Back-and-forth relations on Boolean Algebras.

Kenneth Harris.
U. of Michigan

AMS Special Session
Middletown, CT
October 2008

Joint work with Antonio Montalbán (University of Chicago).

Outline

1. Low Boolean algebras
2. Back-and-forth relations
3. Invariants for back-and-forth relations

Boolean Algebras

Definition A *Boolean algebra, BA,* is a structure $\mathcal{B} = (B, \leq, 0, 1, \vee, \wedge, \neg)$, where

- $(B, \leq)$ is a partial ordering,
- 0 is the least element and 1 the greatest,
- $x \vee y$ is the least upper bound of x and y,
- $x \wedge y$ is the greatest lower bound of x and y,
- $\neg x \vee x = 1$ and $\neg x \wedge x = 0$

Boolean Algebras

Definition A *Boolean algebra, BA,* is a structure $\mathcal{B} = (B, \leq, 0, 1, \vee, \wedge, \neg)$, where

- $(B, \leq)$ is a partial ordering,
- 0 is the least element and 1 the greatest,
- $x \vee y$ is the least upper bound of x and y,
- $x \wedge y$ is the greatest lower bound of x and y,
- $\neg x \vee x = 1$ and $\neg x \wedge x = 0$

Restriction: All BAs are countable with domain ω.

A BA $\mathcal{B}$ is *X-computable* if

X can compute B and all the operations in $\mathcal{B}$.

Low Boolean Algebras

Theorem: [Downey, Jockusch 94]

Every low Boolean Algebra has a computable copy.

i.e. If X is low and $\mathcal{B}$ is X-computable, then

there is a computable BA isomorphic to $\mathcal{B}$.

Low Boolean Algebras

Theorem: [Downey, Jockusch 94]

Every low Boolean Algebra has a computable copy.

i.e. If X is low and $\mathcal{B}$ is X-computable, then

there is a computable BA isomorphic to $\mathcal{B}$.

Theorem: [Thurber 95]

Every low_2 Boolean Algebra has a computable copy.

Low Boolean Algebras

Theorem: [Downey, Jockusch 94]

Every low Boolean Algebra has a computable copy.

i.e. If X is low and $\mathcal{B}$ is X-computable, then

there is a computable BA isomorphic to $\mathcal{B}$.

Theorem: [Thurber 95]

Every low_2 Boolean Algebra has a computable copy.

Theorem: [Knight, Stob 00]

Every low_4 Boolean Algebra has a computable copy.

Low Boolean Algebras

Theorem: [Downey, Jockusch 94]

Every low Boolean Algebra has a computable copy.

i.e. If X is low and $\mathcal{B}$ is X-computable, then

there is a computable BA isomorphic to $\mathcal{B}$.

Theorem: [Thurber 95]

Every low_2 Boolean Algebra has a computable copy.

Theorem: [Knight, Stob 00]

Every low_4 Boolean Algebra has a computable copy.

Open Question:

Does every low_n Boolean Algebra have a computable copy?

Outline

Computable Infinitary languages

The *computable infinitary formulas* at finite levels over the language of Boolean algebras are as follows.

1. A Σ^c_0 or Π^c_0 formula is a finitary, quantifier-free formula.
2. A Σ^c_{n+1} formula is a c.e. disjunction $\bigvee_{i\in\omega} \exists \bar{z}_i \varphi_i$ where each φ_i is Π^c_n.
3. A Π^c_{n+1} formula is a c.e. conjunction $\bigwedge_{i\in\omega} \forall \bar{z}_i \psi_i$ where each ψ_i is Σ^c_n.

We write $\Sigma^c_n(\mathcal{B})$ for the set of Σ^c_n sentences true in the BA $\mathcal{B}$, and similarly for $\Pi^c_n(\mathcal{B})$.

Σ_n^c-definable relations

Σ^c_{n+1}*-definable relations* are the relation enumerable in n-jumps.

Theorem: [Manasse,Slaman; Ash, Knight; Chisolm]
For a relation R on a computable structure $\mathcal{B}$, the following are equivalent

1. R is definable in $\mathcal{B}$ by a Σ^c_n formula.
2. The image of R in all copies $\mathcal{A}$ of $\mathcal{B}$ is $\Sigma^0_n(\mathcal{A})$.

n-approximable Boolean algebras

The Σ^c_n*-diagram* of a BA $\mathcal{B}$ is
the set of Σ^c_n sentences with parameters true in $\mathcal{B}$.

Definition

A BA $\mathcal{B}$ is *n-approximable* if the Σ^c_{n+1}-diagram of $\mathcal{B}$ is Σ^0_{n+1}.

n-approximable Boolean algebras

The Σ^c_n*-diagram* of a BA $\mathcal{B}$ is
the set of Σ^c_n sentences with parameters true in $\mathcal{B}$.

Definition

A BA $\mathcal{B}$ is *n-approximable* if the Σ^c_{n+1}-diagram of $\mathcal{B}$ is Σ^0_{n+1}.

Note: $\mathcal{B}$ is 0-approximable $\iff$ $\mathcal{B}$ is computable.

Note: $\mathcal{B}$ is low$_n$ $\implies$ $\mathcal{B}$ is n-approximable.

n-approximable Boolean algebras

Theorem:[Downey, Jockusch 94; Thurber 95; Knight, Stob 00]
For $n = 1, 2, 3, 4$,
every n-approximable BA has an $(n-1)$-approximable copy.

n-approximable Boolean algebras

Theorem:[Downey, Jockusch 94; Thurber 95; Knight, Stob 00]
For $n = 1, 2, 3, 4$,
every n-approximable BA has an $(n-1)$-approximable copy.

Open Question:
Does every n-approximable BA have an $(n-1)$-approximable copy?

n-equivalence

Idea. $\mathcal{A}$ and $\mathcal{B}$ are n-equivalent iff $0^{(n)}$ cannot distinguish them.

Def:
Let $\mathcal{A} \leq_n \mathcal{B} \iff \Pi_n^c(\mathcal{A}) \subseteq \Pi_n^c(\mathcal{B}) \iff \Sigma_n^c(\mathcal{B}) \subseteq \Sigma_n^c(\mathcal{A})$.

We will write $\mathcal{A} \equiv_n \mathcal{B}$ iff both $\mathcal{A} \leq_n \mathcal{B}$ and $\mathcal{B} \leq_n \mathcal{A}$.

n-equivalence

Idea. $\mathcal{A}$ and $\mathcal{B}$ are *n*-equivalent iff $0^{(n)}$ cannot distinguish them.

Def:
Let $\mathcal{A} \leq_n \mathcal{B} \iff \Pi^c_n(\mathcal{A}) \subseteq \Pi^c_n(\mathcal{B}) \iff \Sigma^c_n(\mathcal{B}) \subseteq \Sigma^c_n(\mathcal{A})$.

We will write $\mathcal{A} \equiv_n \mathcal{B}$ iff both $\mathcal{A} \leq_n \mathcal{B}$ and $\mathcal{B} \leq_n \mathcal{A}$.

We study the following family of *ordered monoids*

$$(BAs/\equiv_n \; , \; \leq_n \; , \; \oplus)$$

n-equivalence

Idea. $\mathcal{A}$ and $\mathcal{B}$ are n-equivalent iff $0^{(n)}$ cannot distinguish them.

Def:
Let $\mathcal{A} \leq_n \mathcal{B} \iff \Pi_n^c(\mathcal{A}) \subseteq \Pi_n^c(\mathcal{B}) \iff \Sigma_n^c(\mathcal{B}) \subseteq \Sigma_n^c(\mathcal{A})$.

We will write $\mathcal{A} \equiv_n \mathcal{B}$ iff both $\mathcal{A} \leq_n \mathcal{B}$ and $\mathcal{B} \leq_n \mathcal{A}$.

We study the following family of *ordered monoids*

$$(BAs/\equiv_n \,,\ \leq_n \,,\ \oplus)$$

We define $T_n(\mathcal{B}) = [\mathcal{B}]$, the $\equiv_n$-equivalence class of $\mathcal{B}$,
and call $T_n(\mathcal{B})$ the *back-and-forth type* of $\mathcal{B}$.

Back-and-Forth relations

Notation: $a_1, \ldots, a_k$ is a partition of a BA $\mathcal{B}$ if
$$a_0 \vee \ldots \vee a_k = 1 \text{ and } \forall i \neq j \ (a_i \wedge a_j = 0).$$

Back-and-Forth relations

Notation: $a_1, \ldots, a_k$ is a partition of a BA $\mathcal{B}$ if

$$a_0 \vee \ldots \vee a_k = 1 \text{ and } \forall i \neq j \ (a_i \wedge a_j = 0).$$

We write $\mathcal{B} \upharpoonright a$ for the BA whose domain is $\{x \in \mathcal{B} : x \leq a\}$

Back-and-Forth relations

Notation: $a_1, \ldots, a_k$ is a partition of a BA $\mathcal{B}$ if

$$a_0 \vee \ldots \vee a_k = 1 \text{ and } \forall i \neq j \ (a_i \wedge a_j = 0).$$

We write $\mathcal{B} \upharpoonright a$ for the BA whose domain is $\{x \in \mathcal{B} : x \leq a\}$

Theorem TFAE

1. $\mathcal{A} \leq_n \mathcal{B}$.
2. Given $\mathcal{C}$ that's isomorphic to either $\mathcal{A}$ or $\mathcal{B}$, deciding whether $\mathcal{C} \cong \mathcal{A}$ is $\mathbf{\Sigma}_n^0$-hard.

Back-and-Forth relations

Notation: $a_1, \ldots, a_k$ is a partition of a BA $\mathcal{B}$ if

$$a_0 \vee \ldots \vee a_k = 1 \text{ and } \forall i \neq j \ (a_i \wedge a_j = 0).$$

We write $\mathcal{B} \upharpoonright a$ for the BA whose domain is $\{x \in \mathcal{B} : x \leq a\}$

Theorem TFAE

1. $\mathcal{A} \leq_n \mathcal{B}$.
2. Given $\mathcal{C}$ that's isomorphic to either $\mathcal{A}$ or $\mathcal{B}$, deciding whether $\mathcal{C} \cong \mathcal{A}$ is $\mathbf{\Sigma}_n^0$-hard.
3. for every partition $(b_i)_{i \leq k}$ of $\mathcal{B}$, there is a partition $(a_i)_{i \leq k}$ of $\mathcal{A}$ such that $\forall i \leq k$ $\mathcal{B} \upharpoonright b_i \leq_{n-1} \mathcal{A} \upharpoonright a_i$.

Examples of back-and-forth relations

$$\mathcal{A} \leq_0 \mathcal{B} \iff (|\mathcal{A}| = 1 \iff |\mathcal{B}| = 1)$$

$$\mathcal{A} \leq_1 \mathcal{B} \iff \mathcal{A} \leq_0 \mathcal{B} \text{ and } |\mathcal{A}| \geq |\mathcal{B}|.$$

Does $\mathcal{B}$ bound *infinitely many elements* or *one*?

$$\mathcal{A} \leq_2 \mathcal{B} \iff |\mathcal{A}| = |\mathcal{B}| \text{ and } |\mathrm{At}(\mathcal{A})| \geq |\mathrm{At}(\mathcal{B})|,$$

At($\mathcal{B}$) is the set of atoms of $\mathcal{B}$.

Does $\mathcal{B}$ bound *infinitely many atoms* or *none*?

Outline

1. Low Boolean algebras
2. Back-and-forth relations
3. Invariants for back-and-forth relations

Indecomposable Boolean Algebras

Definition

A BA $\mathcal{B}$ is *n-indecomposable* if for every partition $a_1, \ldots, a_k$ of $\mathcal{B}$, there is an $i \leq k$ such that $\mathcal{B} \equiv_n \mathcal{B} \upharpoonright a_i$.

Indecomposable Boolean Algebras

Definition

A BA $\mathcal{B}$ is *n-indecomposable* if for every partition $a_1, \ldots, a_k$ of $\mathcal{B}$, there is an $i \leq k$ such that $\mathcal{B} \equiv_n \mathcal{B} \upharpoonright a_i$.

Theorem

Every BA is a finite product of n-indecomposable BAs.

Indecomposable Boolean Algebras

Definition

A BA $\mathcal{B}$ is *n-indecomposable* if for every partition $a_1, \ldots, a_k$ of $\mathcal{B}$, there is an $i \leq k$ such that $\mathcal{B} \equiv_n \mathcal{B} \upharpoonright a_i$.

Theorem

Every BA is a finite product of n-indecomposable BAs.

There are finitely many n-indecomposable $\equiv_n$-equivalence classes.

n	1	2	3	4	5	6	...
#	2	3	5	9	27	1578	...

Exclusive and Isomorphism types

Let $\mathbf{BF}_n = \{\mathcal{B}/\equiv_n : \mathcal{B}$ is n-indecomposable$\}$.

Definition

$\alpha \in \mathbf{BF}_n$ is a *isomorphism type* if whenever $T_n(\mathcal{A}) = T_n(\mathcal{B}) = \alpha$, $\mathcal{A} \cong \mathcal{B}$.

$\alpha \in \mathbf{BF}_n$ is an *exclusive type* if whenever $T_n(\mathcal{B}) = \alpha$ and $a \in \mathcal{B}$ exactly one of $\mathcal{B} \upharpoonright a \equiv_n \mathcal{B}$ or $\mathcal{B} \upharpoonright (\neg a) \equiv_n \mathcal{B}$.

Key for list of indecomposables that follows.

- exclusive isotype
- isotype, nonexclusive
- exclusive, nonisotype

Picture - Levels 1 and 2

bf-relations for 1- and 2-indecomposable types

projection	a_0	b_0	b_1
	b_0 — b_1	c_0	c_2 — c_1

1-indecomposable types

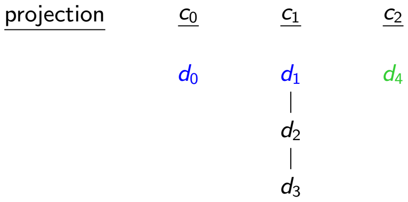
Name	Type	Example
b_0	atom	atom
b_1	infinite bounding	atomless, 1-atom

2-indecomposable types

Name	Type	Example
c_0	atom	atom
c_1	infinite	1-atom, 1-atomless
c_2	atomless	atomless

Picture - Level 3

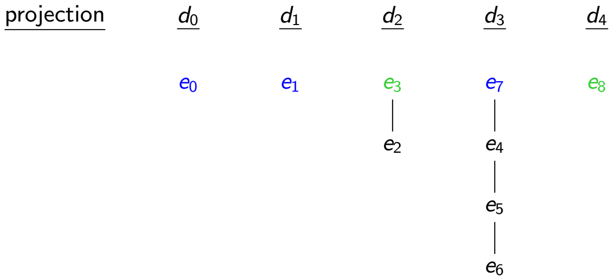
bf-relations for 3-indecomposable types



Name	Type	Example
d_0	atom	atom
d_1	1-atom	1-atom
d_2	atomic & infinite	2-atom, 1-atomless
d_3	atominf	$Int(\omega + \eta)$, $Int(\omega^2 + \eta)$
d_4	atomless	atomless

Picture - Level 4

bf-relations for 4-indecomposable types

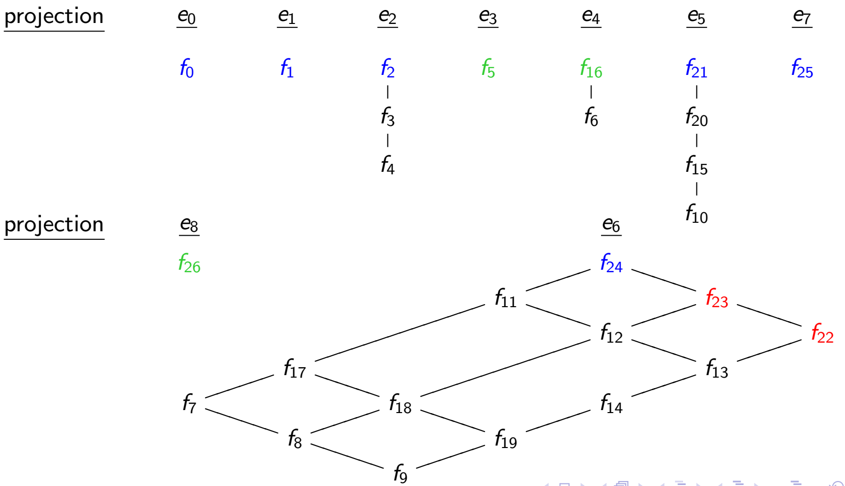


4-indecomposable isotypes

Name	Type
e_0	atom
e_0	1-atom
e_3	1-atomless
e_7	$Int(\omega+\eta)$
e_8	atomless

Picture - Level 5

bf-relations for 5-indecomposable types



Difficulties at level 5

Observation: For $n \leq 4$, and $\alpha \in \mathbf{BF}_n$,

$$\alpha \text{ is an exclusive type} \Longrightarrow \alpha \text{ is an isomorphism type.}$$

This is not true for $n = 5$.

Difficulties at level 5

Observation: For $n \leq 4$, and $\alpha \in \mathbf{BF}_n$,

α is an exclusive type $\Longrightarrow$ α is an isomorphism type.

This is not true for $n = 5$.

Observation: For $n \leq 4$, every low_n BA is isomorphic to a computable one

via a $0^{(n+2)}$*-computable isomorphism*.

Theorem (Montalbán)

There is a low_5 BA which is not isomorphic to a computable on via a $0^{(7)}$-computable isomorphism.

Invariants for back-and-forth bypes

We assign finite *invariants* to back-and-forth types as follows.

$\mathbf{BF}_0$: $* \longrightarrow a_0$, where $*$ is a symbol.

$\mathbf{BF}_{n+1}$: assign subsets of $\mathbf{BF}_n$, up to $\equiv_{n+1}$-equivalence, and subject to *realizability* by some algebra.

Invariants for back-and-forth bypes

We assign finite *invariants* to back-and-forth types as follows.

$\mathbf{BF}_0$: $* \longrightarrow a_0$, where $*$ is a symbol.

$\mathbf{BF}_{n+1}$: assign subsets of $\mathbf{BF}_n$, up to $\equiv_{n+1}$-equivalence, and subject to *realizability* by some algebra.

We assign finite sums of $\mathbf{BF}_n$ to n-types, up to $\equiv_n$-equivalence, based on how algebras *partition into n-indecomposables*.

Invariants for back-and-forth bypes

We assign finite *invariants* to back-and-forth types as follows.

$\mathbf{BF}_0$: $* \longrightarrow a_0$, where $*$ is a symbol.

$\mathbf{BF}_{n+1}$: assign subsets of $\mathbf{BF}_n$, up to $\equiv_{n+1}$-equivalence, and subject to *realizability* by some algebra.

We assign finite sums of $\mathbf{BF}_n$ to n-types, up to $\equiv_n$-equivalence, based on how algebras *partition into n-indecomposables*.

The assignments form an ordered monoid $(\mathbf{INV}_n, \leq_n, +)$ with

$$(BAs/\equiv_n, \leq_n, \oplus) \cong (\mathbf{INV}_n, \leq_n, +),$$

$(\mathbf{INV}_n, \leq_n, +)$ is a computable structure.

Boolean Algebra Predicates

Definition

For each $\alpha \in \mathbf{BF}_n$ we define a relation $\mathrm{R}_\alpha(\cdot)$ on $\mathcal{B}$:

$$\mathrm{R}_\alpha(x) \Longleftrightarrow T_n(\mathcal{B} \upharpoonright x) \geq_n \alpha.$$

Boolean Algebra Predicates

Definition

For each $\alpha \in \mathbf{BF}_n$ we define a relation $\mathrm{R}_\alpha(\cdot)$ on $\mathcal{B}$:

$$\mathrm{R}_\alpha(x) \Longleftrightarrow T_n(\mathcal{B} \restriction x) \geq_n \alpha.$$

Lemma

Each R_α for $\alpha \in \mathbf{BF}_n$ can be defined by a Π^c_n formula.

Quantifier Elimination.

Theorem

Every infinitary Σ^c_{n+1} *formula is equivalent to an infinitary* Σ_1 *formula over the predicates* R_α *for* $\alpha \in \mathbf{BF}_n$.

This formula is a $0^{(n)}$*-computable disjunction of finitary* Σ_1 *over the predicates* R_α *for* $\alpha \in \mathbf{BF}_n$.

Quantifier Elimination.

Theorem

Every infinitary Σ^c_{n+1} formula is equivalent to an infinitary Σ_1 formula over the predicates R_α for $\alpha \in \mathbf{BF}_n$.

This formula is a $0^{(n)}$-computable disjunction of finitary Σ_1 over the predicates R_α for $\alpha \in \mathbf{BF}_n$.

Theorem

Let $\mathcal{B}$ be a presentation of a Boolean algebra. TFAE.

1. *The Σ^c_{n+1}-diagram of $\mathcal{B}$ is Σ^0_{n+1};*
2. *The relations $\mathrm{R}_\alpha(\mathcal{B})$ for $\alpha \in \mathbf{BF}_n$ are computable in $0^{(n)}$.*