vdash \varphi(\underset{\sim}{\tau}_0, \underset{\sim}{\tau}_1, \ldots \underset{\sim}{\tau}_{n-1})$ iff for every relevant V, for every $\mathbb{P}$-generic G over V, we have $V[G] \models \varphi(\underset{\sim}{\tau}^0_G, \underset{\sim}{\tau}^1_G, \ldots \underset{\sim}{\tau}^{n-1}_G)$.

A good definition but seems too hard to satisfy. That is where the magic comes:

The forcing relation

Definition

Suppose that $\varphi(x_0, x_1, \dots x_{n-1})$ is a formula in the language of set theory $\mathcal{L} = \{\in\}$ and $\underset{\sim}{\tau}^0, \underset{\sim}{\tau}^1, \dots \underset{\sim}{\tau}^{n-1}$ are $\mathbb{P}$-names, while $p \in \mathbb{P}$. We say that $p \Vdash \varphi(\underset{\sim}{\tau}_0, \underset{\sim}{\tau}_1, \dots \underset{\sim}{\tau}_{n-1})$ iff for every relevant V, for every $\mathbb{P}$-generic G over V, we have $V[G] \models \varphi(\underset{\sim}{\tau}^0_G, \underset{\sim}{\tau}^1_G, \dots \underset{\sim}{\tau}^{n-1}_G)$.

A good definition but seems too hard to satisfy. That is where the magic comes:

Forcing Theorem, part 2 (1) For any φ and $\underset{\sim}{\tau}^0, \underset{\sim}{\tau}^1, \dots \underset{\sim}{\tau}^{n-1}$, V as above, $V[G] \models \varphi(\underset{\sim}{\tau}^0_G, \underset{\sim}{\tau}^1_G, \dots \underset{\sim}{\tau}^{n-1}_G)$ iff for some $p \in G$ we have $p \Vdash \varphi(\underset{\sim}{\tau}_0, \underset{\sim}{\tau}_1, \dots \underset{\sim}{\tau}_{n-1})$.

The forcing relation

Definition

Suppose that $\varphi(x_0, x_1, \ldots x_{n-1})$ is a formula in the language of set theory $\mathcal{L} = \{\in\}$ and $\underset{\sim}{\tau}^0, \underset{\sim}{\tau}^1, \ldots \underset{\sim}{\tau}^{n-1}$ are $\mathbb{P}$-names, while $p \in \mathbb{P}$. We say that $p \Vdash \varphi(\underset{\sim}{\tau}_0, \underset{\sim}{\tau}_1, \ldots \underset{\sim}{\tau}_{n-1})$ iff for every relevant V, for every $\mathbb{P}$-generic G over V, we have $V[G] \models \varphi(\underset{\sim}{\tau}^0_G, \underset{\sim}{\tau}^1_G, \ldots \underset{\sim}{\tau}^{n-1}_G)$.

A good definition but seems too hard to satisfy. That is where the magic comes:

Forcing Theorem, part 2 (1) For any φ and $\underset{\sim}{\tau}^0, \underset{\sim}{\tau}^1, \ldots \underset{\sim}{\tau}^{n-1}$, V as above, $V[G] \models \varphi(\underset{\sim}{\tau}^0_G, \underset{\sim}{\tau}^1_G, \ldots \underset{\sim}{\tau}^{n-1}_G)$ iff for some $p \in G$ we have $p \Vdash \varphi(\underset{\sim}{\tau}_0, \underset{\sim}{\tau}_1, \ldots \underset{\sim}{\tau}_{n-1})$.
(2) There is a relation $\Vdash^*$ definable in the ground model such that $p \Vdash \varphi$ iff $p \Vdash^* \varphi$.

Iterated Forcing

Two Step Iteration Suppose that P is a forcing notion and $\underset{\sim}{Q}$ is a P-name for a forcing notion (i.e. $\emptyset_P \Vdash \underset{\sim}{Q}$ is a forcing notion.)

Iterated Forcing

Two Step Iteration Suppose that P is a forcing notion and $\underset{\sim}{Q}$ is a P-name for a forcing notion (i.e. $\emptyset_P \Vdash \underset{\sim}{Q}$ is a forcing notion.) We define $P * \underset{\sim}{Q}$ to consist of all pairs $(p, \underset{\sim}{q})$ such that $p \in P$ and $p \Vdash_P \underset{\sim}{q} \in \underset{\sim}{Q}$, ordered by

Iterated Forcing

Two Step Iteration Suppose that P is a forcing notion and $\underset{\sim}{Q}$ is a P-name for a forcing notion (i.e. $\emptyset_P \Vdash \underset{\sim}{Q}$ is a forcing notion.) We define $P * \underset{\sim}{Q}$ to consist of all pairs $(p, \underset{\sim}{q})$ such that $p \in P$ and $p \Vdash_P \underset{\sim}{q} \in \underset{\sim}{Q}$, ordered by $(p_0, \underset{\sim}{q}) \le (p_1, \underset{\sim}{q}_1)$ iff $p_0 \le p_1$ and $p_1 \Vdash \underset{\sim}{q}_0 \le \underset{\sim}{q}_1$.

Iterated Forcing

Two Step Iteration Suppose that P is a forcing notion and $\underset{\sim}{Q}$ is a P-name for a forcing notion (i.e. $\emptyset_P \Vdash \underset{\sim}{Q}$ is a forcing notion.) We define $P * \underset{\sim}{Q}$ to consist of all pairs $(p, \underset{\sim}{q})$ such that $p \in P$ and $p \Vdash_P \underset{\sim}{q} \in \underset{\sim}{Q}$, ordered by $(p_0, \underset{\sim}{q}) \leq (p_1, \underset{\sim}{q}_1)$ iff $p_0 \leq p_1$ and $p_1 \Vdash \underset{\sim}{q}_0 \leq \underset{\sim}{q}_1$.

Finite Support Iteration A finite support iteration of forcing $\langle P_\alpha, \underset{\sim}{Q}_\beta : \alpha \leq \alpha^*, \beta < \alpha^* \rangle$ is defined by induction.

Iterated Forcing

Two Step Iteration Suppose that P is a forcing notion and $\underset{\sim}{Q}$ is a P-name for a forcing notion (i.e. $\emptyset_P \Vdash \underset{\sim}{Q}$ is a forcing notion.) We define $P * \underset{\sim}{Q}$ to consist of all pairs $(p, \underset{\sim}{q})$ such that $p \in P$ and $p \Vdash_P \underset{\sim}{q} \in \underset{\sim}{Q}$, ordered by $(p_0, \underset{\sim}{q}) \le (p_1, \underset{\sim}{q}_1)$ iff $p_0 \le p_1$ and $p_1 \Vdash \underset{\sim}{q}_0 \le \underset{\sim}{q}_1$.

Finite Support Iteration A finite support iteration of forcing $\langle P_\alpha, \underset{\sim}{Q}_\beta : \alpha \le \alpha^*, \beta < \alpha^* \rangle$ is defined by induction.

The Inductive hypotheses are that each $P_\alpha \in V$ and that $\underset{\sim}{Q}_\alpha$ is a P_α-name for a forcing notion.

Iterated Forcing

Two Step Iteration Suppose that P is a forcing notion and $\underset{\sim}{Q}$ is a P-name for a forcing notion (i.e. $\emptyset_P \Vdash \underset{\sim}{Q}$ is a forcing notion.) We define $P * \underset{\sim}{Q}$ to consist of all pairs $(p, \underset{\sim}{q})$ such that $p \in P$ and $p \Vdash_P \underset{\sim}{q} \in \underset{\sim}{Q}$, ordered by $(p_0, \underset{\sim}{q}) \leq (p_1, \underset{\sim}{q}_1)$ iff $p_0 \leq p_1$ and $p_1 \Vdash \underset{\sim}{q}_0 \leq \underset{\sim}{q}_1$.

Finite Support Iteration A finite support iteration of forcing $\langle P_\alpha, \underset{\sim}{Q}_\beta : \alpha \leq \alpha^*, \beta < \alpha^* \rangle$ is defined by induction.

The Inductive hypotheses are that each $P_\alpha \in V$ and that $\underset{\sim}{Q}_\alpha$ is a P_α-name for a forcing notion.

$P_0 = \{\emptyset\}$.

Iterated Forcing

Two Step Iteration Suppose that P is a forcing notion and $\underset{\sim}{Q}$ is a P-name for a forcing notion (i.e. $\emptyset_P \Vdash \underset{\sim}{Q}$ is a forcing notion.) We define $P * \underset{\sim}{Q}$ to consist of all pairs $(p, \underset{\sim}{q})$ such that $p \in P$ and $p \Vdash_P \underset{\sim}{q} \in \underset{\sim}{Q}$, ordered by $(p_0, \underset{\sim}{q}) \le (p_1, \underset{\sim}{q}_1)$ iff $p_0 \le p_1$ and $p_1 \Vdash \underset{\sim}{q}_0 \le \underset{\sim}{q}_1$.

Finite Support Iteration A finite support iteration of forcing $\langle P_\alpha, \underset{\sim}{Q}_\beta : \alpha \le \alpha^*, \beta < \alpha^* \rangle$ is defined by induction.

The Inductive hypotheses are that each $P_\alpha \in V$ and that $\underset{\sim}{Q}_\alpha$ is a P_α-name for a forcing notion.

$P_0 = \{\emptyset\}$. Given P_α we have that $P_{\alpha+1} = P_\alpha * \underset{\sim}{Q}_\alpha$.

Iterated Forcing

Two Step Iteration Suppose that P is a forcing notion and $\underset{\sim}{Q}$ is a P-name for a forcing notion (i.e. $\emptyset_P \Vdash \underset{\sim}{Q}$ is a forcing notion.) We define $P * \underset{\sim}{Q}$ to consist of all pairs $(p, \underset{\sim}{q})$ such that $p \in P$ and $p \Vdash_P \underset{\sim}{q} \in \underset{\sim}{Q}$, ordered by $(p_0, \underset{\sim}{q}) \leq (p_1, \underset{\sim}{q}_1)$ iff $p_0 \leq p_1$ and $p_1 \Vdash \underset{\sim}{q}_0 \leq \underset{\sim}{q}_1$.

Finite Support Iteration A finite support iteration of forcing $\langle P_\alpha, \underset{\sim}{Q}_\beta : \alpha \leq \alpha^*, \beta < \alpha^* \rangle$ is defined by induction.

The Inductive hypotheses are that each $P_\alpha \in V$ and that $\underset{\sim}{Q}_\alpha$ is a P_α-name for a forcing notion.

$P_0 = \{\emptyset\}$. Given P_α we have that $P_{\alpha+1} = P_\alpha * \underset{\sim}{Q}_\alpha$. If $\alpha > 0$ is a limit ordinal then,

Iterated Forcing

Two Step Iteration Suppose that P is a forcing notion and $\underset{\sim}{Q}$ is a P-name for a forcing notion (i.e. $\emptyset_P \Vdash \underset{\sim}{Q}$ is a forcing notion.) We define $P * \underset{\sim}{Q}$ to consist of all pairs $(p, \underset{\sim}{q})$ such that $p \in P$ and $p \Vdash_P \underset{\sim}{q} \in \underset{\sim}{Q}$, ordered by $(p_0, \underset{\sim}{q}) \leq (p_1, \underset{\sim}{q}_1)$ iff $p_0 \leq p_1$ and $p_1 \Vdash \underset{\sim}{q}_0 \leq \underset{\sim}{q}_1$.

Finite Support Iteration A finite support iteration of forcing $\langle P_\alpha, \underset{\sim}{Q}_\beta : \alpha \leq \alpha^*, \beta < \alpha^* \rangle$ is defined by induction.

The Inductive hypotheses are that each $P_\alpha \in V$ and that $\underset{\sim}{Q}_\alpha$ is a P_α-name for a forcing notion.

$P_0 = \{\emptyset\}$. Given P_α we have that $P_{\alpha+1} = P_\alpha * \underset{\sim}{Q}_\alpha$. If $\alpha > 0$ is a limit ordinal then,

$$P_\alpha = \text{ all functions } p \text{ such that } :$$

- $\mathrm{dom}(p) = \alpha$,

Iterated Forcing

Two Step Iteration Suppose that P is a forcing notion and $\underset{\sim}{Q}$ is a P-name for a forcing notion (i.e. $\emptyset_P \Vdash \underset{\sim}{Q}$ is a forcing notion.) We define $P * \underset{\sim}{Q}$ to consist of all pairs $(p, \underset{\sim}{q})$ such that $p \in P$ and $p \Vdash_P \underset{\sim}{q} \in \underset{\sim}{Q}$, ordered by $(p_0, \underset{\sim}{q}) \le (p_1, \underset{\sim}{q}_1)$ iff $p_0 \le p_1$ and $p_1 \Vdash \underset{\sim}{q}_0 \le \underset{\sim}{q}_1$.

Finite Support Iteration A finite support iteration of forcing $\langle P_\alpha, \underset{\sim}{Q}_\beta : \alpha \le \alpha^*, \beta < \alpha^* \rangle$ is defined by induction.

The Inductive hypotheses are that each $P_\alpha \in V$ and that $\underset{\sim}{Q}_\alpha$ is a P_α-name for a forcing notion.

$P_0 = \{\emptyset\}$. Given P_α we have that $P_{\alpha+1} = P_\alpha * \underset{\sim}{Q}_\alpha$. If $\alpha > 0$ is a limit ordinal then,

$$P_\alpha = \text{ all functions } p \text{ such that } :$$

- $\mathrm{dom}(p) = \alpha$,
- $\forall \gamma < \alpha, p \restriction \gamma \Vdash p(\gamma) \in \underset{\sim}{Q}_\gamma$,

Iterated Forcing

Two Step Iteration Suppose that P is a forcing notion and $\underset{\sim}{Q}$ is a P-name for a forcing notion (i.e. $\emptyset_P \Vdash \underset{\sim}{Q}$ is a forcing notion.) We define $P * \underset{\sim}{Q}$ to consist of all pairs $(p, \underset{\sim}{q})$ such that $p \in P$ and $p \Vdash_P \underset{\sim}{q} \in \underset{\sim}{Q}$, ordered by $(p_0, \underset{\sim}{q}) \leq (p_1, \underset{\sim}{q}_1)$ iff $p_0 \leq p_1$ and $p_1 \Vdash \underset{\sim}{q}_0 \leq \underset{\sim}{q}_1$.

Finite Support Iteration A finite support iteration of forcing $\langle P_\alpha, \underset{\sim}{Q}_\beta : \alpha \leq \alpha^*, \beta < \alpha^* \rangle$ is defined by induction. The Inductive hypotheses are that each $P_\alpha \in V$ and that $\underset{\sim}{Q}_\alpha$ is a P_α-name for a forcing notion.

$P_0 = \{\emptyset\}$. Given P_α we have that $P_{\alpha+1} = P_\alpha * \underset{\sim}{Q}_\alpha$. If $\alpha > 0$ is a limit ordinal then,

$$P_\alpha = \text{ all functions } p \text{ such that } :$$

- $\mathrm{dom}(p) = \alpha$,
- $\forall \gamma < \alpha$, $p \restriction \gamma \Vdash p(\gamma) \in \underset{\sim}{Q}_\gamma$,
- $\{\gamma < \alpha : \neg(p \restriction \gamma \Vdash p(\gamma) = \underset{\sim}{\emptyset}_\gamma)\}$ is finite.

Killing Souslin trees

Solovay and Tennenbaum 1970:

Killing Souslin trees

Solovay and Tennenbaum 1970: It is (relatively) consistent that there are no Souslin trees.

Killing Souslin trees

Solovay and Tennenbaum 1970: It is (relatively) consistent that there are no Souslin trees.

Start with say $V = L$ and do an iteration of length ω_2 in which each individual step $\underset{\sim}{Q}_\alpha$ comes equipped with a P_α-name $\underset{\sim}{R}_\alpha$ for a Souslin tree and it achieves that $\underset{\sim}{R}_\alpha$ is no longer a Souslin tree in $P_{\alpha+1}$.

Killing Souslin trees

Solovay and Tennenbaum 1970: It is (relatively) consistent that there are no Souslin trees.

Start with say $V = L$ and do an iteration of length ω_2 in which each individual step $\underset{\sim}{Q}_\alpha$ comes equipped with a P_α-name $\underset{\sim}{R}_\alpha$ for a Souslin tree and it achieves that $\underset{\sim}{R}_\alpha$ is no longer a Souslin tree in $P_{\alpha+1}$.

It is possible to arrange the "bookkeeping" so that $\langle \underset{\sim}{R}_\alpha : \alpha < \omega_2 \rangle$ goes over all relevant names, i.e. any Souslin tree in the extension by P_{ω_2} is named at some stage as $\underset{\sim}{R}_\alpha$.

Killing Souslin trees

Solovay and Tennenbaum 1970: It is (relatively) consistent that there are no Souslin trees.

Start with say $V = L$ and do an iteration of length ω_2 in which each individual step $\underset{\sim}{Q}_\alpha$ comes equipped with a P_α-name $\underset{\sim}{R}_\alpha$ for a Souslin tree and it achieves that $\underset{\sim}{R}_\alpha$ is no longer a Souslin tree in $P_{\alpha+1}$.

It is possible to arrange the "bookkeeping" so that $\langle \underset{\sim}{R}_\alpha : \alpha < \omega_2 \rangle$ goes over all relevant names, i.e. any Souslin tree in the extension by P_{ω_2} is named at some stage as $\underset{\sim}{R}_\alpha$.

For all said so far can use as reference ""Proper and Improper Forcing" by Shelah and for iterated forcing also the article "Iterated Forcing" by Jim Baumgartner.

But how to preserve cardinals?

ccc is a property of forcing that guarantees that the forcing preserves all cardinals.

But how to preserve cardinals?

ccc is a property of forcing that guarantees that the forcing preserves all cardinals.

Theorem

(Martin, Solovay 1970) An iteration of ccc forcing with finite supports is ccc.

But how to preserve cardinals?

ccc is a property of forcing that guarantees that the forcing preserves all cardinals.

Theorem

(Martin, Solovay 1970) An iteration of ccc forcing with finite supports is ccc.

****

But how to preserve cardinals?

ccc is a property of forcing that guarantees that the forcing preserves all cardinals.

Theorem

(Martin, Solovay 1970) An iteration of ccc forcing with finite supports is ccc.

******************* Do you find iterated forcing a bit hairy?

But how to preserve cardinals?

ccc is a property of forcing that guarantees that the forcing preserves all cardinals.

Theorem

(Martin, Solovay 1970) An iteration of ccc forcing with finite supports is ccc.

******************* Do you find iterated forcing a bit hairy?

We have got a solution for you ....

Forcing Axioms or forcing without forcing

Martin's Axiom (MA) : For every ccc forcing notion $\mathbb{P}$

Forcing Axioms or forcing without forcing

Martin's Axiom (MA) : For every ccc forcing notion $\mathbb{P}$ and every family $\mathfrak{F}$ of $< \mathfrak{c}$ many dense sets in $\mathbb{P}$,

Forcing Axioms or forcing without forcing

Martin's Axiom (MA) : For every ccc forcing notion $\mathbb{P}$ and every family $\mathfrak{F}$ of $< \mathfrak{c}$ many dense sets in $\mathbb{P}$, there is a filter in $\mathbb{P}$ which intersects all elements of $\mathfrak{F}$.

Forcing Axioms or forcing without forcing

Martin's Axiom (MA) : For every ccc forcing notion $\mathbb{P}$ and every family $\mathfrak{F}$ of $< \mathfrak{c}$ many dense sets in $\mathbb{P}$, there is a filter in $\mathbb{P}$ which intersects all elements of $\mathfrak{F}$.

Under CH, MA is true.

Forcing Axioms or forcing without forcing

Martin's Axiom (MA) : For every ccc forcing notion $\mathbb{P}$ and every family $\mathfrak{F}$ of $< \mathfrak{c}$ many dense sets in $\mathbb{P}$, there is a filter in $\mathbb{P}$ which intersects all elements of $\mathfrak{F}$.

Under CH, MA is true. It is consistent to have $MA + \neg CH$,

Forcing Axioms or forcing without forcing

Martin's Axiom (MA) : For every ccc forcing notion $\mathbb{P}$ and every family $\mathfrak{F}$ of $< \mathfrak{c}$ many dense sets in $\mathbb{P}$, there is a filter in $\mathbb{P}$ which intersects all elements of $\mathfrak{F}$.

Under CH, MA is true. It is consistent to have $MA + \neg CH$, as one can prove using an iteration of ccc forcing.

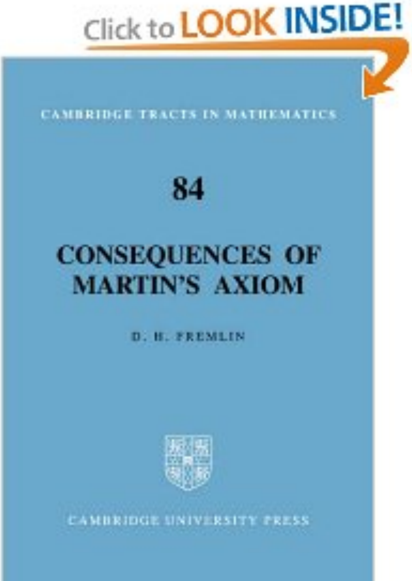
Forcing Axioms or forcing without forcing

Martin's Axiom (MA) : For every ccc forcing notion $\mathbb{P}$ and every family $\mathfrak{F}$ of $< \mathfrak{c}$ many dense sets in $\mathbb{P}$, there is a filter in $\mathbb{P}$ which intersects all elements of $\mathfrak{F}$.

Under CH, MA is true. It is consistent to have $MA + \neg CH$, as one can prove using an iteration of ccc forcing.

Using $MA + \neg CH$ set theorists and non-set theorists have proved a variety of consistency results, mostly about ω_1.

31yk4LxV7rL._BO2,204,203,200_PIsitb-sticker-arrow-click,TopRight,35,-76_AA300_SH20_OU01



In the absence of ccc

There are nice forcings that preserve cardinals,

In the absence of ccc

There are nice forcings that preserve cardinals, yet they are not ccc.

In the absence of ccc

There are nice forcings that preserve cardinals, yet they are not ccc. For example, adding a Sacks real.

In the absence of ccc

There are nice forcings that preserve cardinals, yet they are not ccc. For example, adding a Sacks real.To iterate those we need a more involved notion.

In the absence of ccc

There are nice forcings that preserve cardinals, yet they are not ccc. For example, adding a Sacks real.To iterate those we need a more involved notion.

Properness is a property that guarantees that ω_1 is preserved.

In the absence of ccc

There are nice forcings that preserve cardinals, yet they are not ccc. For example, adding a Sacks real.To iterate those we need a more involved notion.

Properness is a property that guarantees that ω_1 is preserved. The definition of properness is a strike of genius, by Shelah.

In the absence of ccc

There are nice forcings that preserve cardinals, yet they are not ccc. For example, adding a Sacks real.To iterate those we need a more involved notion.

Properness is a property that guarantees that ω_1 is preserved. The definition of properness is a strike of genius, by Shelah.

Theorem

(Shelah 1980) Properness is preserved under countable support iterations.

In the absence of ccc

There are nice forcings that preserve cardinals, yet they are not ccc. For example, adding a Sacks real.To iterate those we need a more involved notion.

Properness is a property that guarantees that ω_1 is preserved. The definition of properness is a strike of genius, by Shelah.

Theorem

(Shelah 1980) Properness is preserved under countable support iterations.

PFA The same as MA but with "ccc" replaced by proper and "$< \mathfrak{c}$" with ω_1 dense sets.

In the absence of ccc

There are nice forcings that preserve cardinals, yet they are not ccc. For example, adding a Sacks real.To iterate those we need a more involved notion.

Properness is a property that guarantees that ω_1 is preserved. The definition of properness is a strike of genius, by Shelah.

Theorem

(Shelah 1980) Properness is preserved under countable support iterations.

PFA The same as MA but with "ccc" replaced by proper and "$< \mathfrak{c}$" with ω_1 dense sets.
[Why ω_1 ? Todorčević and Veličković proved that PFA implies $\mathfrak{c} = \omega_2$.]

In the absence of ccc

There are nice forcings that preserve cardinals, yet they are not ccc. For example, adding a Sacks real.To iterate those we need a more involved notion.

Properness is a property that guarantees that ω_1 is preserved. The definition of properness is a strike of genius, by Shelah.

Theorem

(Shelah 1980) Properness is preserved under countable support iterations.

PFA The same as MA but with "ccc" replaced by proper and "$< \mathfrak{c}$" with ω_1 dense sets.
[Why ω_1 ? Todorčević and Veličković proved that PFA implies $\mathfrak{c} = \omega_2$.]

Theorem

(Baumgartner 1984) Modulo a supercompact cardinal, PFA is consistent.

Properness and generic conditions

Definition

Suppose that $M \prec (H(\chi), \in)$ with $\mathbb{P} \in M$.

Properness and generic conditions

Definition

Suppose that $M \prec (H(\chi), \in)$ with $\mathbb{P} \in M$. A condition q is $(\mathbb{P}, M)$*-generic* if for every maximal antichain $A \in M$, the antichain $A \cap M$ is maximal above q (i.e. for conditions r with $q \leq r$).

Properness and generic conditions

Definition

Suppose that $M \prec (H(\chi), \in)$ with $\mathbb{P} \in M$. A condition q is $(\mathbb{P}, M)$-*generic* if for every maximal antichain $A \in M$, the antichain $A \cap M$ is maximal above q (i.e. for conditions r with $q \leq r$).

$\mathbb{P}$ is *proper* if for every countable $M \prec (H(\chi), \in)$ with $\mathbb{P} \in M$, for every $p \in \mathbb{P} \cap M$, there is $q \geq p$ which is $(\mathbb{P}, M)$-generic.

Properness and generic conditions

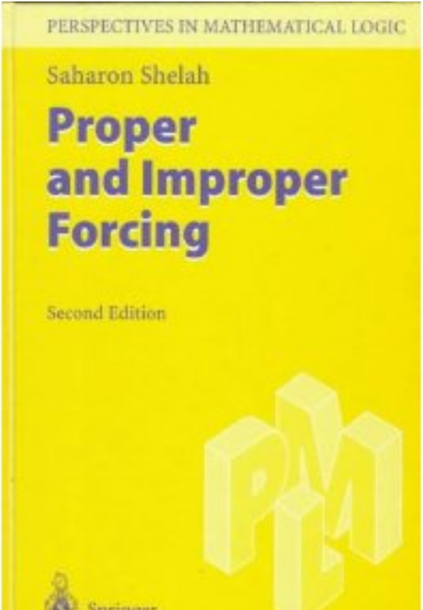
Definition

Suppose that $M \prec (H(\chi), \in)$ with $\mathbb{P} \in M$. A condition q is $(\mathbb{P}, M)$*-generic* if for every maximal antichain $A \in M$, the antichain $A \cap M$ is maximal above q (i.e. for conditions r with $q \leq r$).

$\mathbb{P}$ is *proper* if for every countable $M \prec (H(\chi), \in)$ with $\mathbb{P} \in M$, for every $p \in \mathbb{P} \cap M$, there is $q \geq p$ which is $(\mathbb{P}, M)$-generic.

A posteriori, this is a lot like master conditions in large cardinal forcing.
Note ccc implies proper.

416NGXCBCNL._SL500_AA300_.jpg 300 × 300 pixels



Some facts about proper forcing

Proper forcing cannot be iterated with finite supports.

Some facts about proper forcing

Proper forcing cannot be iterated with finite supports.

The iteration theorem for countable supports of proper forcing is much more involved than the one for finite supports of ccc forcing.

Some facts about proper forcing

Proper forcing cannot be iterated with finite supports.

The iteration theorem for countable supports of proper forcing is much more involved than the one for finite supports of ccc forcing.

Proper forcing of size $\aleph_1$ or with strong $\aleph_2$-cc properties

Some facts about proper forcing

Proper forcing cannot be iterated with finite supports.

The iteration theorem for countable supports of proper forcing is much more involved than the one for finite supports of ccc forcing.

Proper forcing of size $\aleph_1$ or with strong $\aleph_2$-cc properties preserves cardinals, cofinalities and stationary subsets of ω_1.

Some facts about proper forcing

Proper forcing cannot be iterated with finite supports.

The iteration theorem for countable supports of proper forcing is much more involved than the one for finite supports of ccc forcing.

Proper forcing of size $\aleph_1$ or with strong $\aleph_2$-cc properties preserves cardinals, cofinalities and stationary subsets of ω_1.

The natural applications of proper forcing are therefore on ω_1,

Some facts about proper forcing

Proper forcing cannot be iterated with finite supports.

The iteration theorem for countable supports of proper forcing is much more involved than the one for finite supports of ccc forcing.

Proper forcing of size $\aleph_1$ or with strong $\aleph_2$-cc properties preserves cardinals, cofinalities and stationary subsets of ω_1.

The natural applications of proper forcing are therefore on ω_1, with countable conditions

Some facts about proper forcing

Proper forcing cannot be iterated with finite supports.

The iteration theorem for countable supports of proper forcing is much more involved than the one for finite supports of ccc forcing.

Proper forcing of size $\aleph_1$ or with strong $\aleph_2$-cc properties preserves cardinals, cofinalities and stationary subsets of ω_1.

The natural applications of proper forcing are therefore on ω_1, with countable conditions as everybody knows :-)

Some facts about proper forcing

Proper forcing cannot be iterated with finite supports.

The iteration theorem for countable supports of proper forcing is much more involved than the one for finite supports of ccc forcing.

Proper forcing of size $\aleph_1$ or with strong $\aleph_2$-cc properties preserves cardinals, cofinalities and stationary subsets of ω_1.

The natural applications of proper forcing are therefore on ω_1, with countable conditions as everybody knows :-)

We shall see.

Above ω_1

It turns out that naive analogues of MA *do not* work with ω_2.

Above ω_1

It turns out that naive analogues of MA *do not* work with ω_2. For example, the iteration of κ^+-cc $< \kappa$-closed (every increasing sequence of length $< \kappa$ has an upper bound) forcing does not have to be κ^+-cc (various examples, a known one by Mitchell, involving Souslin trees).

Above ω_1

It turns out that naive analogues of MA *do not* work with ω_2. For example, the iteration of κ^+-cc $< \kappa$-closed (every increasing sequence of length $< \kappa$ has an upper bound) forcing does not have to be κ^+-cc (various examples, a known one by Mitchell, involving Souslin trees).

To generalize MA to κ^+ with $\kappa^{<\kappa} = \kappa$

Above ω_1

It turns out that naive analogues of MA *do not* work with ω_2. For example, the iteration of κ^+-cc $< \kappa$-closed (every increasing sequence of length $< \kappa$ has an upper bound) forcing does not have to be κ^+-cc (various examples, a known one by Mitchell, involving Souslin trees).

To generalize MA to κ^+ with $\kappa^{<\kappa} = \kappa$ we need to assume a strong form of $\kappa^+ - cc$ (Baumgartner, Shelah 1984)

Above ω_1

It turns out that naive analogues of MA *do not* work with ω_2. For example, the iteration of κ^+-cc $< \kappa$-closed (every increasing sequence of length $< \kappa$ has an upper bound) forcing does not have to be κ^+-cc (various examples, a known one by Mitchell, involving Souslin trees).

To generalize MA to κ^+ with $\kappa^{<\kappa} = \kappa$ we need to assume a strong form of $\kappa^+ - cc$ (Baumgartner, Shelah 1984) $< \kappa$-directed completeness or similar and

Above ω_1

It turns out that naive analogues of MA *do not* work with ω_2. For example, the iteration of κ^+-cc $< \kappa$-closed (every increasing sequence of length $< \kappa$ has an upper bound) forcing does not have to be κ^+-cc (various examples, a known one by Mitchell, involving Souslin trees).

To generalize MA to κ^+ with $\kappa^{<\kappa} = \kappa$ we need to assume a strong form of $\kappa^+ - cc$ (Baumgartner, Shelah 1984) $< \kappa$-directed completeness or similar and some sort of "well met property" :

Above ω_1

It turns out that naive analogues of MA *do not* work with ω_2. For example, the iteration of κ^+-cc $< \kappa$-closed (every increasing sequence of length $< \kappa$ has an upper bound) forcing does not have to be κ^+-cc (various examples, a known one by Mitchell, involving Souslin trees).

To generalize MA to κ^+ with $\kappa^{<\kappa} = \kappa$ we need to assume a strong form of $\kappa^+ - cc$ (Baumgartner, Shelah 1984) $< \kappa$-directed completeness or similar and some sort of "well met property" : every two compatible conditions have a lub.

Above ω_1

It turns out that naive analogues of MA *do not* work with ω_2. For example, the iteration of κ^+-cc $< \kappa$-closed (every increasing sequence of length $< \kappa$ has an upper bound) forcing does not have to be κ^+-cc (various examples, a known one by Mitchell, involving Souslin trees).

To generalize MA to κ^+ with $\kappa^{<\kappa} = \kappa$ we need to assume a strong form of $\kappa^+ - cc$ (Baumgartner, Shelah 1984) $< \kappa$-directed completeness or similar and some sort of "well met property" : every two compatible conditions have a lub.

There is no, at least no popular, analogue of properness for ω_2.

Above ω_1

It turns out that naive analogues of MA *do not* work with ω_2. For example, the iteration of κ^+-cc $< \kappa$-closed (every increasing sequence of length $< \kappa$ has an upper bound) forcing does not have to be κ^+-cc (various examples, a known one by Mitchell, involving Souslin trees).

To generalize MA to κ^+ with $\kappa^{<\kappa} = \kappa$ we need to assume a strong form of $\kappa^+ - cc$ (Baumgartner, Shelah 1984) $< \kappa$-directed completeness or similar and some sort of "well met property" : every two compatible conditions have a lub.

There is no, at least no popular, analogue of properness for ω_2.

Solution, for adding an object *once* (no iteration) is sometimes to use finite conditions. (Baumgartner-Shelah, Koszmider, Mitchell, Todorčević, Veličković-Venturi).

Above ω_1

It turns out that naive analogues of MA *do not* work with ω_2. For example, the iteration of κ^+-cc $< \kappa$-closed (every increasing sequence of length $< \kappa$ has an upper bound) forcing does not have to be κ^+-cc (various examples, a known one by Mitchell, involving Souslin trees).

To generalize MA to κ^+ with $\kappa^{<\kappa} = \kappa$ we need to assume a strong form of $\kappa^+ - cc$ (Baumgartner, Shelah 1984) $< \kappa$-directed completeness or similar and some sort of "well met property" : every two compatible conditions have a lub.

There is no, at least no popular, analogue of properness for ω_2.

Solution, for adding an object *once* (no iteration) is sometimes to use finite conditions. (Baumgartner-Shelah, Koszmider, Mitchell, Todorčević, Veličković-Venturi). Models as side conditions.

Let us try to add a club to ω_1 using *finite* conditions, Let us try to add a club to ω_1 using *finite* conditions, so finite subsets of the club.

Let us try to add a club to ω_1 using *finite* conditions, so finite subsets of the club. No reason that the generic should be closed.

Let us try to add a club to ω_1 using *finite* conditions, so finite subsets of the club. No reason that the generic should be closed.

Baumgartner (1984) (alternative Abraham 1983) solved this:

Let us try to add a club to ω_1 using *finite* conditions, so finite subsets of the club. No reason that the generic should be closed.

Baumgartner (1984) (alternative Abraham 1983) solved this: each condition gives finitely many ordinals that will be in

Let us try to add a club to ω_1 using *finite* conditions, so finite subsets of the club. No reason that the generic should be closed.

Baumgartner (1984) (alternative Abraham 1983) solved this: each condition gives finitely many ordinals that will be in and finitely many intervals of the form $(\alpha, \alpha']$ from which we have to keep out.

Let us try to add a club to ω_1 using *finite* conditions, so finite subsets of the club. No reason that the generic should be closed.

Baumgartner (1984) (alternative Abraham 1983) solved this: each condition gives finitely many ordinals that will be in and finitely many intervals of the form $(\alpha, \alpha']$ from which we have to keep out.

If we try this with ω_2, then ω_1 will get collapsed.

Let us try to add a club to ω_1 using *finite* conditions, so finite subsets of the club. No reason that the generic should be closed.

Baumgartner (1984) (alternative Abraham 1983) solved this: each condition gives finitely many ordinals that will be in and finitely many intervals of the form $(\alpha, \alpha']$ from which we have to keep out.

If we try this with ω_2, then ω_1 will get collapsed.

Friedman (2006) and Mitchell (2009) independently found a way how to add

Let us try to add a club to ω_1 using *finite* conditions, so finite subsets of the club. No reason that the generic should be closed.

Baumgartner (1984) (alternative Abraham 1983) solved this: each condition gives finitely many ordinals that will be in and finitely many intervals of the form $(\alpha, \alpha']$ from which we have to keep out.

If we try this with ω_2, then ω_1 will get collapsed.

Friedman (2006) and Mitchell (2009) independently found a way how to add a club to ω_2 using finite conditions and proper forcing!

Let us try to add a club to ω_1 using *finite* conditions, so finite subsets of the club. No reason that the generic should be closed.

Baumgartner (1984) (alternative Abraham 1983) solved this: each condition gives finitely many ordinals that will be in and finitely many intervals of the form $(\alpha, \alpha']$ from which we have to keep out.

If we try this with ω_2, then ω_1 will get collapsed.

Friedman (2006) and Mitchell (2009) independently found a way how to add a club to ω_2 using finite conditions and proper forcing! The trick is that each conditions in addition to the ordinals and the intervals has a finite sequence of countable elementary submodels of H_θ

Let us try to add a club to ω_1 using *finite* conditions, so finite subsets of the club. No reason that the generic should be closed.

Baumgartner (1984) (alternative Abraham 1983) solved this: each condition gives finitely many ordinals that will be in and finitely many intervals of the form $(\alpha, \alpha']$ from which we have to keep out.

If we try this with ω_2, then ω_1 will get collapsed.

Friedman (2006) and Mitchell (2009) independently found a way how to add a club to ω_2 using finite conditions and proper forcing! The trick is that each conditions in addition to the ordinals and the intervals has a finite sequence of countable elementary submodels of H_θ that are used to prove that the forcing is actually proper.

Let us try to add a club to ω_1 using *finite* conditions, so finite subsets of the club. No reason that the generic should be closed.

Baumgartner (1984) (alternative Abraham 1983) solved this: each condition gives finitely many ordinals that will be in and finitely many intervals of the form $(\alpha, \alpha']$ from which we have to keep out.

If we try this with ω_2, then ω_1 will get collapsed.

Friedman (2006) and Mitchell (2009) independently found a way how to add a club to ω_2 using finite conditions and proper forcing! The trick is that each conditions in addition to the ordinals and the intervals has a finite sequence of countable elementary submodels of H_θ that are used to prove that the forcing is actually proper.

Introduced the idea of *strongly proper* forcing.

$\square_{\omega_1}$ using finite conditions and proper forcing

Recall Jensen's definition:

$\square_{\omega_1}$ using finite conditions and proper forcing

Recall Jensen's definition:

$\square_{\omega_1}$ is the statement:

$\Box_{\omega_1}$ using finite conditions and proper forcing

Recall Jensen's definition:

$\Box_{\omega_1}$ is the statement:

There is a sequence $\langle C_\alpha : \alpha \text{ limit} < \omega_2 \rangle$ such that:

- C_α is a club in α

$\square_{\omega_1}$ using finite conditions and proper forcing

Recall Jensen's definition:

$\square_{\omega_1}$ is the statement:

There is a sequence $\langle C_\alpha : \alpha \text{ limit} < \omega_2 \rangle$ such that:

- C_α is a club in α
- if α is a limit point of C_β then $C_\alpha = C_\beta \cap \alpha$

$\square_{\omega_1}$ using finite conditions and proper forcing

Recall Jensen's definition:

$\square_{\omega_1}$ is the statement:

There is a sequence $\langle C_\alpha : \alpha \text{ limit} < \omega_2 \rangle$ such that:

- C_α is a club in α
- if α is a limit point of C_β then $C_\alpha = C_\beta \cap \alpha$
- if $\mathrm{cf}(\alpha) = \omega$ then C_α is countable.

$\square_{\omega_1}$ using finite conditions and proper forcing

Recall Jensen's definition:

$\square_{\omega_1}$ is the statement:

There is a sequence $\langle C_\alpha : \alpha \text{ limit} < \omega_2 \rangle$ such that:

- C_α is a club in α
- if α is a limit point of C_β then $C_\alpha = C_\beta \cap \alpha$
- if $\mathrm{cf}(\alpha) = \omega$ then C_α is countable.

This holds in L.

$\Box_{\omega_1}$ using finite conditions and proper forcing

Recall Jensen's definition:

$\Box_{\omega_1}$ is the statement:

There is a sequence $\langle C_\alpha : \alpha \text{ limit} < \omega_2 \rangle$ such that:

- C_α is a club in α
- if α is a limit point of C_β then $C_\alpha = C_\beta \cap \alpha$
- if $\mathrm{cf}(\alpha) = \omega$ then C_α is countable.

This holds in L. Adding a square is a classical forcing question,

$\square_{\omega_1}$ using finite conditions and proper forcing

Recall Jensen's definition:

$\square_{\omega_1}$ is the statement:

There is a sequence $\langle C_\alpha : \alpha \text{ limit} < \omega_2 \rangle$ such that:

- C_α is a club in α
- if α is a limit point of C_β then $C_\alpha = C_\beta \cap \alpha$
- if $\mathrm{cf}(\alpha) = \omega$ then C_α is countable.

This holds in L. Adding a square is a classical forcing question, natural after adding a club.

$\square_{\omega_1}$ using finite conditions and proper forcing

Recall Jensen's definition:

$\square_{\omega_1}$ is the statement:

There is a sequence $\langle C_\alpha : \alpha \text{ limit} < \omega_2 \rangle$ such that:

- C_α is a club in α
- if α is a limit point of C_β then $C_\alpha = C_\beta \cap \alpha$
- if $\mathrm{cf}(\alpha) = \omega$ then C_α is countable.

This holds in L. Adding a square is a classical forcing question, natural after adding a club.

Note: to destroy a square "thread a club".

In a joint paper with Dolinar (APAL 2013), we add a square with finite conditions.

In a joint paper with Dolinar (APAL 2013), we add a square with finite conditions.

Our forcing actually adds a square on a club,

In a joint paper with Dolinar (APAL 2013), we add a square with finite conditions.

Our forcing actually adds a square on a club, a Mitchell-Friedman club.

In a joint paper with Dolinar (APAL 2013), we add a square with finite conditions.

Our forcing actually adds a square on a club, a Mitchell-Friedman club. It can be shown that if there is a square on a club, then there is a square.

In a joint paper with Dolinar (APAL 2013), we add a square with finite conditions.

Our forcing actually adds a square on a club, a Mitchell-Friedman club. It can be shown that if there is a square on a club, then there is a square.

Each condition has a finite domain $\mathcal{F}_p$,

In a joint paper with Dolinar (APAL 2013), we add a square with finite conditions.

Our forcing actually adds a square on a club, a Mitchell-Friedman club. It can be shown that if there is a square on a club, then there is a square.

Each condition has a finite domain $\mathcal{F}_p$, for each $\alpha \in \mathcal{F}_p$ we choose a club,

In a joint paper with Dolinar (APAL 2013), we add a square with finite conditions.

Our forcing actually adds a square on a club, a Mitchell-Friedman club. It can be shown that if there is a square on a club, then there is a square.

Each condition has a finite domain $\mathcal{F}_p$, for each $\alpha \in \mathcal{F}_p$ we choose a club, a finite set of intervals $\mathcal{O}_p$ that keep things out, and

In a joint paper with Dolinar (APAL 2013), we add a square with finite conditions.

Our forcing actually adds a square on a club, a Mitchell-Friedman club. It can be shown that if there is a square on a club, then there is a square.

Each condition has a finite domain $\mathcal{F}_p$, for each $\alpha \in \mathcal{F}_p$ we choose a club, a finite set of intervals $\mathcal{O}_p$ that keep things out, and a finite set of models $\mathcal{M}_p$

In a joint paper with Dolinar (APAL 2013), we add a square with finite conditions.

Our forcing actually adds a square on a club, a Mitchell-Friedman club. It can be shown that if there is a square on a club, then there is a square.

Each condition has a finite domain $\mathcal{F}_p$, for each $\alpha \in \mathcal{F}_p$ we choose a club, a finite set of intervals $\mathcal{O}_p$ that keep things out, and a finite set of models $\mathcal{M}_p$ and "safeguards" $\mathcal{S}_p$.

In a joint paper with Dolinar (APAL 2013), we add a square with finite conditions.

Our forcing actually adds a square on a club, a Mitchell-Friedman club. It can be shown that if there is a square on a club, then there is a square.

Each condition has a finite domain $\mathcal{F}_p$, for each $\alpha \in \mathcal{F}_p$ we choose a club, a finite set of intervals $\mathcal{O}_p$ that keep things out, and a finite set of models $\mathcal{M}_p$ and "safeguards" $\mathcal{S}_p$. Safeguards control the interaction between the models and the club.

In a joint paper with Dolinar (APAL 2013), we add a square with finite conditions.

Our forcing actually adds a square on a club, a Mitchell-Friedman club. It can be shown that if there is a square on a club, then there is a square.

Each condition has a finite domain $\mathcal{F}_p$, for each $\alpha \in \mathcal{F}_p$ we choose a club, a finite set of intervals $\mathcal{O}_p$ that keep things out, and a finite set of models $\mathcal{M}_p$ and "safeguards" $\mathcal{S}_p$. Safeguards control the interaction between the models and the club.

For $\alpha < \omega_2$ with $\mathrm{cf}(\alpha) = \omega_1$, let E_α denote some fixed club in α of order type ω_1.

In a joint paper with Dolinar (APAL 2013), we add a square with finite conditions.

Our forcing actually adds a square on a club, a Mitchell-Friedman club. It can be shown that if there is a square on a club, then there is a square.

Each condition has a finite domain $\mathcal{F}_p$, for each $\alpha \in \mathcal{F}_p$ we choose a club, a finite set of intervals $\mathcal{O}_p$ that keep things out, and a finite set of models $\mathcal{M}_p$ and "safeguards" $\mathcal{S}_p$. Safeguards control the interaction between the models and the club.

For $\alpha < \omega_2$ with $\mathrm{cf}(\alpha) = \omega_1$, let E_α denote some fixed club in α of order type ω_1.

Suppose that $\mathcal{M}_1, \mathcal{M}_2 \prec H_{\omega_2}$ are countable and let $\delta := \sup(M_1 \cap M_2)$.

In a joint paper with Dolinar (APAL 2013), we add a square with finite conditions.

Our forcing actually adds a square on a club, a Mitchell-Friedman club. It can be shown that if there is a square on a club, then there is a square.

Each condition has a finite domain $\mathcal{F}_p$, for each $\alpha \in \mathcal{F}_p$ we choose a club, a finite set of intervals $\mathcal{O}_p$ that keep things out, and a finite set of models $\mathcal{M}_p$ and "safeguards" $\mathcal{S}_p$. Safeguards control the interaction between the models and the club.

For $\alpha < \omega_2$ with $\mathrm{cf}(\alpha) = \omega_1$, let E_α denote some fixed club in α of order type ω_1.

Suppose that $\mathcal{M}_1, \mathcal{M}_2 \prec H_{\omega_2}$ are countable and let $\delta := \sup(M_1 \cap M_2)$. Then the set $\{\min(M_1 \setminus \lambda) \mid \lambda \in M_2,\ \delta < \lambda < \sup(M_1)\} \cup \{\min(M_1 \setminus \delta)\}$ is called the set of M_1*-fences for* M_2;

Definition

The forcing notion P is the set of conditions of the form $p := (\mathcal{F}_p, \ \mathcal{S}_p, \ \mathcal{O}_p, \ \mathcal{M}_p)$, where

(1) $\mathcal{F}_p : \mathrm{Lim}(\omega_2) \to \mathcal{P}(\omega_2)$, $|\mathcal{F}_p| < \omega$ and for all $\alpha \in \mathcal{D}_p := \mathrm{dom}(\mathcal{F}_p)$, $\mathcal{F}_p(\alpha)$ is a club $C_\alpha \subset \alpha$ whose order type is $< \omega_1$ if $\mathrm{cf}(\alpha) = \omega$ and which satisfies $C_\alpha \in \{E_\alpha \setminus \beta \mid \beta \in \mathcal{D}_p \cap \alpha\}$ if $\mathrm{cf}(\alpha) = \omega_1$;

(2) $\mathcal{S}_p \subset \mathcal{D}_p$ and $\alpha \in \mathcal{S}_p$ for every $\alpha \in \mathcal{D}_p$ with $\mathrm{cf}(\alpha) = \omega_1$;

(3) $\mathcal{M}_p$ is a finite set of sets $M \cap \omega_2$ for some ocountable $\mathcal{M}[M] \prec H_\theta$, and $\sup(M) \in \mathcal{S}_p$ for every $M \in \mathcal{M}_p$;

(4) for every $\alpha \neq \beta \in \mathcal{D}_p$, if $\mu \in \mathrm{Lim}(C_\alpha) \cap \mathrm{Lim}(C_\beta)$ then $C_\alpha \cap \mu = C_\beta \cap \mu$;

(5) if $\alpha \in \mathcal{D}_p$ and $\sigma \in \mathcal{S}_p \cap \alpha$, then $C_\alpha \cap \sigma$ is a finite set;

(6) for all $\alpha \in \mathcal{D}_p$ and $M \in \mathcal{M}_p$:

(a) if $\alpha \in M$ then $C_\alpha \in \mathcal{M}[M]$,

(b) if $\alpha \notin M$ is such that $\alpha < \sup(M)$, or if $\alpha \in M$ is such that $\sup(M \cap \alpha) < \alpha$, then $\min(M \setminus \alpha) \in \mathcal{S}_p$ and $\sup(M \cap \alpha) \in \mathcal{D}_p$[1],

[1] Note that if $\alpha \in M$ then $\sup(M \cap \alpha) < \alpha$ iff $\mathrm{cf}(\alpha) = \omega_1$.

Definition

(c) if $\alpha \notin M$, $\sup(M \cap \alpha) < \alpha < \sup(M)$ and there is no $\beta \in \mathcal{D}_p \setminus (\alpha + 1)$, such that $\alpha \in \mathrm{Lim}(C_\beta)$, then $C_\alpha \cap \sup(M \cap \alpha)$ is a finite set,
(d) if $\alpha \notin M$, $\sup(M \cap \alpha) = \alpha$ and there is no $\beta \in \mathcal{D}_p \setminus (\alpha + 1)$, such that $\alpha \in \mathrm{Lim}(C_\beta)$, then C_α is some cofinal sequence in α of length ω;
(7) $\mathcal{O}_p$ is a finite set of half open nonempty intervals $(\beta', \beta] \subset \omega_2$ such that $\mathcal{D}_p \cap \bigcup \mathcal{O}_p = \emptyset$;
(8) if $(\beta', \beta] \in \mathcal{O}_p$ and $M \in \mathcal{M}_p$ then either $(\beta', \beta] \in \mathcal{M}$ or $(\beta', \beta] \cap \mathcal{M} = \emptyset$;
(9) if $M_1, M_2 \in \mathcal{M}_p$ then they are compatible, and the M_1-fence for M_2 is a subset of $\mathcal{S}_p$.

For $p, q \in P$ define

$$p \leq q \overset{\text{def}}{\iff} \mathcal{F}_p \subset \mathcal{F}_q,\ \mathcal{S}_p \subset \mathcal{S}_q,\ \mathcal{O}_p \subset \mathcal{O}_q,\ \mathcal{M}_p \subset \mathcal{M}_q.$$

Neeman's new way of iterating

Inspired by various developments in the forcing with models as side conditions,

Neeman's new way of iterating

Inspired by various developments in the forcing with models as side conditions, Neeman (presentation 2010, preprint 2011, paper 2014) gave a completely new way to iterate:

Neeman's new way of iterating

Inspired by various developments in the forcing with models as side conditions, Neeman (presentation 2010, preprint 2011, paper 2014) gave a completely new way to iterate: PFA with finite supports.

Neeman's new way of iterating

Inspired by various developments in the forcing with models as side conditions, Neeman (presentation 2010, preprint 2011, paper 2014) gave a completely new way to iterate: PFA with finite supports.

How is this possible?

Neeman's new way of iterating

Inspired by various developments in the forcing with models as side conditions, Neeman (presentation 2010, preprint 2011, paper 2014) gave a completely new way to iterate: PFA with finite supports.

How is this possible?

The conditions in the iteration are a mixture of elementary models and conditions in a proper forcing.

Let us fix a large regular χ and consider elementary submodels of H_χ.

Let us fix a large regular χ and consider elementary submodels of H_χ. Let θ be a supercompact cardinal and F a Laver function on θ.

Let us fix a large regular χ and consider elementary submodels of H_χ. Let θ be a supercompact cardinal and F a Laver function on θ.

Definition

(Neeman) The forcing notion $\mathbb{A}$ consists of pairs (s, p) such that

Let us fix a large regular χ and consider elementary submodels of H_χ. Let θ be a supercompact cardinal and F a Laver function on θ.

Definition

(Neeman) The forcing notion $\mathbb{A}$ consists of pairs (s, p) such that

- s is a finite sequence of models which are either countable or of the form H_α THIS IS NEW

Let us fix a large regular χ and consider elementary submodels of H_χ. Let θ be a supercompact cardinal and F a Laver function on θ.

Definition

(Neeman) The forcing notion $\mathbb{A}$ consists of pairs (s, p) such that

- s is a finite sequence of models which are either countable or of the form H_α THIS IS NEW
- s is $\in$-increasing and closed under intersections

Let us fix a large regular χ and consider elementary submodels of H_χ. Let θ be a supercompact cardinal and F a Laver function on θ.

Definition

(Neeman) The forcing notion $\mathbb{A}$ consists of pairs (s, p) such that

- s is a finite sequence of models which are either countable or of the form H_α THIS IS NEW
- s is $\in$-increasing and closed under intersections
- p is a finite partial function from θ such that if $p(\alpha)$ is defined, then $\mathbb{A} \cap H(\alpha)$ forces $F(\alpha)$ to be a proper forcing

Let us fix a large regular χ and consider elementary submodels of H_χ. Let θ be a supercompact cardinal and F a Laver function on θ.

Definition

(Neeman) The forcing notion $\mathbb{A}$ consists of pairs (s, p) such that

- s is a finite sequence of models which are either countable or of the form H_α THIS IS NEW
- s is $\in$-increasing and closed under intersections
- p is a finite partial function from θ such that if $p(\alpha)$ is defined, then $\mathbb{A} \cap H(\alpha)$ forces $F(\alpha)$ to be a proper forcing and $p(\alpha)$ to be in $F(\alpha)$

Let us fix a large regular χ and consider elementary submodels of H_χ. Let θ be a supercompact cardinal and F a Laver function on θ.

Definition

(Neeman) The forcing notion $\mathbb{A}$ consists of pairs (s, p) such that

- s is a finite sequence of models which are either countable or of the form H_α THIS IS NEW
- s is $\in$-increasing and closed under intersections
- p is a finite partial function from θ such that if $p(\alpha)$ is defined, then $\mathbb{A} \cap H(\alpha)$ forces $F(\alpha)$ to be a proper forcing and $p(\alpha)$ to be in $F(\alpha)$
- for such α, if M is a countable model in s and $\alpha \in M$ then $p(\alpha)$ is an M-generic condition for $F(\alpha)$.