Iterated forcing, Part 2: CS products and halving

Martin Goldstern

Institute of Discrete Mathematics and Geometry,
Technische Universität Wien

Hejnice, Feb 4, 2016
1. Iteration

2. Products

3. Intermezzo

4. lim sup forcing

5. liminf forcing and halving
1 Iteration
2 Products
3 Intermezzo
4 lim sup forcing
5 lim inf forcing and halving
Why iterations?

Notation
Recall:
- $P_3 = P_2 * Q_2 = Q_0 * Q_1 * Q_2$.
- $G_2 \subseteq P_2$ generic over $V$, $G(2) \subseteq Q_2$ generic over $V[G_2]$.
  $G_2 * G(2) \subseteq P_2 * Q_2 = P_3$ generic over $V$.

For example: We want to find a model where $2^{\aleph_0} = \kappa = \text{non}(\mathcal{M})$, i.e., every “small” set is meager, and the smallest nonmeager set is of size $\kappa$.
So we construct an iteration $(P_\alpha, Q_\alpha : \alpha < \kappa)$ with last element $P_\kappa$, where in each stage $\alpha$ the forcing notion $Q_\alpha$ will . . .
  - . . . add a new real $\eta_\alpha$
  - . . . add a new meager set $M_\alpha$ covering all reals in $V[G_\alpha]$.
In the end, we will have (at least) $\kappa$ many reals, and every set of size $< \kappa$ will have appeared in an intermediate universe $V[G_\alpha]$ (not obvious, work a little bit), so it will be covered by the meager set $M_\alpha$ in the next universe $V[G_{\alpha+1}]$. 
More generally:
We want to force a statement of the form $\forall X \exists Y : \varphi(X, Y)$, where

- $X$ is usually a set with few elements (e.g., a small set of reals, or a small family of measure zero sets),
- and $Y$ will be an object demonstrating that $X$ is small in some other sense (e.g., a meager set covering $X$, or a new real not contained in any element of $X$)

We start by using a forcing $Q_0$, which adds an object $Y_0$ taking care of all $X \in V$.
But then we get new objects $X$, so we have to force again with $Q_1$, to get a $Y_1$ taking care of those $X$.

etc.

At the end, after $\kappa$ many steps, we (hopefully) catch our tail and have taken care of all $X$. 

Why iterations? - continued
Why not iterations?

- Finite support: can only handle ccc forcing notions.
- Finite support: always adds Cohen reals. (However, see tomorrow’s lecture)
- Countable support: CH after $\alpha + \omega_1$ steps. Cannot get $2^{\aleph_0} > \aleph_2$.
- other supports, other limits: (not in this lecture)
1. Iteration
2. Products
3. Intermezzo
4. lim sup forcing
5. lim inf forcing and halving
Countable support products

Definition
Let \((Q_i : i \in I)\) be a family of forcing notions. The countable support product \(Q = \prod_{i \in I} Q_i\) is the set of all partial functions \(p\) with finite or countable domain \(\subseteq I\) satisfying \(p(i) \in Q_i\) for all \(i\).

\(Q\) is naturally ordered by the pointwise order. Each factor \(Q_i\) is naturally embedded into \(Q\).

If \(G \subseteq Q\) is generic, then its projection \(G(i) \subseteq Q_i\) is generic for \(Q_i\) over \(V\).

The products considered in this talk will always have \(\aleph_2\)-cc.

(All \(Q_i\) will be of size \(2^{\aleph_0}\). Now use CH and a \(\Delta\)-system argument.)
Why not CS products?

Problems

• $G(i)$ is not generic over $V[G(j)]$.
  (Actually: $G(i)$ is generic over $V[G(j)]$, but only for the forcing $Q_i \in V$. Often we have a definition of $Q_i$, and we can evaluate this definition in $V[G(j)]$ yielding a name $Q'_i$; then $G(i)$ is usually not generic for $Q'_i[G(j)]$ over $V[G(j)]$).

• Not clear if the product will preserve $\aleph_1$.

Examples

• The CS product of infinitely many Cohen reals collapses $\omega_1$.

• The CS product of infinitely many unbounded reals collapses $\omega_1$.

• The product of 2 (!) proper forcing notions may collapse $\omega_1$. (ZFC example)
1. Iteration
2. Products
3. Intermezzo
4. lim sup forcing
5. liminf forcing and halving
PLAN  On the following slides I will motivate the technique of “creatures” with “halving”, which was one ingredient in a recent paper of A.Fischer-G-Kellner-Shelah. (not a new technique)

DISCLAIMER To make things more transparent, I will lie occasionally, by downplaying or ignoring important details.

WARNING Still, a lot of technical background needs to be digested.
1. Iteration
2. Products
3. Intermezzo
4. lim sup forcing
5. liminf forcing and halving
Motivation
Fix a sequence $\tilde{J} = (J_n : n \in \omega)$ of intervals of natural numbers, which are far apart and grow quickly:

$$\cdots \ll \min J_n \ll \max J_n \ll \min J_{n+1} \ll \cdots$$

We want to add a generic function $g$ where $g(n) \subseteq 2^{J_n}$ is a set of large relative measure (say, more than $1 - 1/2^n$).

The set $\{ x \in 2^\omega \mid \forall n : x|J_n \in g(n) \}$ has positive measure, so $E_g := \{ x \in 2^\omega \mid \forall \infty n : x|J_n \in g(n) \}$ has measure 1.

We want this set to avoid all ground model reals; “iterating” our forcing many times this will tend to make non(null) big.

(non(null) = the smallest size of a non-Lebesgue-null set)

We let $\text{LARGE}_n := \{ A \subseteq 2^{J_n} : |A|/|2^{J_n}| > 1 - 1/2^n \}$. 
We want to add a generic function \( g \) with \( g(n) \subseteq 2^{J_n} \) a set in
\[
\text{LARGE}_n := \{ A \subseteq 2^{J_n} : |A|/|2^{J_n}| > 1 - 1/2^n \}.
\]

Definition
Let \( Q^J \) be the set of all \( p = (k^p, s^p, \tilde{C}^p) \), where
1. \( s^p = (s^p_0, \ldots, s^p_{k^p-1}) \), \( \forall i < k^p : s_i \in \text{LARGE}_i \).
2. \( \tilde{C} = (C_n : n \geq k) ; \ \forall n : C_n \subseteq \text{LARGE}_n \).
3. \( \lim \sup_{n \to \infty} \| C_n \|_n = \infty \), where
   \[
   \| C \|_n = \log (\text{some reasonable measure of } C)/\min J_n!!.
   \]
   (Here \( x \mapsto x!! \) is some sufficiently fast growing function.)

The sets \( C_n \) are called “creatures”, their elements “possibilities”. (Namely: possibilities for fragments of the generic.) Any generic filter \( G \) defines a generic function \( g \), and the set
\[
E_g := \{ x \in 2^\omega | \forall \infty n : x|J_n \in g(n) \}
\]
has measure 1.

For every old real \( x \in 2^\omega \), the set of all conditions \( p \) satisfying “there are infinitely many \( n \) such that \( x|J_n \) avoids all \( A \in C_n^p \)” is dense (explain why!); hence \( x \in 2^\omega \setminus E_g \), a null set.
Lemma
The forcing $Q^\tilde{J}$ has “continuous reading of names”, even “rapid reading”. (=Lipschitz reading)
More explicitly: For any name $\tilde{x} \in 2^\omega$, and any condition $p$ there is a stronger condition $q$ such that:

- For all $n$, the value of $\tilde{x} \upharpoonright \max(I_n)$ will depend only on $g \upharpoonright \max(I_n)$.

Moreover, if we demand the above only for $n \geq n_0$, then we may also demand that $p$ and $q$ agree on all creatures below $n_0$.

Proof.
A fusion argument. (blackboard?)

Corollary
Let $\tilde{J}$ and $\tilde{J}'$ be “very disjoint” sequences of intervals, and let $G \times G'$ be generic for the forcing $Q^\tilde{J} \times Q^\tilde{J}'$. Then the set $2^\omega \setminus E_g$ will cover not only all reals from $V$, but also all reals from $V[G']$. Every $Q^\tilde{J}$-name $\tilde{x} \in 2^\omega$

(For the proof, we have to work a bit with the norms.)
By modifying the forcing notion $Q^\check{J}$ a little bit, we get the following stronger version:

**Theorem**

Assume GCH for simplicity, $\kappa$ uncountable and regular. Let $P = \prod_{i<\kappa} Q_i$ be a countable support product of forcing notions $Q_i$, each isomorphic to (the same) $Q^\check{J}$. Then each coordinate $i^*$ comes conceptually “after” all the other coordinates. That means:

Whenever $\check{x}$ is a $\prod_{i \neq i^*} Q_i$-name of a function in $2^\omega$, then $x$ avoids the measure 1 set $E_{g^*}$ (where $g^*$ is the generic function added by $Q_{i^*}$).

As a consequence, $\vDash_{Q} \text{non}(\text{null}) \geq \kappa$. 
Outline

1. Iteration
2. Products
3. Intermezzo
4. lim sup forcing
5. liminf forcing and halving
WARNING Everything so far was just a warm-up. The serious stuff starts now.

We start by recalling the description of the generic null set, and change it to a generic meager set.
Motivation

Fix a sequence \( \bar{J} = (J_n : n \in \omega) \) of intervals of natural numbers, which are far apart and grow quickly:

\[
\cdots \ll \min J_n \ll \max J_n \ll \min J_{n+1} \ll \cdots
\]

We want to add a generic function \( g \) where \( g(n) \subseteq 2^{J_n} \) is a set of large relative measure (say, more than \((1 - 1/2^n))\).

The set \( \{x \in 2^\omega \mid \forall n : x \upharpoonright J_n \in g(n)\} \) has positive measure, so \( E_g := \{x \in 2^\omega \mid \forall \infty n : x \upharpoonright J_n \in g(n)\} \) has measure 1.

We want this set to avoid all ground model reals; “iterating” our forcing many times this will tend to make non(null) big.

(non(null) = the smallest size of a non-Lebesgue-null set)
Motivation

Fix a sequence $\bar{I} = (I_n : n \in \omega)$ of intervals of natural numbers, which are far apart and grow quickly:

$$\cdots \ll \min I_n \ll \max I_n \ll \min I_{n+1} \ll \cdots$$

We want to add a generic function $g$, defined on $\bigcup_n I_n$. The set $R_g = \{ x \in 2^\omega | \exists \infty n : x|_{I_n} = g|_{I_n} \}$ is residual (co-meager), its complement $M_g := \{ x \in 2^\omega | \forall \infty n : x|_{I_n} \neq g|_{I_n} \}$ is meager. We want the set $M_g$ to contain all ground model reals. This means that in our forcing conditions we must have the possibility to remove $x|_{I_n}$ from almost all $C_n$. This will make fusion more difficult.
We want to add a generic function $g$ defined on $\bigcup_n I_n$.

**Definition**

Let $Q^I$ be the set of all $p = (k^p, s^p, \tilde{C}^p, \tilde{d}^p)$, where

1. $s^p = (s^p_0, \ldots, s^p_{k^p-1})$, $\forall i < k^p : s_i \in 2^{|i|}$.
2. $\tilde{C} = (C_n : n \geq k)$; $\forall n : \emptyset \neq C_n \subseteq 2^{|n|}$.
3. $d^p = (d_n : n \geq k)$, each $d_n \in \mathbb{R}^+$.
4. $\liminf_{n \to \infty} \|C_n\|_n = \infty$, where $\|C\|_n = \frac{\log(|C| - d_n)}{\min J_n!!}$.

$q \leq p$ means all the obvious things: $k$ becomes bigger, $s$ becomes longer (inside the appropriate $C_i$), the $C_i$ shrink, and $d^q_n \geq d^p_n$ for all $n \geq k^q$. 
Halving and unhalving

**Halving** = Take 50% of all our possessions (not counting those which are already hidden), and hide them in a secret stash. Logarithmically speaking, we have lost almost no money. (At most one zero, from 1000 million to 500 million)
Concretely: Halving a creature \((C_n, d_n)\) means: replace \(d_n\) by 
\[
d'_n := d_n + \frac{1}{2}(|C_n| - d_n).
\]
From \((|C_n| - d_n)\) to \((|C_n| - d'_n)\) we lose 50%, so the norm 
\[
\log(|C| - d_n)/\min J_n!!
\] changes by at most \(1/\min J_n!!\).

**Unhalving** = When you lose “all” your money, remember your secret stash and recover it. You are now almost as rich as before. (Logarithmically speaking, at most one digit less.)
Concretely: go back from \(d'_n\) to \(d_n\).
Technical lemma: If you apply unhalving to finitely many creatures of a condition \(q\), resulting in a condition \(q'\), then 
\[
q' =^* q.
\]
Continuous reading, using halving

We use the lim sup forcing $Q^\tilde{I}$ which adds a meager set. (“Wlog” we use concrete numbers, for better readability.)

**Lemma (Unhalving Lemma)**

Let $\tilde{\alpha}$ be the name of an ordinal.

Given a condition, say $p = (s = \emptyset, (C_0, d_0), (C_1, d_1), \ldots)$. Assume that $C(0)$ allows only 3 possibilities, $C(1)$ allows 10 possibilities, and all norms $\log(|C_n| - d_n)/n!!$ are bigger than 1000 for $n \geq 2$.

Then there is a condition $q \leq p$ such that

- $C_0^q = C_0^p$ and $C_1^q = C_1^p$,
- $\forall n \geq 2: \log(|C_n^q| - d_n^q)/n!! \geq 970$ (actually: $\geq 1000 - 30/2!!$)
- If there is a condition $r \leq q$, $r = (s_0, s_1, (C_2^r, d_2^r), \ldots)$ deciding $\tilde{\alpha}$, with all norms $> 0$, then already $q \land (s_0, s_1) := (s_0, s_1, (C_2^q, d_2^q), \ldots)$ decides $\tilde{\alpha}$.

This lemma, rewritten with the proper parameters, allows a fusion argument to show continuous reading for our forcing.
Proof of the unhalving lemma

Start with $p$. For each possibility $s$ of the 30 possibilities from $C(0) \times C(1)$, say the $i$-th one, do the following:

- Strengthen the condition by replacing $C(0)$ and $C(1)$ by $s$.
- ("DECISION") Can you strengthen the current version of $C(2), C(3), \ldots$ in such a way that $\tilde{\alpha}$ is (essentially) decided, but all norms are still $\geq 1000 - i$? If so, do it.
- ("HALVING") Otherwise, apply "halving" to $C(2), C(3), \ldots$.

At the end we get a condition $q$.
Assume that $r = (s_0, s_1, (C'_2, d'_2), \ldots) \leq q$ decides $\tilde{\alpha}$. What did we do when we dealt (in step $i$) with $(s_0, s_1)$?

- Decided $\tilde{\alpha}$? Good.
- Halving? Try to get a contradiction.
  Apply unhalving to all those $(C'_j, d'_j)$ with norm $< 1000$ (there are only finitely many) to get a condition $r' =^* r$. But now in $r'$ all creatures have norm $\geq 1000 - i$, so $r'$ witnesses that we were in the DECISION case.
Theorem (Fischer-G-Kellner-Shelah 2015)
Assume GCH, and let $\kappa, \lambda$ be regular uncountable. Let $(I_n : n \in \omega)$ and $(J_n : n \in \omega)$ be as above. (Fast growing sequences of intervals).
Let $Q$ be a product of $\kappa$ many copies of the "generic null" forcing $Q_J$ and $\lambda$ many copies of the "generic meager" forcing $Q_I$. (not actually true... Use common halving parameter)
Then $\models_Q "any set of size < \kappa is null, and any set of size < \lambda is meager"$.
Moreover: $\models_Q \text{non}(\text{null})=\kappa$, $\text{non}(\text{meager})=\lambda$.
Moreover: We can combine this with other forcings (e.g. making $2^{\kappa_0} = \mu$).