Complete Relations on Linear Orders and Boolean Algebras

Daniel Turetsky

University of Wisconsin - Madison

October 21, 2008

Definition

If L is a linear order, the *interval algebra on L* ($\mathcal{I}(L)$) is the boolean algebra $B \subset \mathcal{P}(L)$ generated by the half-open intervals $[a, b)$ for $a, b \in L$, $a <_L b$. Equivalently, B is the clopen sets in the lower-limit topology.

Observations

If L is computable, so is $\mathcal{I}(L)$.
If $L \cong L'$, then $\mathcal{I}(L) \cong \mathcal{I}(L')$.

Theorem

For any countable boolean algebra B, $B \cong \mathcal{I}(L)$ for some linear order L. If B is computable, we may take L to be computable and effectively obtained from an index for B.

Proof.

Inductively build L as a generating chain for B. □

Note

$\omega + 1 \ncong \omega + 2$, but $\mathcal{I}(\omega + 1) \cong \mathcal{I}(\omega + 2)$.

Observations

The pair $\langle a, b \rangle$ is a successivity in L iff $[a, b)$ is an atom (bounds only 0) in $\mathcal{I}(L)$.

The interval (a, b) is infinite in L iff $[a, b)$ is infinite (bounds infinitely many elements) in $\mathcal{I}(L)$.

The interval (a, b) is infinite dense in L iff $[a, b)$ is atomless (bounds no atoms) in $\mathcal{I}(L)$.

Theorem (Downey-Lempp-Wu)

A computable linear order L is isomorphic to a computable linear order L' such that $Succ(L')$ (the set of successivities in L') is Turing complete iff L has infinitely many successivities.

Theorem (Downey-Lempp-Wu)

A computable linear order L is isomorphic to a computable linear order L' such that $Succ(L')$ (the set of successivities in L') is Turing complete iff L has infinitely many successivities.

Corollary (?)

A computable boolean algebra A is isomorphic to a computable boolean algebra B such that $Atom(B)$ (the set of atoms in B) is Turing complete iff A has infinitely many atoms.

Lemma (Remmel)

Suppose $A \leq B$ are countable boolean algebras satisfying:

- *A contains infinitely many atoms.*
- *Every atom in A is a finite join of atoms in B.*
- *Every atom in B is below an atom in A.*
- *B is generated by A and the atoms of B.*

Then $A \cong B$.

Proof.

Back and forth. □

Proof.

Fix K Σ^0_1-complete.

Given computable boolean algebra A, will build:

- B a computable boolean algebra,
- $\iota : A \hookrightarrow B$ an embedding,
- Γ a Turing functional

satisfying:

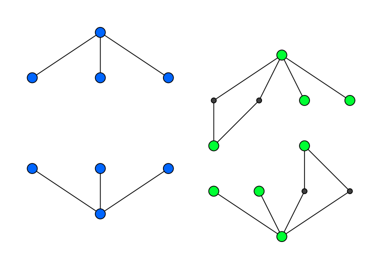
- $\Gamma^{\text{Atom}(B)} = K$
- $\iota(A) \subseteq B$ satisfies the hypothesis of the lemma.

Proof.

We use a priority argument on a tree with a Δ^0_2 true path, attempting to meet the following requirements:

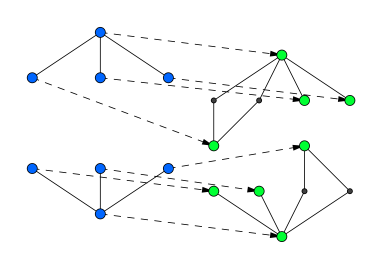
$$D_i : \ \iota(a_i) \text{ is well-defined.}$$
$$S_i : \ \iota^{-1}(b_i) \text{ is well-defined.}$$
$$R_i : \ \Gamma^{\mathrm{Atom}(B)}(i) = K(i).$$

Proof.



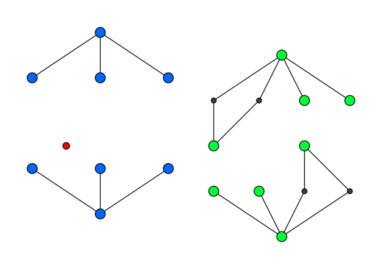
A strategy will inherit a partial embedding ι^-.

Proof.



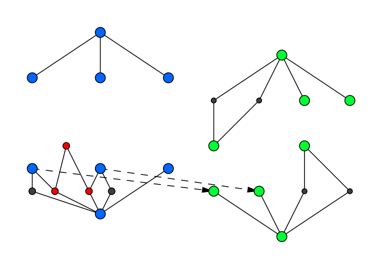
A strategy will inherit a partial embedding ι^-.

Proof.



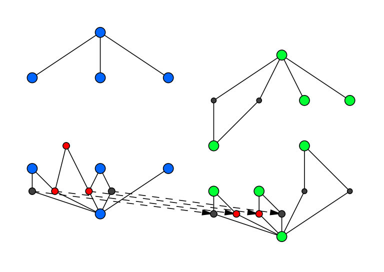
Strategy for D_i: Decompose $a_i = e_1 \vee e_2 \vee \cdots \vee e_n$ with each e_j below an atom in domain(ι^-).

Proof.



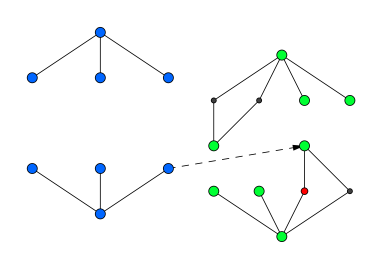
Strategy for D_i: Decompose $a_i = e_1 \vee e_2 \vee \cdots \vee e_n$ with each e_j below an atom in domain(ι^-).

Proof.



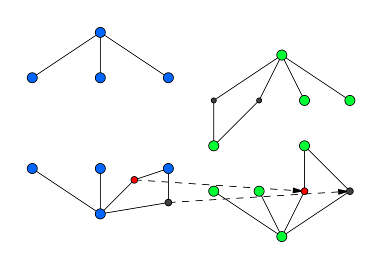
Create new elements f_j in B and extend ι by $\iota(e_j) = f_j$.

Proof.



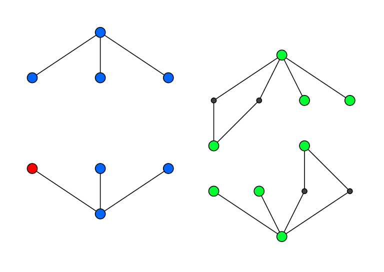
Strategy for S_i: Decompose $b_i = f_1 \vee f_2 \vee \cdots \vee f_n$ with each f_j below an atom in domain(ι^-).

Proof.



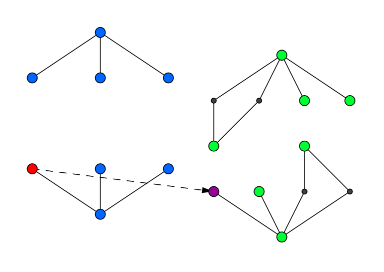
Wait for e_j to appear in A in the appropriate location and extend ι by $\iota(e_j) = f_j$.

Proof.



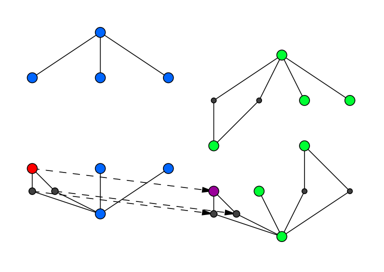
Strategy for R_i: Choose x the Gödel least element which we currently believe is an atom, and which has not been previously chosen by some R-strategy.

Proof.



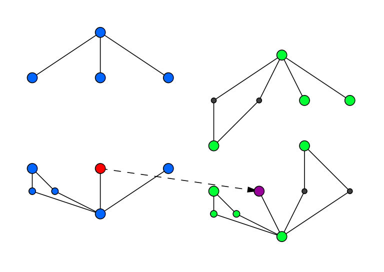
Choose a $y \leq \iota^{-}(x)$ which is currently an atom in B. Define $\Gamma^{\text{atom}(y)}(i)$ to be our current guess for $K(i)$.

Proof.



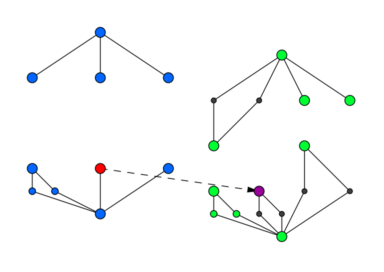
Maybe x (and thus y) stops being an atom.

Proof.



So choose a new x and y.

Proof.



If $K_s(i)$ changes, split y. Then choose a new x and y. □

Theorem

A computable boolean algebra A is isomorphic to a computable boolean algebra B such that $Atom(B)$ (the set of atoms in B) is Turing complete iff A has infinitely many atoms.

Theorem

A computable boolean algebra A is isomorphic to a computable boolean algebra B such that $Atom(B)$ (the set of atoms in B) is Turing complete iff A has infinitely many atoms.

Theorem (T)

A computable boolean algebra A is isomorphic to a computable boolean algebra B such that $Infinite(B)$ (the set of infinite elements in B) is Turing complete for Δ_3^0 iff A is not a finite join of atoms, 1-atoms and atomless elements.

Theorem (T)

A computable linear order L is isomorphic to a computable linear order L' such that $Dense(L')$ (the set of dense intervals in L') is Turing complete for Δ^0_3 iff $\mathcal{I}(L)$ has no maximal atomless element.

Corollary

A computable boolean algebra A is isomorphic to a computable boolean algebra B such that $Atomless(B)$ (the set of atomless elements in B) is Turing complete for Δ^0_3 iff A has no maximal atomless element.

Theorem (Downey-Moses)

There is a computable linear order L such that for every computable copy L', $Succ(L')$ is Turing complete.

Theorem (T)

There is a computable linear order L such that for every computable copy L', $Infinite(L')$ is Turing complete for Δ^0_3.

Theorem (T)

There is a computable linear order L such that for every computable copy L', $Dense(L')$ is Turing complete for Δ^0_3.

Theorem (Downey)

Every computable boolean algebra A is isomorphic to some computable boolean algebra B such that Atom(B) is not Turing complete.

Theorem (T)

Every computable boolean algebra A is isomorphic to some computable boolean algebra B such that Infinite(B) is not Turing complete for Δ^0_3.

Theorem (T)

Every computable boolean algebra A is isomorphic to some computable boolean algebra B such that Atomless(B) is not Turing complete for Δ^0_3.

Lemma (T)

If A is a computable boolean algebra, then A is isomorphic to some Δ^0_2-computable boolean algebra B such that $Atom(B)$ is Δ^0_2-computable, and $Atomless(B) \oplus 0'$ is not Turing complete for Δ^0_3. Skip Proof

Lemma (Downey-Jockusch)

If B is a Δ^0_2-computable boolean algebra such that $Atom(B)$ is also Δ^0_2-computable, then B is isomorphic to a computable boolean algebra via a Δ^0_2-computable isomorphism.

Proof.

Work in the presence of a Δ^0_2 oracle. In particular, we can tell when an element of A is an atom.

Fix X which is Σ^0_2 but not Δ^0_2.

We use a priority argument on a tree with a (relatively) Δ^0_2 true path, attempting to meet the following requirements:

$$D_i : \ \iota(a_i) \text{ is well-defined.}$$
$$S_i : \ \iota^{-1}(b_i) \text{ is well-defined.}$$
$$R_i : \ \Gamma_i^{\mathrm{Atomless}(B) \oplus 0'} \neq X.$$

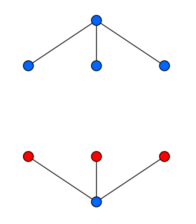
Proof.

Strategies for D_i and S_i are basically the same.

When a_i is an atom, D_i labels $\iota(a_i)$ as an atom.

When S_i puts a non-atom below an atom, split the non-atom into two elements, and label them both as atoms.

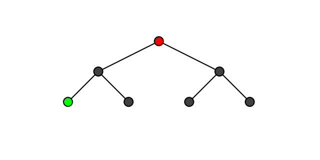
Proof.



For R_i, we consider the atoms of domain(ι^-).

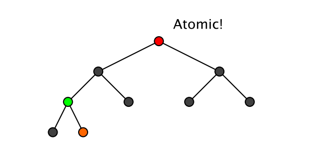
We make guesses as to whether or not these are atomic (above no atomless element), and if they are above infinitely many atoms.

Proof.



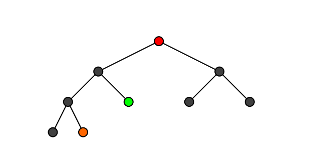
We guess that x is atomic every time an atom appears below the Gödel least element beneath x which does not already have an atom beneath it.

Proof.



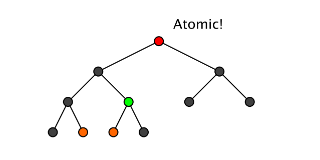
We guess that x is atomic every time an atom appears below the Gödel least element beneath x which does not already have an atom beneath it.

Proof.



We guess that x is atomic every time an atom appears below the Gödel least element beneath x which does not already have an atom beneath it.

Proof.



We guess that x is atomic every time an atom appears below the Gödel least element beneath x which does not already have an atom beneath it.

Proof.

We arrange our guesses as:

not atomic $<$ atomic

We make a tuple of guesses (e.g. (atomic, not atomic, atomic)) for the atoms of domain(ι^-).

We extend the ordering to a partial ordering on guesses.

The correct result will be greatest amongst those results guessed infinitely often.

Proof.

We search for a computation $\Gamma_i^{\sigma \oplus 0'}(0)$ which is plausible for our current guess.

Proof.

We search for a computation $\Gamma_i^{\sigma \oplus 0'}(0)$ which is plausible for our current guess.

If a computation puts an atomless element below an atomic element, it is not plausible.

If a computation puts a non-atomless element beneath an element with no atoms, it is not plausible.

Proof.

We search for a computation $\Gamma_i^{\sigma \oplus 0'}(0)$ which is plausible for our current guess.

If a computation puts an atomless element below an atomic element, it is not plausible.

If a computation puts a non-atomless element beneath an element with no atoms, it is not plausible.

If a computation is plausible for guess g, then it is plausible for all guesses $g' < g$.

Proof.

Act to preserve the first plausible computation found.

Proof.

Act to preserve the first plausible computation found.

If $\Gamma_i^{\sigma\oplus 0'}(0)$ differs from our current guess for $X(0)$, pause R_i.

Proof.

Act to preserve the first plausible computation found.

If $\Gamma_i^{\sigma\oplus 0'}(0)$ differs from our current guess for $X(0)$, pause R_i.

Otherwise, move on to searching for a computation $\Gamma_i^{\tau\oplus 0'}(1)$ with $\sigma \subseteq \tau$.

Proof.

If for some n no plausible computation is preserved for the correct result, then $\Gamma_i^{\text{Atomless}(B)\oplus 0'}(n)$ does not converge.

If for some n a plausible computation is preserved for the correct result with $\Gamma_i^{\sigma\oplus 0'}(n) \neq X(n)$, then $\Gamma_i^{\text{Atomless}(B)\oplus 0'} \neq X$.

If for every n a plausible computation is preserved for the correct result with $\Gamma_i^{\sigma\oplus 0'}(n) = X(n)$, then X is Δ_2^0-computable, a contradiction.

Thus $\Gamma_i^{\text{Atomless}(B)\oplus 0'} \neq X$. □

Theorem (T)

Every computable boolean algebra A is isomorphic to some computable boolean algebra B such that $Atomless(B)$ is not Turing complete for Δ_3^0.

Lemma (T)

If A is a computable boolean algebra, then A is isomorphic to some Δ_2^0-computable boolean algebra B such that $Atom(B)$ is Δ_2^0-computable, and $Atomless(B) \oplus 0'$ is not Turing complete for Δ_3^0.

Lemma (Downey-Jockusch)

If B is a Δ_2^0-computable boolean algebra such that $Atom(B)$ is also Δ_2^0-computable, then B is isomorphic to a computable boolean algebra C via a Δ_2^0-computable isomorphism.

Lemma (T)

If A is a computable boolean algebra, then A is isomorphic to some Δ^0_2-computable boolean algebra B such that Atom(B) is Δ^0_2-computable, and Atomless(B) $\oplus\, 0'$ is not Turing complete for Δ^0_3.

Lemma (Downey-Jockusch)

If B is a Δ^0_2-computable boolean algebra such that Atom(B) is also Δ^0_2-computable, then B is isomorphic to a computable boolean algebra C via a Δ^0_2-computable isomorphism.

Lemma (T)

If A is a computable boolean algebra, then A is isomorphic to some Δ^0_2-computable boolean algebra B such that Atom(B) is Δ^0_2-computable, and Atomless(B) $\oplus$ $0'$ is not Turing complete for Δ^0_3.

Lemma (Downey-Jockusch)

If B is a Δ^0_2-computable boolean algebra such that Atom(B) is also Δ^0_2-computable, then B is isomorphic to a computable boolean algebra C via a Δ^0_2-computable isomorphism.

Proof.

Atomless(B) $\oplus$ $0'$ computes Atomless(C). □

$L \cong L'$ with $\mathrm{Succ}(L')$ complete $\iff$ $\mathrm{Succ}(L)$ infinite.

$A \cong B$ with $\mathrm{Atom}(B)$ complete $\iff$ $\mathrm{Atom}(A)$ infinite.

$L \cong L'$ with $\mathrm{Dense}(L')$ complete $\iff$ No maximal atomless in $\mathcal{I}(L)$.

$A \cong B$ with $\mathrm{Atomless}(B)$ complete $\iff$ No maximal atomless in A.

$A \cong B$ with $\mathrm{Infinite}(B)$ complete $\iff$ A not a finite join of atoms, 1-atoms and atomless.

Definition

Call a linear order "almost discrete" if every element (except possibly the first element) with an immediate successor also has an immediate predecessor, and every element (except possibly the last element) with an immediate predecessor also has an immediate successor.

Theorem (T)

An almost discrete computable linear order L is isomorphic to a computable linear order L' such that Infinite(L') (the set of infinite intervals in L') is Turing complete for Δ^0_3 iff $\mathcal{I}(L)$ is not a finite join of atoms, 1-atoms and atomless.