Π^0_1 問題の解答不可能性の次数構造は稠密か？

木原 貴行
東北大学理学研究科数学専攻 修士課程 2 年

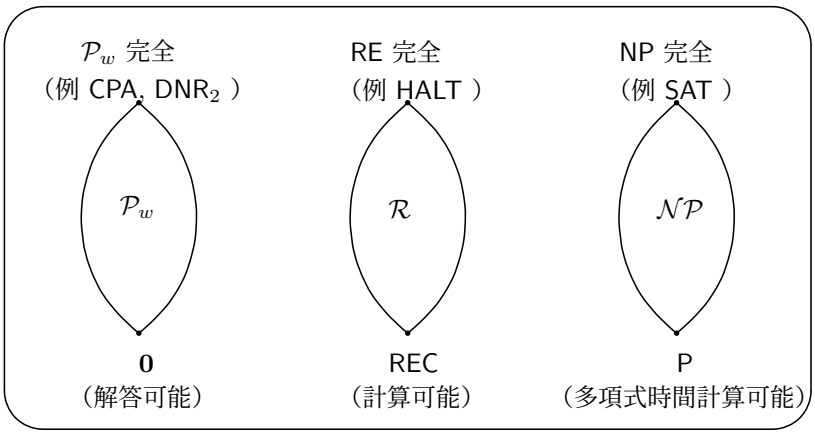
2008 年 9 月 20 日
（ラムダ計算と論理の晩夏セミナー）

概要

問題 φ が与えられたとき， $\varphi(X)$ を満たす解を一つでも具体的に得られるとき， φ に**解答可能**であるという．問題全体を相対的解答可能性による同値類で割った，**解答不可能性の次数構造**を考えよう．特に，ここでは Π^0_1 問題に対する解答不可能性の次数構造 $\mathcal{P}_w$ を対象とする．この $\mathcal{P}_w$ がどのような構造を持っているかの調査は最近になって始まった．$\mathcal{P}_w$ の構造の研究に関する基本的結果は S. Binns (2003) による $\mathcal{P}_w$ の $\exists(\leq, \wedge, \vee, 0, 1)$-理論の決定可能性である．したがって，次に問題とすべきは $\forall\exists$-理論の決定方法であろう．そこへの到達の大きな壁として，稠密性問題がある．$\mathcal{P}_w$ の稠密性は単純な $\forall\exists$-文であるが，これの真偽は現在もまだ知られていない．

1 前置き

次数構造の研究は，再帰理論の主題である．古くから調べられている次数構造としては，様々な重要なクラスを相対的計算可能性 $\leq_T$ による同値類で割った順序構造（上半束構造）がある．その典型的な例は，計算不可能性の次数構造 $\mathcal{D}$, あるいはその部分構造である Δ^0_2 次数構造 $\mathcal{D}(\leq \mathbf{0}')$, r.e. 次数構造 $\mathcal{R}$ などであろう．その他， NP 問題を相対的多項式時間計算可能性で割った NP 次数構造 $\mathcal{NP}$ は，計算量理論の観点からも重要である．今回の主題となる Π^0_1 多数問題の解答不可能性の次数構造 $\mathcal{P}_w$ は， $\mathcal{D}$ や $\mathcal{D}(\leq \mathbf{0}')$ よりもむしろ $\mathcal{R}$ や $\mathcal{NP}$ との類似が多い．$\mathcal{R}$ と $\mathcal{NP}$ が共に完全問題をもつ（ RE 完全問題には停止問題 HALT などがあり， NP 完全問題には充足可能性問題 SAT などがある）のと同様に， $\mathcal{P}_w$ も理論の完全化問題 CPA や部分関数の全域拡張問題などの多くの完全問題をもつ．



ある次数構造において，最初に問題となるのは，最小次数 $\mathbf{0}$ と最大次数 $\mathbf{1}$ は異なるものかという問いであろう．良く知られているように， NP 次数構造においては， $\mathbf{0}=\mathbf{1}$ かどうかは未解決である．(P vs. NP 問題.）しかし， RE 次数構造 $\mathcal{R}$，あるいは， Π^0_1 多数問題の次数構造 $\mathcal{P}_w$ では， $\mathbf{0}\neq\mathbf{1}$ は早い段階で証明された．次数構造において， $\mathbf{0}\neq\mathbf{1}$ が分かったら，次の問題は，中間次数は存在するかどうかという問いである．つまり， $\mathbf{0}<\mathbf{a}<\mathbf{1}$ を満たす次数 $\mathbf{a}$ の存在についての問いで， RE 次数構造においては，1944 年のポストの問題として知られる．

ポストの問題 [1944]

中間 RE 次数は存在するか？　つまり，

$$\mathcal{R}\models(\exists\mathbf{a})\,(\,\mathbf{0}<\mathbf{a}<\mathbf{1}\,),$$

かどうかを決定せよ．

ポストの問題の肯定的解，すなわち中間次数をもつ r.e. 集合の存在は，フリードバーグ [1957] とムチニク [1956] によって独立に発見された．その後，次数の理論において，単純な条件 φ について「φ を満たす次数たちは存在するか」という類の問題について数多くの研究がなされた．つまり，次数構造の $\exists$-理論 $\mathcal{R}\models\exists\vec{x}\varphi(\vec{x})$ についての研究のスタートである．この研究は次第に進んでいき， $\mathcal{R}$ の $\exists$-文の真偽を一般的に決定するアルゴリズムの探求へと興味は移っていく．つまり， $\mathcal{R}$ の $\exists$-理論の決定可能性の問題，ひいては「 r.e. 集合の計算不可能性の度合いの構造のうち，少なくとも $\exists$-理論について，我々は全てを知ることができるか？」という問いに繋がる．決定可能性に関する最初の結果はサックス [1963] による $\mathcal{R}$ の $\exists(\leq,\vee)$-理論の決定可能性である．

このようにして，次第に，多くの研究者が，次数構造の数理論理学的性質に関する興味を抱き始めた．その最たる例が，シェーンフィールドとサックスの予想である．

シェーンフィールド予想 [1965]

L_0 を $\{\leq,\vee,0,1\}$ を言語とする有限上半束の性質， L_1 を $\{\leq,\vee,0,1\}$-束として L_0 の拡張であるような無矛盾な性質であるとする．このとき，

$$\mathcal{R}\models(\forall\vec{\mathbf{a}})(\exists\vec{\mathbf{b}})\,(\,L_0(\vec{\mathbf{a}})\to L_1(\vec{\mathbf{a}},\vec{\mathbf{b}})\,).$$

特に， RE 次数構造 $\mathcal{R}$ は $\aleph_0$-範疇的である．

つまり，シェーンフィールドは， $\mathcal{R}$ の $\forall\exists(\leq,\vee,0,1)$-理論の興味深い部分クラスである「 $0,1$ を持つ有限上半束の拡張の埋め込み問題」について，その問題が束として無矛盾ならば $\mathcal{R}$ で必ず真であると予想した．

サックス予想 [1966]

RE 次数構造 $\mathcal{R}$ の一階理論 $\mathrm{Th}(\mathcal{R})$ は決定可能である．

しかし，シェーンフィールド予想は，すぐにラクラン [1966] による $\mathcal{R}$ 内に極小対が存在することの証明によって否定され，また，サックス予想も後にハーリントンとシェラー [1982] による $\mathcal{R}$ に Δ^0_2 半順序がコード可能であることの証明によって否定されることとなった．

定理 1.1 (ラクラン,1966) シェーンフィールド予想は偽である．特に，次のような反例が存在する．

$$\mathcal{R} \models (\exists \mathbf{a_0}, \mathbf{a_1})(\forall \mathbf{b})\ (\ \mathbf{b} < \mathbf{a_0}, \mathbf{a_1} \rightarrow \mathbf{b} = \mathbf{0}\ ).$$

定理 1.2 (ハーリントン-シェラー,1982) サックス予想は偽である．つまり， $\mathcal{R}$ の一階理論 $\mathrm{Th}(\mathcal{R})$ は決定不可能である．

特にハーリントンとシェラーの定理より，次数構造 $\mathcal{R}$ の一階の性質について，全てを知ることはできない．ならば， $\mathcal{R}$ の構造についてどこまで知ることができるだろうか？ サックスの定理により，少なくとも $\exists(\leq, \vee)$-理論までなら全てを知ることができる．ならば， $\forall\exists$-理論はどうだろう．さらに $\forall\exists\forall$-理論はどうだろう．全てを知ることができる（つまり，決定可能な）断片と，部分的にしか理解できない（つまり，決定不可能な）断片の境界は一体どこなのだろう？

この疑問への探求が，次数構造の一階理論の断片についての決定可能性の研究である．

さて，本文では， $\mathcal{R}$ のような計算不可能性の度合いの構造ではなく，そのある種の一般化である，解答不可能性の度合いの構造についての決定可能性を探る．これは，「絶対に具体的には解けない（多数）問題」がどれくらい解けないかの度合い（解答不可能性の度合い）の構造を調べる研究である．特に，解答不可能性の度合いの構造について，その一階の性質（または断片）の全てを知ることが可能かどうかという問題（決定可能性）が主題となる．

この研究に関する主要な結果は，ビンズ [2003] による Π^0_1 （多数）問題の解答不可能性の次数構造 $\mathcal{P}_w$ および $\mathcal{P}_s$ についての $\exists(\leq, \vee, \wedge, 0, 1)$-理論の決定可能性である．つまり， Π^0_1 問題の解答不可能性の次数構造について，単純な存在型の文については全てを知る方法が存在する．

すると， $\mathcal{R}$ の場合と同様に，次の問題のうち一つは $\forall\exists$-理論の決定可能性であろう．しかし，これに関して，最も簡単な部類の $\forall\exists$-文である稠密性の問題

$$\mathcal{P}_w \models (\forall \mathbf{a_0}, \mathbf{a_1})(\exists \mathbf{b})\ (\ \mathbf{a_0} < \mathbf{a_1} \rightarrow \mathbf{a_0} < \mathbf{b} < \mathbf{a_1}\ ).$$

の真偽すら未だ決定できていない段階である．まず，このような具体的な $\forall\exists$-文の真偽から少しずつ決定していく方法を見つけることが， $\mathcal{P}_w$ の構造についての研究の課題である．

また，もう一つの重要な問題として， $\mathcal{P}_w$ の一階理論（あるいはその断片）の決定可能性の問題である．$\mathrm{Th}(\mathcal{P}_w)$ はおそらく決定不可能であるが，算術をコードする等の何らかの方法により，それを実際に証明しなければならない．しかし，現状ではまだ $\mathcal{P}_w$ の次数構造についての情報が少なすぎる．このため， $\mathcal{P}_w$ の細かい性質について調べ上げていくことも， $\mathcal{P}_w$ の構造についての研究の課題となるであろう．

2 予備知識

2.1 計算不可能性とその次数

本文中では，集合とその特性関数を同一視する．つまり，集合 A に対して， $A(x)$ と書いた場合， $x \in A$ ならば 1 を $x \notin A$ ならば 0 を返す関数のこととする．

本文中において，**アルゴリズム** (*algorithm*) あるいは**プログラム** (*program*) と言った場合，通常想像するようなアルゴリズムやプログラム（例： C 言語， Pascal 言語など）の命令に関数 `Oracle` の呼び出し命令を付け加えたものを指す．集合 $A \in 2^\omega$ と自然数 x を引数とするプログラム $\Phi(A;x)$ の計算中に `y:=Oracle(z)` という命令が現れた場合，変数 y に $A(z)$ の値を代入する．

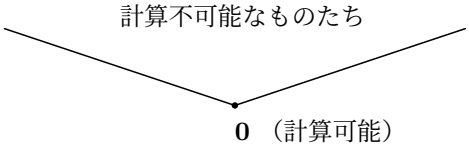
アルゴリズム Φ にオラクル A と値 x を入力した計算 $\Phi(A;x)$ が停止して y を出力したとき，$\Phi(A;x)\downarrow = y$ と書く．逆に，無限ループに陥るなどして計算が停止しない場合， $\Phi(A;x)\uparrow$ と書く．

以後， $(\forall x)\,\Phi(A;x)\downarrow = B(x)$ であるときは， $\Phi(A) = B$ と略記する．また， $\Phi(;x)$ を単に $\Phi(x)$ と書く．恒等アルゴリズム $\mathsf{Id} : (A,x) \mapsto A(x)$ による計算 $\mathsf{Id}(A;x)$ を単に $(A;x)$ と書く．また，アルゴリズム（プログラム） Ψ の記述を 2 進表記に直して，それを自然数とみなしたときにその値が e であった場合，そのアルゴリズム Ψ を Φ_e と書く．

定義 2.1 (計算可能性) 集合 B が**計算可能** (*computable, recursive*) とは， $\Phi = B$ となるアルゴリズム Φ が存在するときを指す．また，集合 B が集合 A から**相対的に計算可能** (*computable relative to* A) とは，$\Phi(A) = B$ となるアルゴリズム Φ が存在するときを指す．集合 B が集合 A から相対的に計算可能であるとき， $B \leq_T A$ と書く．

定義 2.2 (チューリング次数) $A \leq_T B$ かつ $B \leq_T A$ であるとき， $A \equiv_T B$ と書く．$\mathcal{P}(\mathbb{N})$ を同値関係 $\leq_T$ によって割った構造 $\mathcal{P}(\mathbb{N})/\equiv_T$ を $\mathcal{D}$ と書き，**計算不可能性の次数構造** (*degrees of unsolvability*) と呼ぶ．各 $\mathbf{a} \in \mathcal{D}$ を**計算不可能性の次数**または**チューリング次数** (*Turing degree*) と呼ぶ．また，計算可能な集合たちの次数を $\mathbf{0}$ と書く．計算可能なものの次数 $\mathbf{0}$ は計算不可能性の次数構造 $\mathcal{D}$ の最小元である．

$\mathcal{D}$



$\mathbf{0}$ （計算可能）

定義 2.3 (r.e. 集合) 集合 A が *r.e.*(*recursively enumerable, computably enumerable*) とは， A の元を並べるアルゴリズムが存在することを言う．つまり，次を満たすアルゴリズム Φ が存在するときを指す．

$$x \in A \;\leftrightarrow\; (\exists s)\,\Phi(s) = x.$$

集合 A が集合 B から相対的に **r.e.** とは，次を満たすアルゴリズム Φ が存在するときを指す．

$$x \in A \;\leftrightarrow\; (\exists s)\,\Phi(B;s) = x.$$

定義 2.4 (停止問題) 停止問題 (*halting problem*) とは，与えられたアルゴリズムが停止するかどうかの判定である．つまり，停止問題 K^A とは次により定義される．

$$K^A = \{\, e \,:\, \Phi_e(A;0)\downarrow \,\}.$$

命題 2.1 停止問題 K^A は A から相対的に r.e. である．

証明 A-相対的な計算を並列に実行して，停止したものから順に並べていけばよい． □

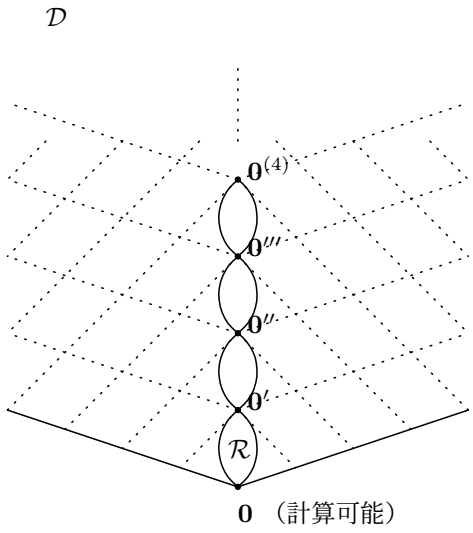
定理 2.1 (停止問題の計算不可能性) $(\forall A)\; A <_T K^A$. □

以後，K^A を A' と書き，A の**チューリングジャンプ** (*Turing jump*) と呼ぶ．同様に，次数 $\mathbf{a}$ の集合 A のチューリングジャンプ A' の次数を $\mathbf{a}'$ と書く．

定理 2.2 任意の集合 A について，A から相対的に r.e. な集合 B は A' から相対的に計算可能．

証明 $x \in B$ かどうかを確かめるには，B を A-相対的に並べるアルゴリズム Φ に対して，それを x を並べた時点で停止するというアルゴリズム Φ' に書き換え，それに対して停止問題を用いればよい． □

定義 2.5 r.e. 集合全体 RE を $\equiv_T$ で割った構造を *r.e.* **次数構造** (*r.e. degrees*) と呼び，$\mathcal{R}$ と書く．計算可能なものの次数 $\mathbf{0}$ は $\mathcal{R}$ の最小元となり，停止問題の次数 $\mathbf{0}'$ は $\mathcal{R}$ の最大元となる．



2.2 オラクル使用量

以後の議論において，計算 $\Phi(A;x)$ の値を保ったまま A を拡張する，あるいは計算 $\Gamma(B;y)$ の値を壊すために B に元を並べるという構成を行う．オラクルの変化に応じた計算結果の制御を行うために，**オラクル使用量** (*use*) に注目する．

たとえば，次のような Pascal ライクなプログラムを考えよう．

プログラム Φ

```
begin
        read(x);
        y:=Oracle(314);
        write(y);
end
```

つまり，オラクルが 314 を含むならば 1 を出力し，さもなくば 0 を出力するプログラムである．このとき，A に 314 が投げ込まれるまでは $\Phi(A;x)\downarrow=0$ である．しかし，その後，もし A に 314 が投げ込まれることがあれば，$\Phi(A;x)\downarrow=1$ となり，計算結果が崩れてしまう．しかし，逆に言えば，314 以外の元は幾ら A に投げ込んでも，計算 $\Phi(A;x)\downarrow=0$ を害しない．つまり，計算途中に呼び出したオラクル関数 Oracle の**使用** (*use*) 部分さえオラクルを変化させなければ，計算結果には何の影響も及ぼさない．これがオラクル使用量のアイデアである．

計算 $\Phi(A;x)$ の**オラクル使用量** (use) とは， $\Phi(A;x)$ の計算を実行中におけるオラクル関数の呼び出し命令 `z:=Oracle(u)` について，そのような u の最大値である．もし計算 $\Phi(A;x)$ が停止するならば，停止までに掛かる時間は有限時間であり，したがって，オラクル関数 `Oracle` の呼び出し命令も有限回しか実行できない．したがって，そのような呼び出しの最大値は必ず存在する．

命題 2.2 $\Phi(A;x)\downarrow$ とし， u を計算 $\Phi(A;x)$ のオラクル使用量とする．このとき，任意の $v > u$ について，$\Phi(A\cup\{v\};x)\downarrow = \Phi(A;x)\downarrow$. □

注意すべき点として，オラクル使用量は，プログラム Φ だけでなく，オラクルと引数の値にも依存することである．たとえば，次のようなプログラムで確かめてみよう．

プログラム Φ

```
begin
        read(x);
        i:=0; y:=0;
        while (i=0) do begin
                i:=Oracle(x), y:=y*2, x++;
        end
        write(y);
end
```

このプログラム Φ において，計算 $\Phi(A;x)$ が停止するならば，そのオラクル使用量は $\min(A\setminus\{y : y < x\})$ となる．もちろん，これは非常に単純な例であり，一般にはオラクル使用量は非常に複雑な形となる．このように，オラクル使用量は，オラクルと引数の値によっても変化するため，以後，計算 $\Phi(A;x)$ のオラクル使用量は $\varphi(A;x)$ のように書く．

定義 2.6 (オラクル使用量) 計算 $\Phi(A;x)$ の**オラクル使用量** (use) とは， $\Phi(A;x)$ の計算を実行中におけるオラクル関数の呼び出し命令 `z:=Oracle(u)` について，そのような u の最大とする．もし計算 $\Phi(A;x)$ が停止しないならば，オラクル使用量は ∞ とする．計算 $\Phi(A;x)$ のオラクル使用量を $\varphi(A;x)$ と書く．

定義 2.7 (計算可能汎関数) Φ が**計算可能汎関数**とは，次を満たす 3 つ組 $\langle\sigma,x,y\rangle$ の r.e. 集合である．

$$(\forall\langle\sigma,x,y\rangle,\langle\tau,x,z\rangle\in\Phi)\ (\ \sigma\subseteq\tau\rightarrow y=z\ ).$$

さらに，ある $\sigma\subset A$ について $\langle\sigma,x,y\rangle\in\Phi$ であるとき， $\Phi(A;x)\downarrow = y$ と書く．

命題 2.3 任意の計算可能汎関数 Φ に対して，同値なアルゴリズム Φ' が存在する．つまり，

$$(\forall A,x,y)\ (\ \Phi(A;x)\downarrow = y \leftrightarrow \Phi'(A;x)\downarrow = y\ ).$$

を満たすアルゴリズム Φ' が存在する． □

注意 後の証明において，計算可能関数 Γ の構成を行う際，「 $\Gamma(A;x)=y$ をオラクル使用量 $\gamma(A;x)=u$ により定義する」という操作を行う場合がある．これは実際には， A のステージ s での近似 $A[s]$ に対して， 3 つ組 $\langle A[s]\upharpoonright u,x,y\rangle$ を Γ に並べ，計算可能汎関数として Γ を構成していくことを意味する．

計算可能汎関数を考えることのメリットは，計算を壊すためにオラクルに何を追加すればよいか明確に分かることである．

2.3 多数問題の解答不可能性とその次数

計算不可能性の分類を**多数問題** (*mass problem*) へと拡張しよう．これまでの議論では，一つの対象 $S \subseteq \mathbb{N}$ がどれくらいの計算不可能性を持つかを問題にしていた．つまり， S がアルゴリズム Φ により生成される r.e. 集合ならば,「アルゴリズム Φ により生成される集合を求めよ」という S のみを解に持つ**単数問題**を考える．この単数問題の解答不可能性の度合いは，ちょうど S の計算不可能性の度合いに対応するであろう．この考え方を一般化し，解が 1 つしかない問題だけではなく，複数の解を持つ問題（非可算個の解を持ってもよい）をも扱おう．以後，論理式 φ とその解全体の集合 $\mathsf{P} = \{X \in \omega^\omega : \varphi(X)\}$ を同一視し，これを（多数）問題と呼ぶ.

問題 P が与えられたとき，それを解決するための難しさの度合いを導入する．もし，問題 P の解のいずれかを具体的なアルゴリズムにより得られるならば，それはとても簡単な解であると考え，問題 P を難しさ $\mathbf{0}$ だと考えよう．一方，解が存在しないような問題なら，これは最も難しい問題と考え， P の難しさは ∞ であると考える．では，その中間程度の難しさの問題をどう扱おう？

今回，問題の解答可能性は計算可能性（アルゴリズムの存在）を基準として考えている．もし，問題 P の解のどれかを見つけられるという状況下なら，必ず問題 Q の解のどれかを見つけられるとき， Q は P から相対的に解答可能な問題と考えてよいだろう．つまり， P のどの解を見つけてきても，それをオラクルとして Q の解のいずれかを計算可能であるとき， P から Q を**各点解答可能**と言う．あるいは， Q は P に**弱還元可能** (*weak reducible*) または**ムチニク還元可能** (*Muchnik reducible*) であるといい， $\mathsf{Q} \leq_w \mathsf{P}$ と書く．このとき，弱還元可能性 $\leq_w$ は，次のように定義される.

$$\mathsf{Q} \leq_w \mathsf{P} \ :\Leftrightarrow\ (\forall A \in \mathsf{P})(\exists \Phi)\ (\ \Phi(A) \in \mathsf{Q}\ ).$$

この各点解答可能性により生成される問題の難しさの度合いの構造 $\omega^\omega / \equiv_w$ を**解答不可能性の弱次数構造** (*weak degrees of difficulty*) または**ムチニク束** (*Muchnik lattice*) という.

さて，各点計算可能性 $\mathsf{Q} \leq_w \mathsf{P}$ においては， P の解から Q の解を計算する方法は P の解ごとに見つけてよかった．しかし，この操作に一様性を要求しよう．つまり，あらかじめ何か計算アルゴリズム Φ があって， P のどの解を見つけてきたとしても，アルゴリズム Φ から Q の解のいずれかを計算できる場合を考える．このとき， P から Q を**一様解答可能**と言う．あるいは， Q は P に**強還元可能** (*strong reducible*)，または**メドヴェージェフ還元可能** (*Medvedev reducible*) といい， $\mathsf{Q} \leq_s \mathsf{P}$ と書く.

$$\mathsf{Q} \leq_s \mathsf{P} \ :\Leftrightarrow\ (\exists \Phi)(\forall A \in \mathsf{P})\ (\ \Phi(A) \in \mathsf{Q}\ ).$$

この一様解答可能性により生成される問題の難しさの度合いの構造 $\omega^\omega / \equiv_s$ を**解答不可能性の強次数構造** (*strong degrees of difficulty*) または**メドヴェージェフ束** (*Medvedev lattice*) という.

さて，特に問題が Π^0_1 論理式により与えられているとき，その解全体の集合を Π^0_1 クラスと呼ぶ．つまり，問題 P が Π^0_1 クラスとは，ある Π^0_1 論理式 φ が存在して，次を満たすときを指す.

$$\mathsf{P} = \{\ X\ :\ \varphi(X)\ \}$$

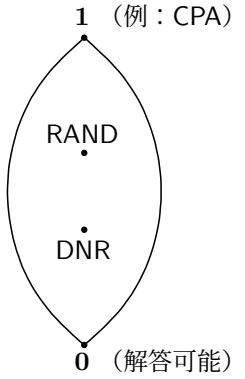
さて，問題 $\mathsf{WKL}(T, P)$ を「無限二分木 T に対する無限パス P を求めよ」というものとしよう．解 P 全体の集合は，木 T によって決まるので，単に $[T]$ と書くことにする．このとき， P が Π^0_1 クラスであること

と，ある木 T に対して， $\mathsf{P} = [T]$ となっていることは同値である．これにより， Π^0_1 クラスに関する研究は，木に関する研究であると考えることができる．

Π^0_1 問題に該当する問題として様々な例が知られている．次にリストアップするのは Π^0_1 問題の一例である．

- 計算可能無限二分木 T の無限パス P を求めよ．
- 計算可能グラフ G の 4-彩色を求めよ．
- 計算可能部分 2 値関数 f の全域 2 値拡大を求めよ．
- 任意の e について $\Phi_e(e) \neq f(e)$ なる関数 f を求めよ．(DNR)
- 無矛盾な計算可能一階理論 T の無矛盾な完全拡大を求めよ．(CT)
- Σ^0_1 条件 ϕ と ψ を分離せよ．
- マルティン=レーフ・ランダムを求めよ．(RAND)（これは実際には Π^0_2 問題であるが，解答不可能性が等しい Π^0_1 問題が存在することが知られている．）

解を持つ Π^0_1 クラスの強及び弱次数構造 $\mathcal{P}_s$ と $\mathcal{P}_w$ が本文における中心的な対象である．$\mathcal{P}_s$ と $\mathcal{P}_w$ は共に最大元 $\mathbf{1}$ と最小元 $\mathbf{0}$ を持つ．最小元 $\mathbf{0}$ は，解答可能な問題，すなわち計算可能な解を持つ問題が該当する．最大元 $\mathbf{1}$ は，例として，本質的に不完全な理論の完全拡大問題，たとえばペアノ算術 PA の完全拡大問題 CPA が該当することが知られている．また， DNR と RAND は共に $\mathcal{P}_s$ と $\mathcal{P}_w$ 両方の中間次数になることが知られており，さらにどちらの次数構造においても， $\mathbf{0} < \mathsf{DNR} < \mathsf{RAND} < \mathbf{1}$ が成立する．



3 優先論法

ポストの問題（中間 c.e. 次数は存在するか？）を解くため，フリードバーグ [1957] とムチニク [1956] は独立に**優先論法** (*priority argument, priority method*) を導入した．優先論法は，再帰理論における最も基本的なテクニックで，特に次数構造の一階理論およびその断片の決定可能性の研究においては，至る所で必要となる基本的な証明技法である．これは，本文中でも例外ではない．このため，幾つかの定理を例に用いて，優先論法の簡単な解説を行う．

3.1 フリードバーグ・ムチニクの定理

フリードバーグ・ムチニクによる最初の優先論法は，現在，**有限害** (*finite injury*) 優先論法と呼ばれる最も簡単なタイプの優先論法に属すことが知られている．

フリードバーグ・ムチニクの定理

互いに比較不可能な次数を持つような *c.e.* 集合 A と B が存在する．

要件． 各計算可能部分関数 Φ について，次の全ての要件を満たすように，計算可能な操作で元を並べていけばよい．

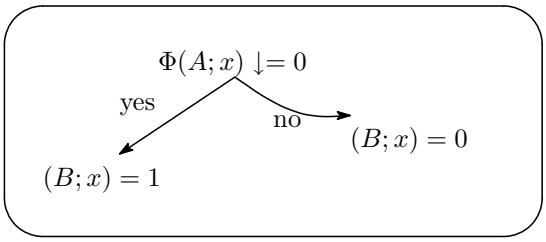
$$P_\Phi : \Phi(A) \neq B.$$
$$Q_\Phi : \Phi(B) \neq A.$$

3.1.1 証明のための戦略の案（基本戦略）

注意すべき点は，得たいものは r.e. 集合であるため，その集合を並べるアルゴリズムを見つけなければならない．構成のイメージとしては，それぞれ A と B という名前の付いた箱に延々と数を投入し続けるアルゴリズムを与えることである．ただし，箱に一度元を入れた後はもう二度と元を取り出すことはできない．

さて，要件 P_Φ を満たすためには， $\Phi(A;x) \neq B(x)$ を満たす x が存在すればよい．このために，要件毎にそのような x の候補を選んでおく．最初は B は空箱なので，基本的には $B(x) = 0$ である．このため， $\Phi(A;x)$ の計算を実行したとき，出力 0 を返さない限りには特に気にしないでよい．しかし，もし $\Phi(A;x) \downarrow = 0$ になってしまったら**要注意** (*requires attention*) である．このとき， x を B に並べるなどの対処をして， $\Phi(A) \neq B$ を保つ努力をすべきだろう．

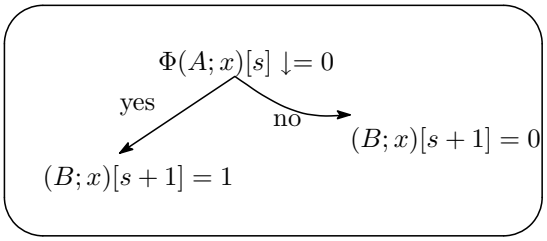
よって，要件 P_Φ を満たすためには，次の**基本単位** (*basic module*) を満たすように， P_Φ-**戦略** (*strategy*) を編み出せばよい．



この基本単位の $\Phi(A;x) \downarrow = 0$ を**指示文** (*directing sentence*) と呼び，それが偽であるときに実行する文を**活性文** (*activated sentence*)，真であるときに実行する文を**確認文** (*validated sentence*) と呼ぶ．

これは P_Φ のための戦略であるが，実際は無限個あるプログラム全てに対してこのような戦略を考える．目的の r.e. 集合の構成のステージ s で， s 番目までの要件に関する戦略を並列に実行する．もし上述の戦略を全て正常に実行することができれば，全ての要件が満たされることは明らかであろう．

しかし，もちろん $\Phi(A;x) \downarrow = 0$ になるかどうかは有限時間では分からない．この基本単位を満たす実行可能な戦略を作るためには，上述のような戦略の各文を計算可能に確認・実行できる文にまで**分解** (*decompose*) する．つまり，最終的に $\Phi(A;x) \downarrow = 0$ かどうかを調べるのではなく，現在のところ $\Phi(A;x) \downarrow = 0$ であるかを調べるのである．このために，ステージ s の段階で， s 番目までの要件に関する戦略の指示文の計算を s 時間だけ実行する．このとき，実際に s までに制限した指示文が正しいかどうかは計算可能に確認できる．もし正しければ，その戦略が現在要件を満たすための候補として選んでいる x を B に投入し，さもなくば，現在の段階では x は B に入れない．



並列に有限個動作している各 P_Φ-戦略 α は， α の稼動時間が s の段階で上図のような**導** (*derivative*) を実行する．この導戦略は計算可能に実行可能なアルゴリズムである．では，この操作を延々続けることにより，全ての要件を満たす r.e. 集合は構成されるだろうか？

いや，この戦略にはまだ問題点がある．たとえば，次のような状況を考えよう．

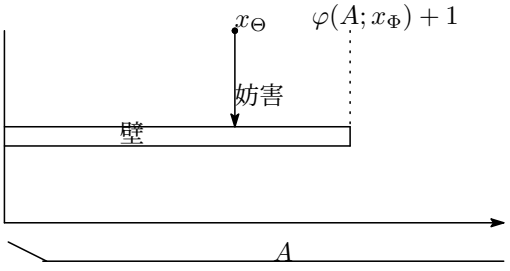
1. $\Phi(A;x_\Phi)\downarrow=0$ が分かったので， x_Φ を B に並べる．
2. $\Theta(B;x_\Theta)\downarrow=0$ が分かったので， x_Θ を A に並べる．

しかし，このとき，x_Θ を A に並べたことにより $\Phi(A)$ のオラクル部分が変化してしまうため， $\Phi(A)$ の結果が変わってしまうかもしれない．たとえば，もしかしたら $\Phi(A;x_\Phi)\downarrow=1$ となってしまうことだって十分あり得る．また，状況 1. により $B(x_\Phi)=1$ も成立しているが， B は r.e. 集合として構成しなければならないので，一度入れてしまった x_Φ はもう二度と B から取り出すことはできない．これにより，$\Phi(A;x_\Phi)\downarrow=B(x_\Phi)$ となり， x_Φ は要件 P_Φ の成立を保証せず，要件 P_Φ の成立のために x_Φ を用いるという戦略は諦めなければならない．もちろん，別の x'_Φ が P_Φ の成立を保証してくれればよいので， x'_Φ を用いて新しい戦略を行えばよい．しかし，このような方針では，戦略のやり直しを延々繰り返すだけで，一体いつこの戦略のやり直し操作が終わるのかを保証できない．たった 2 つの要件を満たすための戦略についてだけでこの調子なのだから，無限個の要件を満たさなければならない実際の構成では尚更無理がある．これでは，比較不可能な次数が存在することを保証できない．

3.1.2 証明のための戦略の案（枚挙の束縛）

可能な限り，戦略のやり直しを行いたくない．このためには，一度 $\Phi(A;x_\Phi)\downarrow=0$ になったら，この計算結果が二度と変わらないようにすればいい．つまり， $\Phi(A;x_\Phi)\downarrow$ の計算のオラクル使用量 $\varphi(A;x_\Phi)$ 以下の元は絶対に並べないことを宣言することにすればよい．そうすれば， $\Phi(A;x_\Phi)\downarrow=0$ は構成中はずっと保たれる．これは計算を保つために $y<\varphi(A;x_\Phi)$ 以下の元を A に入れることを**束縛** (*restraint*) するというアイデアである．

イメージ図としては，この戦略は，箱 A の上に大きさ $\varphi(A;x_\Phi)+1$ の**壁**を建築する．これにより，$\varphi(A;x_\Phi)$ 以下の元が上空から放り込まれたとしても，それが A の中に入ってくるのを妨害することができる．

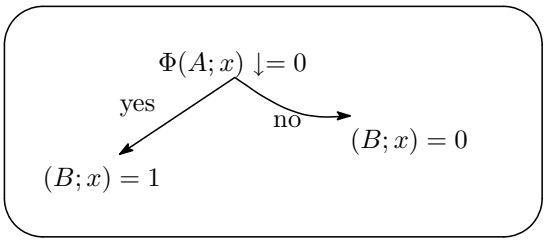


しかし，この処理によって，他の要件 Q_Θ を満たすために別の戦略が元 x_Θ を放り込もうとしても，壁に阻まれて要件 Q_Θ を満たすことはできないかもしれない．これに対処するためには，別の戦略における要件を満たすための証人 x を壁を越えるような別の証人 x' として選びなおせばいいはずである．しかし，どの戦略も，ある戦略が束縛を行うための壁を建設するたびに，新しい証人を繰り返し選びなおさなければならないかもしれない．このとき，結局どの元が戦略を満たすための最終的な証人になるか確定せず，要件は永遠に満たされないかもしれない．

3.1.3 証明のための戦略の案（優先論法）

以上の問題を解決するために，要件たちの間に**優先順位** (*priority order*) を入れる．各要件を満たすように構成を行いたいが，たとえば P_Φ を満たすような構成を行うためには，別の要件 P_Θ の充足を害してしまうかもしれない．これを防ぐために，各要件の戦略は，自分の行った戦略が害されないように，前述のように束縛を行って壁を建築する．しかし，高い優先順位を持つ要件の充足のためなら，低い優先順位を持つ要件の充足は害してしまって構わない．低い優先順位の要件を満たす戦略は，証人を壁を越えるものとして選びなおした別の戦略へと切り替えよう．これが**優先論法** (*priority argument, priority method*) という名称の由来である．

さて， P_Φ を満たすための戦略の**基本単位** (*basic module*) は，次のようなものである．



これを元に，先ほどのように実行可能な戦略を導く．まず，要件 P_Φ を満たす責任を負う α は，まず最初に P_Φ を満たすための証人 $x=\breve{\alpha}$ を現在建築されている壁 ϱ を越えるものとして選ぶ．その後，$\Phi(A;\breve{\alpha})[s]\downarrow=0$ なるステージ s を待ち，もしその時が訪れたなら， $\breve{\alpha}$ を B に並べ， $\Phi(A;\breve{\alpha})$ のオラクル使用量 $\varphi(A;\breve{\alpha})$ より大きくなるように壁を増設する．

つまり， α が行うべき戦略は次のようなものである．

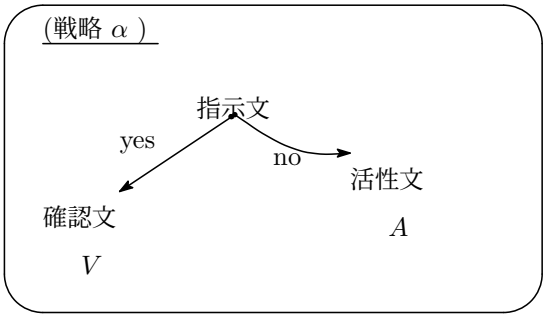
1. $\breve{\alpha}$ を現在の壁 ϱ より大きい値として割り当てる．
2. $\Phi(A;\breve{\alpha})\downarrow=0$ を待つ．
3. $\breve{\alpha}$ を B に並べ，壁 ϱ を $\Phi(A;\breve{\alpha})\downarrow=0$ の計算のオラクル使用量 $\varphi(A;\breve{\alpha})$ より大きくなるように増設する．その後，戦略の稼動を停止する．

現在の構成が作る道筋（**現在のトゥルーパス**;*current true path*）が α を通っている最中， α はこの戦略アルゴリズムを実行し続ける．構成の道筋が α から逸れている間は α は戦略アルゴリズム実行を一時停止しておき，再び構成の道筋が α を通るように戻ってきたとき，続きから戦略アルゴリズムを再開する．（ただし，フリードバーグ・ムチニク構成においては，一度，構成の道筋が α から逸れたら，二度と構成は α の下へは戻ってこない.）この戦略アルゴリズムの計算を実行中は α のアウトカムはずっと A であり，ステップ 3 に到達して計算が停止して初めて，アウトカムが V に切り替わる．

古典優先論法　古典的なフリードバーグ・ムチニクの手法による構成を考えよう．古典優先論法においては，各要件 P_Φ に対する戦略はただ 1 つである．構成のステージ s でそのような戦略を s 個並列に稼動させる．P_Φ について，現在動作している戦略の指示文が真になったとき，古典優先論法では P_Φ は**要注意** *(requires attention)* であるという．要注意であるような最も優先度の高い要件は**注目** (*receives attention*) され，確認文が実行される．さて， P_Φ についての戦略を行っている一方で，別の要件 Q_Θ についての戦略も同時に行われているかもしれない．このとき， P_Φ の戦略の実行中に上記のような操作を行った場合，これは Q_Θ の戦略にも影響を及ぼす．たとえば，P_Φ の戦略によって B に x を入れた場合，これはより優先度の低い要件 Q_Θ の戦略における指示文 $\Theta(B;z)$ の計算を壊し，要件 Q_Θ の充足を**害** (*injury*) する．このとき，優先度の低い要件 Q_Θ たちは要件の充足のための証人を壁を越えるものに取り替えて，戦略を最初からやり直す．（**初期化** (*initialization*) を行う．）しかし，この構成過程において，各要件が**害される回数が有限回**であるというのが**有限害**優先論法 (*finite injury* priority argument) の名称の由来である．

標準的な戦略木の優先論法では，古典優先論法のような**害**は発生しない．木の論法では，要件は害されるのではなく，要件のための戦略が別の戦略に**切り替わる**．そして，一度戦略 α が割り当てた証人 $\check{\alpha}$ は二度と変化しない．これを詳しく解説しよう．

まず，要件 P_Φ を割り当てられた戦略 α が $x = \check{\alpha}$ について，指示文 $\Phi(A;x)\downarrow = 0$ の成立を待っている間，自動的反応として**活性文** (*activated sentence*) $(B;x) = 0$ が満たされる．このとき，戦略 α の**アウトカム**は A であるとする．逆に指示文 $\Phi(A;x)\downarrow = 0$ が成立したとき，指示文の正しさが確認されたとして， **確認文** (*validated sentence*) $(B;x) = 1$ を満たすために x を B に放り込む．このとき，戦略 α の**アウトカム**は V であるとする．



現在の構成において，優先度の高い順に $\delta = VVVAAAVVAAVAV$ のような結果になっていたとする．優先順位は最も高いものを 1 番目の要件とする．たとえば現在稼動中の戦略 α は優先順位が 6 番目の要件を割り当てられているものと仮定しよう．つまり， α に割り当てられた要件より優先度の高い 5 つの要件について稼働中の戦略のアウトカムは，優先度の高いものから順に $VVVAA$ であり，さらに，この条件の下で現在の α のアウトカムは A である．

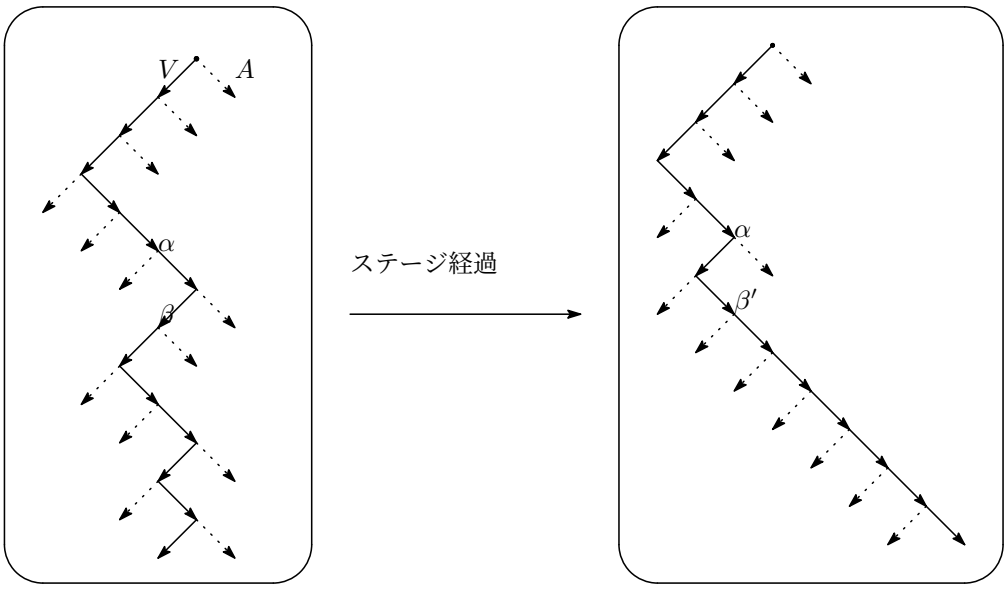
さて，ここで注意をすると，各要件についての戦略は，それより優先度の低い要件についての戦略の影響を受けない．つまり，各要件についての戦略とその現在のアウトカムは，より優先度の高い要件に関する戦略

とその現在のアウトカムのみに依存する．したがって， α を，より優先度の高い戦略（のアウトカム）全体 $VVVAA$ と同一視できる．このように，戦略はそれより優先度の高い要件の戦略のアウトカムのみに依存すると考え， $\alpha = VVVAA$ であるとみなす．同様に，たとえば現在稼働中の戦略 β で優先順位が 8 番目の要件を割り当てられているものを考える．これを $\beta = VVVAAAV$ とみなし，さらに β の現在のアウトカムは V である．

これをまとめると，現在の構成 δ は，木 $\{V, A\}^{<\infty}$ の元であり，戦略 α もまた木 $\{V, A\}^{<\infty}$ の元である．さらに，現在の構成 δ に沿う戦略 α のアウトカムは， $\delta(|\alpha|) \in \{V, A\}$ として定義される．

さて，構成の次のステージにおいて $\alpha = VVVAA$ のアウトカムが A から V に**切り替わった**と仮定しよう．α に割り当てられた要件の優先順位は 6 番目なので，古典的な構成においては， 6 番目よりも優先順位が低い 8 番目の要件は害される．しかし，木の構成においては，そうではない．$\alpha = VVVAA$ のアウトカムが A から V へ切り替わった瞬間，現在の構成も $\delta = VVVAAAVVAAVAV$ からたとえば $\delta' = VVVAAVAAAAAAAAA$ へと**切り替わり**，したがって $\beta = VVVAAAV$ はもはや現在の構成 δ' 上には存在しない．新たに稼動する 8 番目の要件を割り当てられる戦略は $\beta = VVVAAAV$ でなく，$\beta' = VVVAAVA$ なのである．つまり， 8 番目の要件は害されるのではなく， 8 番目の要件を満たすために稼動する戦略が β から β' へと切り替わるのである．

現在の構成のアウトカムから逸れた β のような戦略たちは，その稼動をやめて眠りにつく．



この構成の極限として得られる V と A の無限長の列 $P \in \{V, A\}^{\infty}$ は，各要件がどのようにして充足されるかをコードしている．つまり，たとえば $P(n) = V$ だったとしたら，優先順位が n 番目の要件についての指示文が正しいことを確認され，戦略 $P[n-1]$ が証人 x を並べる動作を行っていることを表している．この P は**トゥルーパス**と呼ばれる．有限害優先論法を木の論法で解釈したとき，トゥルーパスの次数は $\mathbf{0}'$ になる．このため，木の論法の文脈においては，有限害優先論法を **$\mathbf{0}'$-優先論法**と呼ぶ．

3.1.4 フリードバーグ・ムチニクの定理の証明

$\{R_i\}_{i\in\omega}$ を要件全体の計算可能な枚挙とする．ϱ を単調増大なグローバル変数とする．つまり，ステージ s における ϱ の値を ϱ_s とすれば，

$$(\forall s,t)\ (\ s\le t \to \varrho_s \le \varrho_t\ ).$$

戦略木 (*tree of strategies*) は次のように定義される．

$$T = \{V, A\}^{<\omega}.$$

ここで，$V < A$ なる順序 $<$ が定義されている．

$\Lambda_s \in T$ がステージ s での**現在のトゥルーパス** (*current true path*) とは，任意の $\alpha \subset \Lambda$ のステージ s でのアウトカム o_s について，$\alpha^\frown\langle o_s\rangle \subseteq \Lambda_s$ となっているときを指す．**アウトカム** (*outcome*) については後述する．

各 $\alpha \in T$ は**重み** $\breve{\alpha}$ と**戦略** S_α を持つ．ステージ 0 での重みは $\breve{\alpha}_0 = 0$ であり，$S_\alpha = \Xi(|\alpha|, \breve{\alpha})$．ここで Ξ は部分計算可能関数．$\alpha \subseteq \Lambda_s$ となっているとき，α は**行動する資格** (*eligible to act*) を持つ．$(\forall t < s)\ (\ \alpha \not\subseteq \Lambda_t\ ) \wedge (\ \alpha \subseteq \Lambda_s\ )$ となるステージ s で α に重み $\breve{\alpha}$ を

$$\breve{\alpha}_{s+1} = \min\{\ x\ :\ (\ x > \varrho\ )\ \wedge\ (\forall \beta \in T)\ x \neq \breve{\beta}_s\}.$$

なるものとする．かつ，$\breve{\alpha}$ は一度割り当てられたら二度と変化しない．つまり，上記のようなステージ s に対し，

$$(\forall t \le s)\breve{\alpha}_0 = 0\ \wedge\ (\forall t > s)\breve{\alpha}_t = \breve{\alpha}_{s+1}.$$

を満たす．したがって，これより，ステージ $t > s$ の $\breve{\alpha}_t$ を単に $\breve{\alpha}$ と書く．

重みと同様に戦略も $\breve{\alpha}$ が決まった直後に $S_\alpha = \Xi(|\alpha|, \breve{\alpha})$ が確定する．ここで，Ξ は 2 変数の部分計算可能関数で，$\Xi(i, x)$ とは証人 x で要件 R_i を満たすための戦略である．

$i = 0$ について，戦略 $S_\alpha = \Xi(\langle e, i\rangle, \breve{\alpha})$ は次により与えられる．（$i = 1$ の場合は，A と B の役割を入れ替える．）

1. $\Phi_e(A; \breve{\alpha})\downarrow = 0$ を待つ．
2. $\breve{\alpha}$ を A に並べ，壁 ϱ を $\Phi_e(A; \breve{\alpha})\downarrow = 0$ の計算のオラクル使用量 $\varphi_e(A; \breve{\alpha})$ より大きくなるように増設する．その後，戦略の稼動を停止する．

α の現在のアウトカム $o_s(\alpha)$ は，次により定義される．

$$o_s(\alpha) = \begin{cases} V & \text{if } \Xi(|\alpha|, \breve{\alpha})[s]\downarrow, \\ A & \text{o.w.} \end{cases}$$

各 $\alpha \subseteq \Lambda_s$ はステージ s で計算可能関数 $S_\alpha = \Xi(|\alpha|, \breve{\alpha})$ の計算を 1 ステップだけ進める．この計算からアウトカムを生成し，新しいトゥルーパス Λ_{s+1} を得る．

トゥルーパス (*true path*) $\Lambda \in [T]$ とは次を指す．

$$\Lambda = \lim \Lambda_s = \{\ \sigma \in T\ :\ (\exists s)(\forall t > s)\sigma \subset \Lambda_t\ \}.$$

補題 3.1 (トゥルーパス補題 (true path lemma)) 各 $\alpha \in \Lambda$ は要件の充足を保証する． □

戦略逐次木による優先論法　標準的な戦略木の優先論法の一般化として，レンプとラーマンによる**戦略の逐次木** (*iterated trees of strategies*) による優先論法の枠組みも知られている．この方法を用いると，壁（束縛）パラメータの情報や集合に枚挙するような元など，構成中で用いられる情報は全てトゥルーパスにコードされる．これは，要件の充足を直接的に保証するような（非実効的な）基本単位を最も高いレベルの木 T^n の節 α^n に割り当てることから始める．そこから指示文等の各量化を順々におさえることにより，基本単位を徐々に分解し，順々に低いレベルの木 $T^{n-1}, T^{n-2}, \ldots$ の節たち $\{\alpha_i^{n-1}\}_i, \{\alpha_{i,j}^{n-2}\}_{i,j}, \ldots$ に割り当てていく．このとき， α_i^{n-1} は α^n の**導** (*derivative*) であるといい，同様に $\alpha_{i,j}^{n-2}$ は α_i^{n-1} の導であるという．戦略の基本単位は徐々に分解されていき，最終的には T^0 の節に割り当てられた計算可能に真偽を決定できる文にまで落ち着く．したがって，T^0 上でトゥルーパスの構成は計算可能な操作で行うことができる．パス生成関数により T^0 を通るトゥルーパスから順に $T^1, T^2, \ldots$ のトゥルーパスが生成されていく．最終的に T^n 上に出来上がったトゥルーパスが要件の充足を保証する．

この文脈において n 個の木を用いる優先論法は**レベル n の優先論法**と呼ばれ，このうちレベル $1, 2, 3$ の優先論法はそれぞれ有限害 (finite injury)，無限害 (infinite injury)，途方もない害 (monstrous injury) の優先論法に対応することが知られている．

戦略逐次木による優先論法の詳細はラーマン [5] にまとまっている．

3.2 サックスの分離定理

優先論法を用いて証明する定理の別の例として，サックスの分離定理を紹介する．後の決定可能性の証明において，サックスの分離定理の証明技法は基本的な武器となる．

定理 3.1 (サックス, 1963) 任意の計算不可能な r.e. 集合 B と任意の r.e. 集合 A について， $A_0, A_1 \not\geq_T B$ なる r.e. 分割 A_0, A_1 が存在する．

3.2.1 自由落下モデル

現在，構成のステージ s であるとする．以下の議論は，全ての $e \leq s$ について考える．$A_i[s]$ の e-計算で $B[s]$ と異なる結果となる最小の n を見つける．つまり，

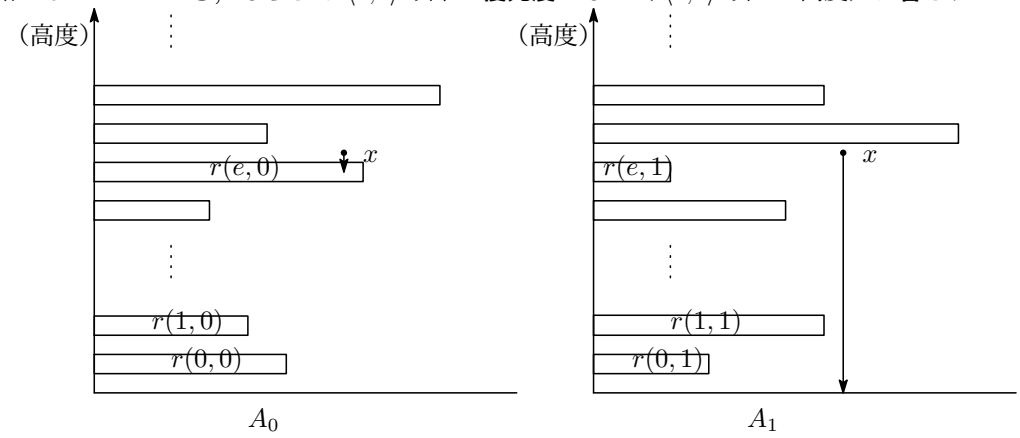
$$\Phi_e(A_i)[s] \upharpoonright n = B[s] \upharpoonright n \ \wedge \ \Phi_e(A_i; n)[s] \neq (B; n)[s].$$

なる n を見つけ，計算 $\Phi_e(A_i)[s] \upharpoonright n$ を束縛したい．

高度は $\langle e, i \rangle$ の大きさで決められているとしよう．$\langle e, i \rangle$ を上述のような計算 $\Phi_e(A_i)[s] \upharpoonright x$ のオラクル使用量の最大の大きさを持つ壁 $r(e, i)$ を高度 $\langle e, i \rangle$ の直下に建設する．

さて， x を A の $s+1$ 番目の枚挙とする．可能なら，壁を崩さないように x を A_0 か A_1 に放り込みたい．x がどの壁よりも大きいなら，とりあえず A_0 に放り込んでおけば問題ない．そうでないときを考えよう．このとき，ある高度より上ではどこかの壁にぶつかって地面に辿り着かない．なので， x を低い高度から順に放り投げ，放り投げたときに壁に弾かれて地面に辿り着かない最初の高度 $\langle e, i \rangle$ を探そう．見つけたら，その

i について A_{1-i} の方に x を放り込む．つまり，優先度の高い壁を出来るだけ壊さないようにという消極的戦略である．このとき，もちろん $\langle e,i\rangle$ 以下の**優先度**のもの（ $\langle e,i\rangle$ 以上の高度）は**害**されるかもしれない．



3.2.2 戦略の基本単位

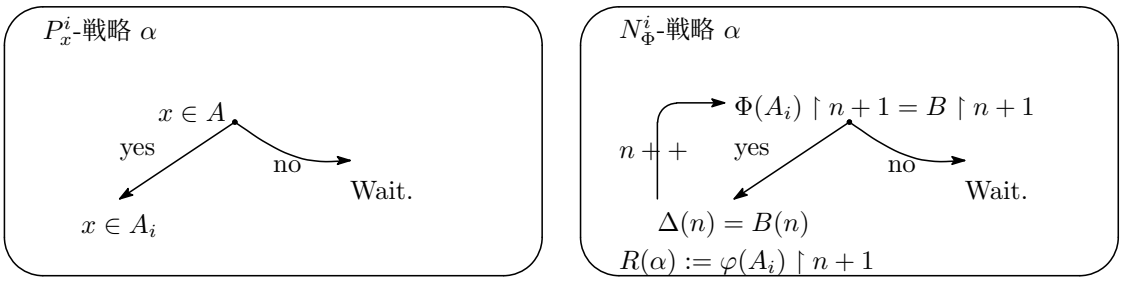
次の条件を満たすように構成を行えば，定理は従うことが分かる．

要件． 各 $x\in\omega$ と各部分計算可能関数 Φ について，

$$\mathcal{P}_x:\ x\in A\ \rightarrow\ x\in A_0\cup A_1,$$
$$\mathcal{N}^i_\Phi:\ B=\Phi(A_i)\rightarrow\ (\exists\Delta)\,(B=\Delta).$$

を要求する．ただし， A に属さない元は A_i には加えず，また， A に属す元についても A_i のどちらか一方にしか加えないようにする．

要件を満たすために，次の基本単位に沿って戦略アルゴリズムは実行される．



ここで，戦略 α が動作を開始してから s ステージ後の段階で，指示文の確認を s ステージまで行い，その段階での活性文または確認文に書かれた動作を実行する．つまり，基本単位の各ステージにおいて対応する**導**(*derivative*) を実行する．

たとえば現在，構成のステージは t であるとする．このとき，ある P^i_x-戦略 τ のステージ t までの稼働時間が s で， N^i_Φ-戦略 α の稼動時間が s' であったとする．このときステージ t での**現在のアウトカム** (*current outcome*) について， P^i_x-戦略 τ はステージ s での指示文 $x\in A[t]$ の真偽を確かめ，もし偽であれば現在のところ活性アウトカム A を， 真ならば確認アウトカム V を返す．さらに，アウトカムが V である場合， x を A_i に投げ入れて，戦略アルゴリズム τ の実行を停止する．同様に， N^i_Φ-戦略 α はステージ s' での指示文 $\Phi(A_i)[s]\upharpoonright n+1=B[s]\upharpoonright n+1$ の真偽を確かめ，もし偽であれば既に α が建設した壁の大きさ $R(\alpha)$ をアウトカムとしておく．指示文が真であれば， $\Delta(n)=B(n)$ とし，さらに指示文の計算を保つようにそのオ

ラクル計算量 $\Phi(A_i)\restriction n+1$ を新しい壁 $R(\alpha)$ を建築する．最後に， n を 1 加算し，戦略は次の新しい指示文の真偽のチェックに移行する．

3.2.3 戦略木による証明

先ほどの自由落下モデルを戦略木へと翻訳する．$\mathcal{N}$-戦略のアウトカムは，上述の自由落下モデルにおける壁の値である．$R(n)$ を高さ n の高度までにある壁の長さの最大とする．先ほどの議論を見返せば，$(R(n-1), R(n)]$ に属す元は，逆サイドの集合に放り込んでいい．したがって， $(R(n-1), R(n))]$ の間の元 x を放り込むための $\mathcal{P}_x$-戦略は， $\mathcal{N}_n$-戦略の直後に順々に置かれる．もちろん $R(n)$ より先の元についてはどちらの集合に放り込んでも何も害さないので，とりあえず A_0 に放り込む戦略 $\mathcal{P}_x^0$ を置いておく．

戦略木 T は $(\omega^* \cup \{V, A\})^{<\omega}$ の部分集合として定義される．まず， $\emptyset \in T$ には最も優先順位の高い $\mathcal{N}$-戦略が置かれ，そこから ω^* の分岐を持つ．

一般に $\mathcal{N}^i$-戦略 α の後には必ず一つの $\mathcal{P}$-戦略を置くことにする．これは，まだそれ以前に割り当てられていないような最小の $\mathcal{P}_x^{1-i}$ を割り当てる．もしこれより高い優先順位を持つ $\mathcal{N}$-戦略のアウトカムたちの最大 R と α のアウトカム $R(\alpha)$ の間 $x \in (R, R(\alpha)]$ でまだ $\mathcal{P}_x$ が割り当てられていないものがあるなら，そのような戦略 $\mathcal{P}_x^{1-i}$ を小さいものから順に戦略木の上に割り当てて行く．この作業が終わったら，次に優先順位の高い $\mathcal{N}$-戦略を次の高さの枝に割り当てる．ここで $\mathcal{N}$-戦略の直後は ω^* の分岐を持ち， $\mathcal{P}$-戦略の直後は $\{V, A\}$ の 2 分岐を持つ．

補題 3.2 トゥルーパス Λ は存在する．さらに，任意の戦略 $\alpha \subseteq \Lambda$ は要件の充足を保証する．

証明 $\alpha \subseteq \Lambda_{s_0}$ なる最小の s_0 を固定する．明らかにトゥルーパスの近似は単調減少なので，任意の $s \geq s_0$ について $\alpha \subseteq \Lambda_{s_0}$.

α が $\mathcal{P}_x$-戦略であれば成立は明らかなので， $\mathcal{N}_\Phi^i$-戦略であると仮定する．r をある $\beta \subset \alpha$ について $\beta\langle r\rangle \subseteq \alpha$ なる最大の数とする．つまり， α より優先度の高い戦略が建設する最大の壁を r とする．α のアウトカムがある値で止まるならば，明らかに $B \neq \Phi(A_i)$. さもなくば， α は Δ を全域関数として定義する．□

4 次数構造の一階理論とその断片の決定可能性

4.1 ∃-理論の決定可能性

定理 4.1 (Binns[1]) φ を解を持つ Π_1^0 問題とし， L を有限結合束とする．このとき， L の $\mathcal{P}_w(\leq \deg(\varphi))$ への $(\leq, \wedge, \vee, 1)$ を保つ埋め込みが存在する．

補題 4.1 任意の有限結合束は，最大元を保つようにして，自由有限結合束へ束の埋め込みが可能. □

補題 4.2 φ を解を持つ Π_1^0 問題， A を r.e. 集合とする．このとき， A の n 分割で，次を満たすものが存在する．

$$\forall i < n \forall f \in P \bigoplus_{j \neq i} A_j \not\geq_T f.$$

サックスの分離定理における要件 $\mathcal{N}_\Phi^i$ を次のように修正する．

要件． P を φ の解全体の集合とする．

$$\mathcal{N}_\Phi^i \ : \ \Phi_e(\bigoplus_{j\neq i} A_j) \notin P$$

証明はほとんどサックスの分離定理と同様である．変更があるのは同時に扱う戦略の数と，基本単位の些細な変化である．詳細を以下に記すが，構成の成功の保証は容易であろう．

4.1.1 自由落下モデル

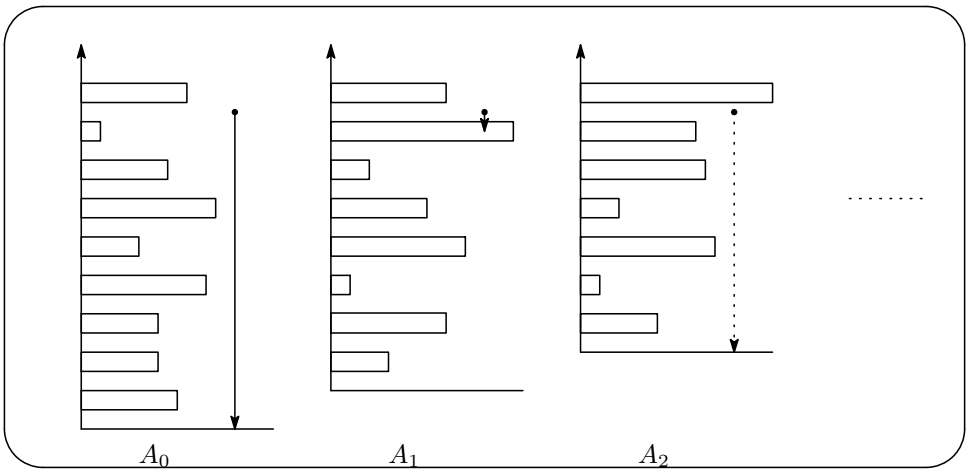
現在，構成のステージ s であるとする．以下の議論は，全ての $e \leq s$ について考える．$\bigoplus A_j[s]$ の e-計算で， P をパスとする木 T_P の中にまだ伸びてこない最小の n を見つける．つまり，

$$\Phi_e(\bigoplus_{j\neq i} A_j)[s]\upharpoonright n \in T_P[s] \ \wedge \ \Phi_e(\bigoplus_{j\neq i} A_j; n)[s] \notin T_P[s].$$

なる n を見つけ，計算 $\Phi_e(\bigoplus A_j)[s]\upharpoonright n$ を束縛したい．

高度は $\langle e,i\rangle$ の大きさで決められているとしよう．$\langle e,i\rangle$ を上述のような計算 $\Phi_e(A_i)[s]\upharpoonright x$ のオラクル使用量の最大の大きさを持つ壁 $r(e,i)$ を高度 $\langle e,i\rangle$ の直下に建設する．

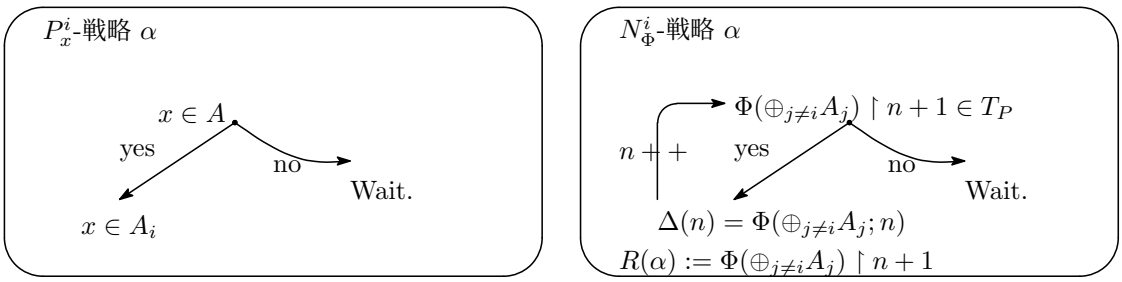
さて， x を A の $s+1$ 番目の枚挙とする．可能なら，壁を崩さないように x をどれかの A_i に放り込みたい．x がどの壁よりも大きいなら，とりあえず A_0 に放り込んでおけば問題ない．そうでないときを考えよう．このとき，ある高度より上ではどこかの壁にぶつかって地面に辿り着かない．なので， x を低い高度から順に放り投げ，放り投げたときに壁に弾かれて地面に辿り着かない最初の高度 $\langle e,i\rangle$ を探そう．見つけたら，その i について A_{i-1} の方に x を放り込む．ここで， A_{-1} は A_{n-1} のこととする．つまり，優先度の高い壁を出来るだけ壊さないようにという消極的戦略である．このとき，もちろん $\langle e,i\rangle$ 以下の**優先度**のもの（ $\langle e,i\rangle$ 以上の高度）は**害**されるかもしれない．



高度の低い順に $A_0, A_1, \ldots$ と順にチェックしていき，最初に壁が弾かれる高度に達してしまったら，そこより先では計算を害してしまうので，その直前にチェックしていた集合に元を投げ入れる．（図では A_1 で壁に弾かれてしまったので， A_0 に元を放り込む．）

4.1.2 戦略の基本単位

次の基本単位に沿って戦略は従う．



ここで，各戦略はその稼動時間が s の段階で各文の s ステージでの段階の操作を実行する．

4.1.3 戦略木

戦略木 T は $(\omega^* \cup \{V, A\})^{<\omega}$ の部分集合として定義される．まず，$\emptyset \in T$ には最も優先順位の高い $\mathcal{N}$-戦略が置かれ，そこから ω^* の分岐を持つ．

一般に $\mathcal{N}^i$-戦略 α の後には必ず一つの $\mathcal{P}$-戦略を置くことにする．これは，まだそれ以前に割り当てられていないような最小の $\mathcal{P}^{1-i}_x$ を割り当てる．もしこれより高い優先順位を持つ $\mathcal{N}$-戦略のアウトカムたちの最大 R と α のアウトカム $R(\alpha)$ の間 $x \in (R, R(\alpha)]$ でまだ $\mathcal{P}_x$ が割り当てられていないものがあるなら，そのような戦略 $\mathcal{P}^{1-i}_x$ を小さいものから順に戦略木の上に割り当てて行く．この作業が終わったら，次に優先順位の高い $\mathcal{N}$-戦略を次の高さの枝に割り当てる．ここで $\mathcal{N}$-戦略の直後は ω^* の分岐を持ち，$\mathcal{P}$-戦略の直後は $\{V, A\}$ の 2 分岐を持つ．

補題 4.3 トゥルーパス Λ は存在する．さらに，任意の戦略 $\alpha \subseteq \Lambda$ は要件の充足を保証する． □

証明 (定理) $FD(n)$ を $\mathcal{P}_w(\leq \deg(\varphi))$ へ $(\leq, \wedge, \vee, 1)$ を保つように埋め込めることを示せば十分．$\deg(\mathsf{Sep}(A, B)) = \mathbf{1}$ であるような A, B を固定する．このとき，$\{P \wedge \mathsf{Sep}(A_i, B)\}_{i\leq n}$ が $FD(n)$ の自由生成子であることを示す．このために，次を示せば十分．

$$\forall I \subsetneq n \ \big(\ P \wedge \bigvee_{i\in I} \mathsf{Sep}(A_i, B) \not\geq_w P \wedge \bigwedge_{i\notin I} \mathsf{Sep}(A_i, B).$$

しかし，もし $\{\oplus_{i\in I} A_i\} \geq_w \bigwedge_{i\notin I} \mathsf{Sep}(A_i, B)$ なら，ある $j \notin I$ について $\{\oplus_{i\in I} A_i\} \geq_w \mathsf{Sep}(A_j, B)$．これより，

$$\{\bigoplus_{i\neq j} A_j\} \geq_w \bigvee_{i<n} \mathsf{Sep}(A_i, B) \equiv_w \mathsf{Sep}(A, B) \geq_w P.$$

これは $\mathcal{N}^i_\Phi$ に矛盾する．故に $\{\oplus_{i\in I} A_i\} \not\geq_w P \wedge \bigwedge_{i\notin I} \mathsf{Sep}(A_i, B)$．よって，$P \wedge \bigvee_{i\in I} \mathsf{Sep}(A_i, B) \not\leq_w P \wedge \bigwedge_{i\notin I} \mathsf{Sep}(A_i, B)$．また，$FD(n)$ の最大元は $P \wedge \bigvee_{i<n} \mathsf{Sep}(A_i, B) \equiv_w P$.

定理 4.2 (Binns[1]) $\mathcal{P}_w$ の $\exists(\leq, \wedge, \vee)$-理論は決定可能.

証明 (スケッチ) $\mathcal{P}_w$ 自身が結合束であることと，任意の有限結合束が $\mathcal{P}_w$ に埋め込めることを用いて，$\mathcal{P}_w$ で $\exists \vec{x} \psi(\vec{x})$ が成り立つかどうかを確かめるためには，ψ を満たす有限結合束が存在するかどうかを調べればよい．ψ の変数の数 n に応じた大きさ 2^{2^n-2} の束だけについて調べれば十分なので，これは有限的に決定できる． □

定理 4.3 (Binns[1]) $\mathcal{P}_w$ の $\exists(\leq, \wedge, \vee, \mathbf{1})$-理論は決定可能. □

定理 4.4 (Binns[1]) $\mathcal{P}_w$ の $\exists(\leq, \wedge, \vee, \mathbf{0}, \mathbf{1})$-理論は決定可能. □

4.2 $\mathcal{P}_s$ の稠密性定理

∃-理論の決定可能性が分かったので，次は ∀∃-理論の決定可能性に挑戦したい．ただし，いきなり一般的な問題に当たるのは難しいだろう．したがって，少しずつ具体的な ∀∃-文から決定していくことを目標にしよう．しかし，まず， ∀∃-文の中で最も簡単な部類に属す稠密性の問題ですら， $\mathcal{P}_w$ での真偽は未解決である．

問題 1 Π^0_1 問題の弱次数構造は稠密か？　つまり，

$$\mathcal{P}_w \models \forall \mathbf{p}, \mathbf{q}\ (\ \mathbf{q} < \mathbf{p} \to \exists \mathbf{r}\ (\ \mathbf{q} < \mathbf{r} < \mathbf{p}\ )\ ).$$

は成立するか？

注意　$\mathcal{P}_w$ の任意の次数 $\mathbf{p}$ に対して，その下に有限結合束を埋め込めるので， $\mathcal{P}_w$ は下に稠密である．

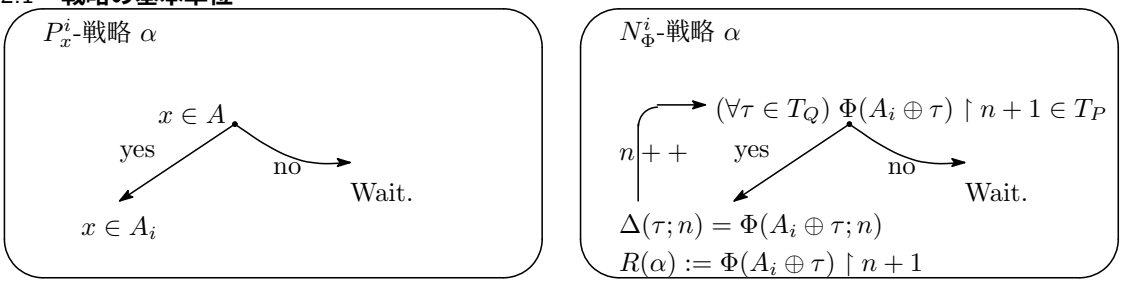
弱次数構造の稠密性が未解決なのに対し，強次数構造の稠密性は非常に簡単に証明できる．

補題 4.4 $P >_s Q$ をそれぞれ解を持つ Π^0_1 問題とする．このとき， A が r.e. 集合ならば， A の分割 A_0, A_1 で次を満たすものが存在する．

$$\{A_0\} \vee Q \not\geq_s P\ \wedge\ \{A_i\} \vee Q \not\geq_s P.$$

戦略木（および自由落下モデル）はサックスの分離定理等と全く同様である．戦略の基本単位に少しの修正を要求する．

4.2.1 戦略の基本単位



補題 4.5 トゥルーパス Λ は存在する．さらに，任意の戦略 $\alpha \subseteq \Lambda$ は要件の充足を保証する．　□

定理 4.5 (Binns[1]) $\mathcal{P}_s \models \forall \mathbf{p}, \mathbf{q}\ (\ \mathbf{q} < \mathbf{p} \to \exists \mathbf{r}\ (\ \mathbf{q} < \mathbf{r} < \mathbf{p}\ )\ ).$

証明　A, B を $\mathsf{Sep}(A, B)$ の強次数が $\mathbf{1}$ であるようなものとする．A_0, A_1 を補題のようなものとし， $S = \mathsf{Sep}(A, B)$, $S_i = \mathsf{Sep}(A_i, B)$ とする．このとき， $X \in S_0$ かつ $Y \in S_1$ ならば $X \cup Y \in S$ なので， S_0, S_1 は極大対となっている．この極大対を (P, Q) 間に埋め込みたい．このためには， S_i を Q の上に持ち上げて，P の下に入れればよい．つまり， $P_i = P \wedge (S_i \vee Q)$ とする．$S_i \vee Q \not\geq_s P$ なので， $P_i <_s P$. また， P_s は結合則を満たすので，

$$P_0 \wedge P_1 = (P \wedge (S_0 \vee Q)) \vee (P \wedge (S_1 \vee Q)) \equiv_s P \wedge (S_0 \vee S_1 \vee Q) \equiv_s P.$$

により，少なくともどちらかの i について $P_i >_s Q$.　□

5 $\mathcal{R}$ の稠密性定理

r.e. 集合 U と V の中間次数をもつ r.e. 集合 A を構成するためには，まず U をコードし，さらに V により**許可** (*permit*) されるようにすればよい.

5.1 許可法

稠密性定理の証明のために，フリードバーグの**許可法** (*permitting method*) を簡単な例と共に紹介する.

定理 5.1 $\mathcal{R} \models (\forall \mathbf{v} > \mathbf{0})(\exists \mathbf{a_0}, \mathbf{a_1} \leq \mathbf{v})\ (\ \mathbf{a_0} \not\leq \mathbf{a_1}\ \wedge\ \mathbf{a_1} \not\leq \mathbf{a_0}\ )$.

いつものように比較不可能な次数をもつ r.e. 集合 A_0 と A_1 を構成したいが，今回の場合， A_0 と A_1 にどのような元を並べたかを V が知っていなければならない．では，どのように A_i に元を投げ入れたことを V に知らせればよいだろうか．V に元の枚挙の知らせるために，指示文が真になったとしてもすぐには A_i に元を並べず， V の許可が降りるまで待機しておく．A_i に並べたい元 x に対応する $\check{x}$ について，もし $V \upharpoonright \check{x}$ の変化があったなら， V の許可が降りたとして x を A_i に並べてよい．このような操作を行えば，いつ A_i に元が並べられ得るかを V は知ることができる.

しかし，これにはまだ問題点がある．x に A_i を投げ入れたいという状況になったとして， V の許可を待っていたとしても，もしかしたら一生許可は降りず永遠に待ちぼうけを食うかもしれない．このために，単に V の許可を待つだけという消極的な心構えで構成を行うのではなく，次の証人の候補 x_1 を選び，この新しい x_1 で同様の戦略を実行するのである．この操作を延々続けると， V の許可待ちの $x, x_1, x_2, \ldots$ が生成される．このとき，これらのどれかについては V による枚挙の許可が下りるだろう．なぜなら，さもなくば各 n について $V \upharpoonright x_n$ を計算可能に確定でき，したがって V の計算可能性を導いてしまう.

実際にこの操作を戦略木の論法によって示そう．満たすべき要件は次である.

要件.

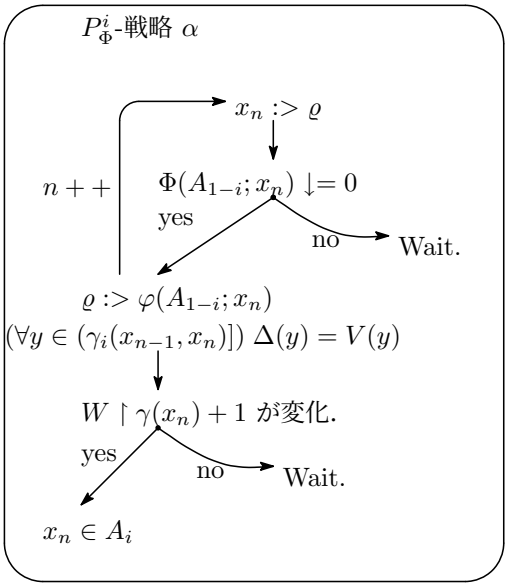
$$R^i : A_i = \Gamma_i(W)$$
$$P^i_\Phi : A_i = \Phi(A_{1-i}) \to (\exists \Delta) W = \Delta.$$

要件 R^i はグローバル戦略によって満たす．まず，各ステージ s で各 $x \leq s$ について $\Gamma_i(V; x) = A_i(x)$ をオラクル使用量 $\gamma_i(V; x) = x$ となるように定義する.

この戦略により， $\gamma_i(V; x)$ の下で V の変化があったとき，これまでの $\Gamma_i(V; x)$ の計算が崩れ， $\Gamma_i(V; x)$ を別の値として定義できるようになる．したがって，この隙にすかさず x を A_i に枚挙したとしたら，$\Gamma_i(V; x) = A_i(x)$ を保つように再定義できる．よって A_i に x を枚挙する際には， $\gamma_i(V; x)$ の下での V の変化を待てばよい.

また，構成のために，グローバル変数 ϱ を用意しておく．これは束縛（壁）パラメータとして用いる.

P^i_Φ-戦略は，基本的にはフリードバーグ・ムチニク構成であるが，元の枚挙の前に V による許可を必要とする.

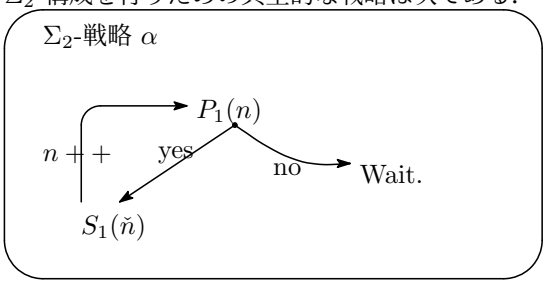


5.2 Σ_2 構成

稠密性証明のための次の段階として，満たしたい要件の複雑さから，どのような戦略を立てればいいかの方針を検討する．まず，要件の複雑さが Σ_2 であるような場合について考察しよう．

Σ_2 構成は，基本単位の指示文 P が $\forall n P_1(n)$ という Π_2 の形であり，確認文も $\forall n S_1(n)$ という Π_2 の形であるようなものである．典型的な例として， Π_2 条件 P に対して $P \to \bot$ という要件を満たし，したがって $\neg P$ という Σ_2 条件が従うという形のものである．戦略は順に指示文 $P_1(0), P_1(1), \cdots$ をチェックしていき， $P_1(n)$ が真であることを確認したら，対応する確認文 $S_1(\check{n})$ を実行し， $P_1(n+1)$ の確認に移る．たとえば，サックスやビンズの分離定理が Σ_2 構成に該当する．また，上述のフリードバーグ・ムチニクの定理と許可法の組み合わせも Σ_2 構成の代表例である．

Σ_2 構成を行うための典型的な戦略は次である．



この構成に沿うことにより， $\exists n \neg P_1(n)$ であるか，さもなくば $\forall n S_1(n)$ が導かれる．よって，要件で求められるような $\forall n P_1(n) \to \forall n S_1(n)$ が従う．

5.3 Π_2 構成

Π_2 構成は，基本単位の指示文 P が Π_2 文 $\forall n P_1(n)$ の形であるような構成である．このとき，活性文（**有限的アウトカム**， Σ アウトカム）は $\exists n \neg R_1(n)$ の形であり，確認文（**無限的アウトカム**， Π アウトカム）は $\forall n \neg S_1(n)$ の形である．戦略木は現在分かっている指示文の状況に応じたアウトカムをもつ．戦略は順に指

示文 $P_1(0), P_1(1), \cdots$ をチェックしていき， $P_1(n)$ が真であることを確認したら $P_1(n+1)$ の確認に移る．ある $P_1(n)$ が真であることをチェックできた瞬間を**拡張ステージ** (*expansionary stage*) と呼び，その瞬間は「もしかしたら $\forall n P_1(n)$ になるんじゃないか」と推測して一時的に無限的アウトカム ∞ を返す．拡張ステージ以外では，現在確認中の指示文に対応する有限的アウトカム w_n を返す．つまり，アウトカムの集合は次により定義される．

$$\{\infty < \ldots < w_{n+1} < w_n < \ldots < w_1 < w_0\},$$

あるいは，単に

$$\{\forall n P_1(n) < \ldots < \neg P_1(n+1) < \neg P_1(n+1) < \ldots < \neg P_1(1) < \neg P_1(0)\}.$$

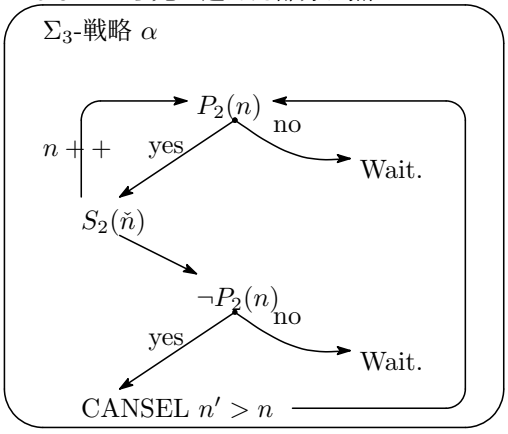
とする．

以前と同様に，戦略木の各節 α には戦略 S_α, 重み $\breve{\alpha}(n)$ を割り当てる．壁パラメータは節には割り当てずに，グローバル変数 ϱ としておく．重みの初期値は 0 である．重みの定義方法の例としは，単に $\breve{\alpha(n)} = n$ とする戦略もあれば，次のような方法もある．まず，現在のトゥルーパスがその戦略 α に初めて到達した時点で，グローバル壁パラメータを越える最小元を $\breve{\alpha}(0)$ に割り当てる．各重み $\breve{\alpha}(n)$ は，戦略の n 段階目，つまり，指示文を $P_1(n)$ とし，活性文は $\neg R$ を保証し，確認文は S の部分を保証するような構成において用いることが出来る．重み $\breve{\alpha}(n)$ はそれが決定される段階になると，グローバル壁パラメータ ϱ を越える値として定義される．壁は何度でも自由に大きくすることができるが，壁の更新はアウトカムの変更，すなわちトゥルーパスの切り替えを伴う．

5.4 Σ_3 構成

Σ_3 構成は，指示文と確認文がそれぞれ Π_3 文 $\forall n P_2(n)$ と $\forall n S_2(n)$ の形の要件であるような構成である．典型的な例として，要件が $P \to \bot$ を意味するものであり，これにより Σ_3 文 $\neg P$ の成立を保証するという形の構成となる．

Σ_2 及び Π_2 構成と同様に，指示文 $P_2(n)$ が真であるたびに n に 1 を加算して次に進むが， $P_2(n)$ が Σ_2-文であるために，幾つかの問題が起こる．Σ_2-構成のように，指示文の各断片 $P_1(n)$ が Σ_1-文の場合，その結果が崩れないような束縛を行っていれば，その戦略が動いている間に $P_1(n)$ が変わることはない．しかし，$P_2(n)$ が Σ_2-文であるとき，他の戦略などは無関係に単なるステージ経過により $P_2(n)$ が再変化する可能性がある．したがって，戦略が進んでいる最中にも常に $P_2(n)$ が崩れていないかどうかの監視を行い，もし崩れてしまったら先に進んだ部分は無かったことにする．



可能なアウトカムは，$P_2(n)$ が真になっているのを待っている間は**有限的アウトカム** (*finitary outcome*) w_n を返し，真であることを確認したがステージ経過によって $\neg P_2(n)$ となってしまったとき**無限的アウトカム** (*infinitary outcome*) ∞_n を返す．ただし，複雑さが Σ_3 の段階の構成では，これらを区別する必要がないので，共にアウトカムは n であるとしてよい．したがって，アウトカムの集合は ω^* により定義される．アウトカムがある n に安定する場合は $\neg P_2(n)$ が従う．他方，延々構成は n を増やし続け，アウトカムが発散する場合は $\forall n P_2(n) \wedge \forall n S_2(n)$ が従う．Σ_3 構成では $\forall n S_2(n)$ は矛盾 $\perp$ を導くため，このようなことは起こりえず，必ずアウトカムは収束し，トゥルーパスは存在する．よって，結論として Σ_3 文 $\exists n \neg P_2(n)$ が導かれる．

5.5 $\mathcal{R}$ の稠密性定理

定理 5.2 (サックス) $\mathcal{R} \models (\forall \mathbf{u}, \mathbf{v})(\exists \mathbf{a})\ (\ \mathbf{u} < \mathbf{v} \to \mathbf{u} < \mathbf{a} < \mathbf{v}\ )$.

証明方針は，許可法と Σ_3 構成の組み合わせである．要件は，許可法のグローバル戦略 R と Σ_3 戦略 N_Φ, P_Ψ からなる．

要件．

$$\begin{aligned} R &: A = \Theta(U \oplus V), \\ N_\Phi &: V = \Phi(A \oplus U) \to (\exists \Gamma)\ V = \Gamma(U), \\ P_\Psi &: A = \Psi(U) \to (\exists \Delta)\ V = \Delta(U). \end{aligned}$$

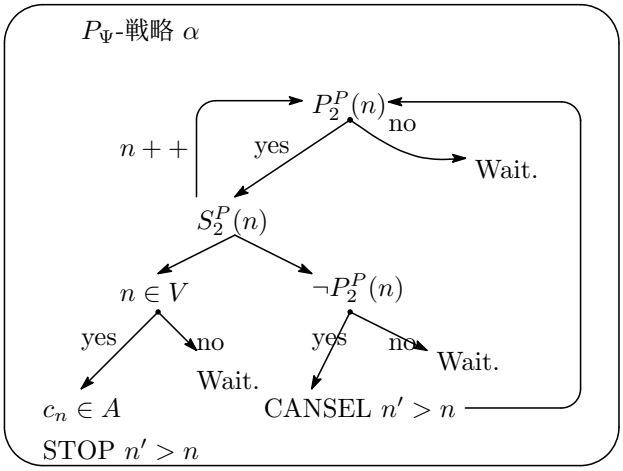
戦略． グローバル戦略 R は，ステージ s において，各 $x \leq s$ について未定義な $\Theta(U \oplus V; x)$ を $A(x)$ と同じ値に再定義する．オラクル使用量については，N-要件のための戦略と P-要件のための戦略を立てた後に述べる．

まず，N-要件を満たすための戦略を考えよう．N-要件の基本単位の指示文 P^N は $\Phi(A \oplus U) = V$ であり，確認文 S^N は $(\exists \Gamma)\ \Gamma(U) = V$ である．これを実際に稼動可能なものに分解したい．まず，導かれる指示文 $P_2^N(n)$ は $\Phi(A \oplus U; n) = V(n)$ であり，確認文 $S_2^N(n)$ は $\Gamma(U; n) = V(n)$ である．ここで，確認文 S_2^N を実行する際の Γ のオラクル使用量 $\gamma(U; n)$ は $\Gamma(U; n)$ と $\Phi(A \oplus U; n)$ と連動させるために $\gamma(U; n) = \varphi(A \oplus U; n)$ とし，さらに，A の変化によっては指示文 $P_2^N(n)$ を壊さないために，グローバル壁パラメータ ϱ を $\varphi(A \oplus U; n)$ より大きい値に更新する．後は単に Σ_3 構成の戦略に従えばよい．

次に P-要件を満たすための戦略を考える．これはサックスのコーディング戦略である．P-要件を満たすために，何らかの元を A に並べる操作が現れるが，このためには $U \oplus V$ の許可が降りるまで待たなければならない．このために，単に Σ_3 構成を行うのではなく，これを許可法と組み合わせなければならない．まず，基本単位の各 n 番目の分解が現れたステージで，A に並べる候補となる元（**コーディング位置**;*coding location*）c_n を壁 ϱ を越える値として取る．また，V に n が並べられたとき，A に c_n を並べる許可が下りたとみなす．基本単位の指示文 P^P は $\Psi(U) = A$ で，確認文 S^P は $(\exists \Delta)\ \Delta(U) = V$ なので，それぞれの各分解として $P_2^P(n)$ は $\Psi(U) \upharpoonright c_n + 1 = A \upharpoonright c_n + 1$ を取り，$S_2^P(n)$ は $\Delta(U; n) = V(n)$ を取る．ここで，確認文 $S_2^P(n)$ を実行する際の Δ のオラクル使用量 $\delta(U; n)$ は $\Delta(U; n)$ 指示文の状態（特に $\Psi(U; c_n)$ ）と連動させるために $\delta(U; n) = \psi(U; c_n)$ としておく．これに対して Σ_3 構成の戦略に基本的には従うが，許可戦略と組み合わせるために，若干の修正を要する．

まず，c_n を選ぶ段階に達した時点において，既に $n \in V$ であれば，絶対に c_n を A に並べる許可は下りないので，この段階は無視して次の $n+1$ の段階に進んでよい．そうでなければ，Σ_3 構成に従い，確認文ま

で実行した後，指示文 $P_2^P(n)$ が崩れないか監視すると同時に V の許可が下りるのを待つ．指示文 $P_2^P(n)$ が崩れてしまったら，許可を待つのもやめて， Σ_3 構成に従い， n 段階の戦略を最初からやり直す．指示文が崩れる前に許可が下りたなら，つまり n が V に並べられたなら， c_n を A に並べ， n より先の段階の戦略の稼動を一旦止める．



最後に，グローバル戦略 R が定義する関数 $\Theta(U \oplus V; x)$ のオラクル使用量 $\vartheta(U \oplus V; x)$ を与える．これは最終的に任意の x について $\Theta(U \oplus V; x) \downarrow = A(x)$ となるように構成を行わなければならない．

1. もう $A(x)$ が変化しないと確信したとき，オラクル使用量 $\vartheta(U \oplus V; x)$ は 0 でよい．つまり，これは現在既に $x \in A$ であるか， $x = c_n$ となるような P-戦略が無いとき，もしくは $x = c_n$ であったとしても c_n を割り当てられた時点で既に $n \in V$ であると分かっているために即座に次の段階 $n+1$ へと戦略が進んだ場合である．
2. ある P-戦略が $c_n = x$ としていて，さらに現在 $n \notin V$ であるとき．つまり， $n \in V$ を待っており，そうなったら即座に c_n を放り込む準備をしているとき．このときは，上述のような許可戦略を行いたい場合であるので，オラクル使用量は n としておけばよい．
3. そうでない場合．つまり， $c_n = x$ と割り当てた P-戦略 α が V の許可を待っていたとする．この待機中に α は現在のトゥルーパスから逸れて眠ってしまうが，その最中に $n \in V$ が並べられたときを考える．このとき， α が再び目を醒ました途端に $c_n \in A$ と置くが，ちょうどそのときに Θ の計算を変えなければならない．$\beta^\frown\langle m \rangle \subseteq \alpha$ を仮定する．β のアウトカムが m に切り替わるのは， β が N-戦略であれば，それに対応する $\Phi_\beta(A \oplus U; m)$ が変化したときであり， β が P-戦略であれば，それに対応する $\Psi_\beta(U; c_{m,\beta})$ が変化したときである．よって，この変化を起こし得る元が並べられたときには念のため Θ を破壊するようにオラクル使用量を決める．β_0 を現在のトゥルーパス Λ_s と α の交点とする．

$$\begin{aligned}\vartheta(U \oplus V; x) = \min(\ \{\ \varphi_\beta(A \oplus U; m)\ :\ \beta_0 \subset \beta^\frown\langle m \rangle \subseteq \alpha\ \wedge\ \beta \text{ is } N_{\Phi_\beta}\text{-strategy.}\ \} \cup \\ \{\ \psi_\beta(U; c_{m,\beta})\ :\ \beta_0 \subset \beta^\frown\langle m \rangle \subseteq \alpha\ \wedge\ \beta \text{ is } P_{\Psi_\beta}\text{-strategy.}\ \}\ )\end{aligned}$$

補題 5.1 トゥルーパス Λ は存在する．さらに，任意の戦略 $\alpha \subseteq \Lambda$ は要件の充足を保証する． □

6　次数の理論の決定不可能性

ある理論の決定不可能性を導くためには，算術などの既に決定不可能であると知られている理論のモデルをコードすることを示せばよい．たとえば，ある次数構造 $\mathcal{S}$ の一階理論 $Th(\mathcal{S})$ の決定不可能性を導く一つの方法として，算術の標準モデルをコードする方法がある．つまり，言語 $\{\leq\}$ の論理式 $\varphi_N(x), \varphi_+(x,y,z), \varphi_\times(x,y,z)$ を見つける．ここで，各論理式が $\mathcal{S}$ での成立と，それぞれ $x \in \mathbb{N}$ と $\mathbb{N}$ で $x+y=z$ および $x \cdot y = z$ が成り立ち，かつ基本的な公理（たとえば PA^- ）を満たしていることに対応しているとする．

このような論理式を見つけることができたと仮定する．各 $\{+,\cdot\}$-論理式 φ の $x+y=z$ および $x \cdot y = z$ の部分をそれぞれ $\varphi_+(x,y,z)$ と $\varphi_\times(x,y,z)$ に変換し，さらに φ に現れる $\exists x \psi$ を $(\exists x)\,(\,\varphi_N(x) \wedge \psi\,)$ にし，$\forall x \psi$ を $(\forall x)\,(\,\varphi_N(x) \to \psi\,)$ に変換したものを φ^* と書くことにする．このとき，各 $\{+,\cdot\}$-論理式 φ について，

$$\mathcal{S} \models \varphi^* \;\Leftrightarrow\; \mathbb{N} \models \varphi$$

となる．このようにして，もし，ある理論において，算術の翻訳を見つけることができたならば，算術の決定不可能性からその理論の決定不可能性を導くことができる．

例として $\mathcal{D}$ と $\mathcal{R}$ の一階理論の決定不可能性証明を紹介する．これを参考にして $\mathcal{P}_w$ あるいは $\mathcal{P}_s$ の一階理論の決定不可能性について考察したい．

6.1　$\mathcal{D}$ の決定不可能性

補題 6.1 (可算反鎖のコード) $\mathbf{A}$ を $\mathcal{D}$ の可算反鎖とし， $\mathbf{b}$ を $\mathbf{A}$ の上界とする．このとき，ある $\mathbf{g}_1, \mathbf{g}_2$ が存在して， $\mathbf{A}$ は次を満たす $\mathbf{b}$ 以下での極小解全体の集合となる．

$$\mathbf{x} < (\mathbf{g}_1 \vee \mathbf{x}) \wedge (\mathbf{g}_2 \vee \mathbf{x}).$$

証明 (スケッチ) $\mathbf{G}_1$ と $\mathbf{G_2}$ を $\mathbf{A}$ に関する情報を幾つか共有し，他のものに対しては独立となるようにする．このためには $\mathbf{A}$ の元をコードする十分に**ジェネリック** (*generic*) な集合を構成すればよい．$\mathbf{A}$ の各代表元の集合 $\{A_i : i \in \omega\}$ に対して， $A_i^* = \{\sigma : \sigma \subset A_i\}$ と定義する．

強制法．　強制条件 (*forcing condition*) は 3 つ組 (p_1, p_2, l_p) で $p_1, p_2 \in 2^\omega$ かつ $l_p \in \omega$ なるものとする．後で定義する順序の意味で，十分多くの稠密な集合に沿って p_1, p_2 を拡張して行った先が目的の次数 $\mathbf{g}_1, \mathbf{g}_2$ を持つ集合 G_1, G_2 である．l_p は現在までにコードを始めている A_i^* の数を表す．また， $p = (p_1, p_2, l_p)$ を強制条件としたとき， (k,a) が p の**コーディング場所** (*coding location*) とは， $(k,a) > lh(p_1) = lh(p_2), k < l_p$ かつ $a \in A_k$ を満たすときを指す．発想としては，コーディング場所 (k,a) において， $a \in A_k^*$ であることを $p_1(k,a) = p_2(k,a)$ によりコードしていこうというものである．

強制条件たちの間に次の順序を定義する．

$$\begin{aligned} q \leq p \;\leftrightarrow\; & q_1 \supset p_1 \;\wedge\; q_2 \supset p_2 \;\wedge\; l_q \geq l_p \;\wedge \\ & (\forall k < l_p)\,\big(\,(\,a \in A_k^* \;\wedge\; lh(p_1) < (k,a) \leq lh(q_1)\,) \;\to\; q_1(k,a) = q_2(k,a)\,\big). \end{aligned}$$

G_1 と G_2 を上の強制法により得られる B-**算術的ジェネリック** (B-*arithmetical generic*) とする．このとき， A_k を用いて A_k^* のコーディング場所における G_1 と G_2 の共通値を求められる．一方ジェネリック性に

より， B-算術的な操作，特に A_k からのどんな計算も， G_1 と G_1 はあるコーディング場所でその計算結果を避ける．つまり，

$$C_k = \{\, G_1(k,a) \;:\; a \in A_k^* \,\}$$

とすれば， $\mathbf{a}_k \not\geq \mathbf{c}_k \leq (\mathbf{g}_1 \vee \mathbf{a}_k) \wedge (\mathbf{g}_2 \vee \mathbf{a}_k)$．さらに， G_1, G_2 のジェネッリック性により， $\mathbf{A}$ のどの次数もコードしないような $\mathbf{b}$ 以下の任意の次数 $\mathbf{d}$ に対しては， $\mathbf{g}_1, \mathbf{g}_2$ は独立な性質を持つので，これより $\mathbf{d} = (\mathbf{g}_1 \vee \mathbf{d}) \wedge (\mathbf{g}_1 \vee \mathbf{d})$ が従う．詳しい証明は Slaman-Woodin[11] を参照． □

補題 6.2 (可算集合のコード) $\mathbf{S}$ を $\mathcal{D}$ の可算集合とし， $\mathbf{b}$ を $\mathbf{S}$ の上界とする．このとき， $\mathbf{S}$ は $\mathcal{D}$ で有限個のパラメータを用いて定義可能．

証明 可算反鎖ならコードできるので，可算反鎖を経由してうまく $\mathbf{S}$ を特徴づければよい．$\mathbf{G}$ を $\mathbf{b}$-算術的な互いにコーエンジェネリック次数であるような可算集合とし， ψ を $\mathbf{S}$ から $\mathbf{G}$ への全単射とする．このとき，$\mathbf{A} = \{\mathbf{x} \vee \psi(\mathbf{x}) : \mathbf{x} \in \mathbf{S}\}$ とすれば， $\mathbf{A}$ と $\mathbf{G}$ は反鎖なので，それぞれ有限個のパラメータを用いて定義可能．これより， $\mathbf{S}$ は次により定義できる．

$$\mathbf{x} \in \mathbf{S} \;\leftrightarrow\; (\; \mathbf{x} < \mathbf{b} \;) \;\wedge\; (\exists \mathbf{g} \in \mathbf{G})(\exists \mathbf{a} \in \mathbf{A})\; (\; \mathbf{x} \vee \mathbf{g} = \mathbf{a} \;).$$

□

注意 $\mathbf{G}$ を B-算術的な互いにコーエンジェネリック次数であるような可算集合とする．さらに， ψ を $\mathbf{S}$ から $\mathbf{G}$ への全単射とする．このとき， 上の定理の証明と同様にして， ψ は $\mathcal{D}$ のパラメータを用いて定義可能である．

定理 6.1 (Slaman-Woodin[11]) $\mathcal{D}$ の n 項可算関係のコード式 $\mathrm{Code}^n(\vec{x}; \vec{y})$ が存在する．つまり，任意の $\mathbf{R} \subseteq \mathcal{D}^n$ に対し，コード $\vec{\mathbf{p}}$ が存在して，次を満たす．

$$(\forall \vec{\mathbf{d}})\; (\; \vec{\mathbf{d}} \in \mathbf{R} \;\Leftrightarrow\; \mathcal{D} \models \mathrm{Code}^n(\vec{\mathbf{d}}; \vec{\mathbf{p}}) \;).$$

証明 (スケッチ) 一旦，各座標に分解することにより，可算集合に変換してコードする．その後，それを繋ぎ合わせればよい． □

定理 6.2 (シンプソン，1977) $\mathcal{D}$ の一階理論の中で算術の二階理論を計算可能に翻訳可能．特に $Th(\mathcal{D})$ は決定不可能である．

証明 (スケッチ) 算術の標準モデル $\mathbb{N}$ は $\mathcal{D}$ の可算関係の有限個の組として表現できる．二階の量化については，たとえば $\mathbb{N} \models \exists X \varphi$ なら， $\mathbb{N}$ のコードを $\vec{\mathbf{n}}$ とすれば，次により表現できる．

$$\mathcal{D} \models (\exists \vec{\mathbf{p}})\; \big(\; (\forall \mathbf{x})\; (\; \mathrm{Code}(\mathbf{x}; \vec{\mathbf{p}}) \to \mathrm{Code}(\mathbf{x}; \vec{\mathbf{n}}) \;) \;\wedge\; \varphi \;\big).$$

□

6.2 $\mathcal{R}$ の決定不可能性

定義 6.1 ハーリントン-シェラー集合 (*Harrington-Shelah set*) **HS** と**スレイマン-ウディン集合** (*Slaman-*

Woodin set) **SW** を次により定義する．

$$\mathbf{HS}(\mathbf{r},\mathbf{b},\mathbf{c}) = \{\mathbf{g} \leq \mathbf{r} : \mathbf{g} \text{ is maximal s.t. } \mathbf{g} \vee \mathbf{b} \not\geq \mathbf{c}\}.$$
$$\mathbf{SW}(\mathbf{r},\mathbf{p},\mathbf{q}) = \{\mathbf{g} \leq \mathbf{r} : \mathbf{g} \text{ is minimal s.t. } \mathbf{g} \vee \mathbf{p} \geq \mathbf{q}\}.$$

定理 6.3 (Harrington-Shelah[2]) $\mathcal{R}$ の一階理論 $Th(\mathcal{R})$ は決定不可能．

証明 (スケッチ) Harrington-Shelah による証明および Harrington-Slaman によるその簡易証明は，無限ハーリントン-シェラー集合の存在を示し Δ^0_2 半順序の理論を $\mathcal{R}$ 内でコードすることによる．詳細は Harrington-Shelah[2] を参照． □

定理 6.4 (Harrington-Slaman) $\mathcal{R}$ の一階理論 $Th(\mathcal{R})$ は算術の一階理論をコード可能．特に $Th(\mathcal{R})$ は決定不可能であり，次数 $\mathbf{0}^{(\omega)}$．

証明 (スケッチ) 証明の方針は無限スレイマン-ウディン集合の存在を示し，それを用いて $\mathcal{R}$ 内で算術をコードすることである．証明は Nies-Shore-Slaman[7] を参照． □

定理 6.5 (Lempp-Nies-Slaman[4]) $\mathcal{R}(\leq)$ の $\forall\exists\forall$-理論は決定不可能．

証明 (スケッチ) ハーリントン-シェラーの方法を用いて無限 2 部グラフの理論を $\mathcal{R}$ 内にコードする．証明は Lempp-Nies-Slaman[4] を参照． □

定理 6.6 (Miller-Nies-Shore[6]) $\mathcal{R}(\leq,\vee,\wedge)$ の $\forall\exists$-理論は決定不可能．

証明 (スケッチ) この証明は，ハーリントン-シェラー集合の存在とスレイマン-ウディン集合の存在を組み合わせることにより行われる．これらの集合の定義には若干の量化が必要であるが，両方を同時に満たせるならば，量化を削ることができる．このようにして，コードに使うベースの複雑さを下げることにより，決定不可能性のための量化の条件を引き下げている．証明は Miller-Nies-Shore[6] を参照． □

7 次数の理論の決定可能性と決定不可能性の狭間

	$\mathcal{R}$	$\mathcal{D}(\leq \mathbf{0}')$	$\mathcal{D}$	$\mathcal{P}_w, \mathcal{P}_s$
$\exists(\leq .\vee)$	○ [サックス,1963]	○ [クリーネ-ポスト,1954]	○ [クリーネ-ポスト,1954]	○ [ビンズ,2003]
$\forall\exists(\leq)$	?	○ [ラーマン-ショア,1988]	○ [ショア,1978]	?
$\forall\exists(\leq,\vee)$	?	?	○ [ジョクシュ他,1993]	?
$\forall\exists\forall(\leq,\vee)$	× [レンプ他,1998]	× [ラーマン,1983]	× [シュメール]	?
$\exists(\leq,\vee,\wedge)$	?	○	○	○ [ビンズ,2003]
$\forall\exists(\leq,\vee,\wedge)$	× [ミラー他,2004]	× [ラーマン,1983]	× [ミラー他,2004]	?

上の表のように $\mathcal{R}$ の $\forall\exists$-理論の決定可能性は未解決であるが， $\forall\exists$-理論の断片である埋め込みの拡張問題については Slaman と Soare が解決している．

定理 7.1 (Slaman-Soare[10]) $\mathcal{R}$ への（上半束の）埋め込みの拡張問題は決定可能．

参考文献

[1] S. Binns, A splitting theorem for the Medvedev and Muchinik lattices, *Mathematical Logic Quarterly* **49**(2003), 327-335.

[2] L. Harrington and S. Shelah, The undecidability of the recursively enumerable degrees, *Bulletin of the American Mathematical Society* **6**(1982), 79-80.

[3] S. Lempp, Priority arguments in computability theory, model theory, and complexity theory, http://www.math.wisc.edu/ lempp/papers/prio.pdf

[4] S. Lempp, A. Nies and T. A. Slaman, The Π_3-theory of the computably enumerable Turing degrees is undecidable, *Transactions of the American Mathematical Society* **350**(1998), 2719-2736.

[5] M. Lerman, A Framework for Priority Arguments, http://www.math.uconn.edu/ lerman/GFposet.pdf

[6] R. G. Miller, A. O. Nies and R. A. Shore, The $\forall\exists$-theory of $\mathcal{R}(\leq, \vee, \wedge)$ is undecidable, *Transactions of the American Mathematical Society* **356**(2004), 3025-3067.

[7] A. Nies, R. A. Shore and T. A. Slaman, Interpretability and definability in the recursively enumerable degrees, *Proceedings of the London Mathematical Society* (3) **77**(1998), 241-291.

[8] P. G. Odifreddi, *Classical Recursion Theory, Volume II,* North Holland - Elsevier, 1999.

[9] R. A. Shore, Degree Structures: Local and Global Investigations, *Bulletin of Symbolic Logic* **12**(2006), 369-389.

[10] T. A. Slaman and R. I. Soare, Extension of embeddings in the recursively enumerable degrees, *Annals of Mathematics* **153**(2001), 1-43.

[11] T. A. Slaman and W. H. Woodin, Definability in the Turing degrees, *Illinois Journal of Mathematics* **30**(1986), 320-334.

[12] R. I. Soare, *Recursively Enumerable Sets and Degrees,* Perspectives in Mathematical Logic. Springer-Verlag, 1987.