ディジタル生命における命令コード解釈の進化

Evolution of Operation Code Interpretation in a Digital Life System

名古屋大学 大学院人間情報学研究科　　○永田あずみ
名古屋大学 大学院情報科学研究科　　有田 隆也
○Azumi Nagata　Graduate School of Human Informatics, Nagoya University
Takaya Arita　Graduate School of Information Science, Nagoya University

Abstract: The origin of the meaning of genetic codes is one of the most important topics in evolutionary biology. This paper focuses on the relationship between code and operation in the framework of a digital life system so as to approach to the origin of the interpretation of the genetic codes. There are two levels of evolution in our model: one is evolution of mapping between code and operation and the other is evolution of program codes. Each mapping specifies a machine architecture, on which an experiment on emergence of digital life is conducted, and takes inverse number of generations required to generate a self-replicating program as its fitness value. The results of the simulation have shown that there is a selective pressure for pairs of codes which have specific relationships based on the locality in code space or the mechanism of template addressing. It has been also shown that the size of the operation set reduces and at the same time the fitness becomes higher in the course of mapping evolution.

１　はじめに

人工生命研究の中心的なトピックスにいわゆるディジタル生命がある．ディジタル生命とは，仮想計算機内のそれぞれ生物個体を表わすプログラムの集合体が，突然変異や淘汰のメカニズムを導入することで，生命に特有な現象を創発する系であり，生命の起源や進化に関わる様々なテーマを構成論的に研究することができる[1]．現実の生命現象を忠実に模擬するシミュレーションではなく，生命の本質を仮想計算機上に創り出すという意思がディジタル生命という言葉には込められており，近年，ディジタル生命を用いた研究の成果は純粋な自然科学のジャーナルでも認知されるようになってきた．

ディジタル生命のパイオニア的研究が T. Ray による Tierra [8]である．Tierra では，確保したメモリ領域に自分自身の命令コードを順にコピーしそれを切り離すことによって自己複製を実現するプログラムを先祖プログラムとして実行開始すると，突然変異や淘汰のメカニズムが働いてオープンエンドな進化が起こり，より効率的なプログラムに進化したり，近隣プログラムの一部を利用したり，利用し返したりするような複雑な「生態系」が創発した．

本研究は，ディジタル生命の世界で，生命の起源，特に，現代生物学における重要なテーマでもある遺伝コード解釈の起源の問題に焦点を合わせる．具体的には，A, C, G, T の 4 種の文字による TAG や CAT のような 3 つ組 64 種類の遺伝暗号とそれによって表現される 20 種類のアミノ酸との対応関係の成り立ちやそこにおける冗長性（複数の遺伝暗号が同一のアミノ酸を表現していること）などの問題を，ディジタル生命における命令コードと処理内容の対応関係の問題として捉える．

松崎らは進化可能性を向上させるために Tierra の各命令をマイクロ命令で書き直した上で，そのマイクロ命令による記述自体も各個体に含ませることにより，プログラムだけでなく，マシン構造も進化させている[7]．寄生個体など多様な個体が発生することが実験により示された．また，杉浦らは，Tierra を等価な書き換え系で表現した上で，文字列と書き換え規則から構成される個体を進化させている[6]. 実験により, Tierra の初期世代の個体が，両者の共進化により，より体長が小さく，また複製に要するステップ数の小さい個体が出現することを示している．

本研究では，自己複製プログラムの存在を前提とせず，自己複製プログラムの出現における遺伝コード解釈の進化を対象とし，命令コードからその処理内容へのマッピングの進化を検討する．人手による自己複製プログラムなしに自己複製プログラムが発生する現象を対象としたディジタル生命モデルに N. Pargellis による Amoeba [2][3][4]がある．彼は，仮想計算機の命令セットを単純化した上で，初期状態ではランダムにつなぎあわされた命令コード列である集団から自己複製プログラムを出現させることに成功した．

本モデルでは，命令コードと処理内容のマッピングそのものを個体として表現し進化させる．そのマッピングが規定する仮想計算機を用い，Amoeba に準じたディジタル生命の出現実験を繰り返し，自己複製プログラムが出現するまでの世代数を基にマッピングの適応度を評価する．プログラムとマッピングの両者を合わせて一つの個体として同レベルで表現し同時に進化させる手法も考えられるが，マッピングの進化に焦点を絞ること，また，ディジタル生命系の設計論に関する知見を得たいことから，このような進化手法を採用

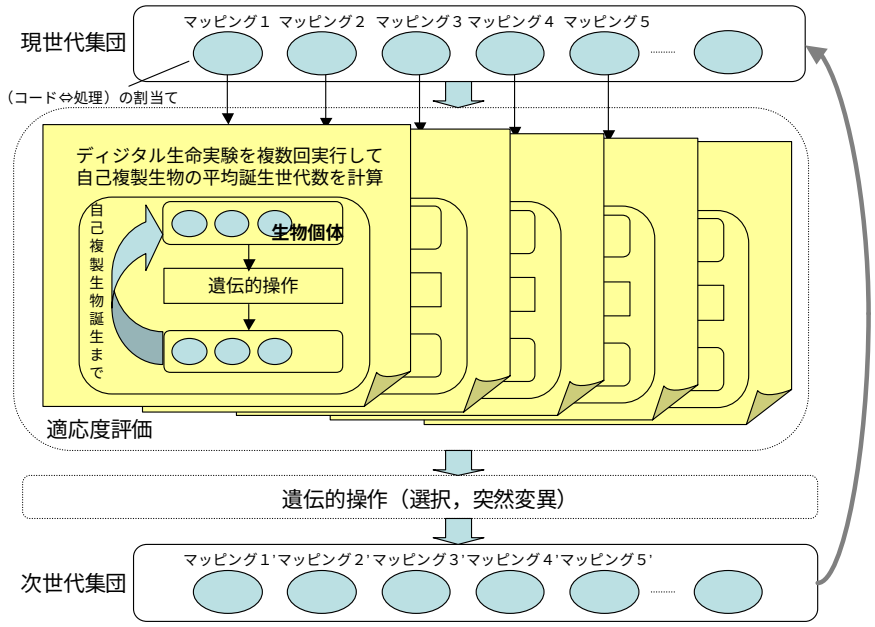

Fig. 1: モデルの構成

0000	CALL
0001	JMPB
0010	BEQA
0011	COPY
0100	IFAL
0101	ADRF
0110	IFAG
0111	RETN
1000	DIVD
1001	PYOP
1010	JMPF
1011	ADRB
1100	MALL
1101	RESA
1110	RESB
1111	PNOP

Fig. 2: マッピングの表現

した.

Pargellis も Amoeba を拡張して，自己複製プログラムの誕生，及び進化における命令コードの自己組織化を検討している[5]. そのモデルでは，64 個の命令コードと 32 個の処理内容を用意しており，プログラム進化の中にマッピング進化的な要素を加えたものである. 実験の結果，割り当てが進化することにより，自己複製プログラムが出現する確率が高まることや，命令コードが自己複製プログラムが誕生するまでは，コピーやメモリ割付に，自己複製プログラム出現後はコピーの処理に命令コードが極めて高い確率でマッピングされることを示した．ただし，このモデルは，各個体内に独立したデータとしてのマッピングは存在せず，単に突然変異で処理内容が書き換わり（同一個体内で同一命令コードが別の処理内容に解釈されうる），また，命令コードから処理内容へマッピングする確率を「割り当て行列」として集中的に保持し，うまく子孫を残すとその確率を増加させるというメカニズムを採用している．このような意図的なメカニズムを採用した理由について，自己複製生物が誕生する確率が低いからとしているが，これの採用により，必要な命令が除去されてしまうので，「割り当て行列」を使ってランダムに作られた生物を毎世代投入している．また，突然変異が起きて処理内容が変わる際に全処理内容からランダムに選ぶため，生体における３つ組によって表現される命令コード間に存在するような局所性を検討対象にできない.

2　モデル

Fig. 1 に本モデル全体の構成を示す. 命令コードと処理内容の対応関係を表す Fig. 2 に示すようなマッピングそれぞれを個体（潜在的な種とみなしうる）して集団を進化させる．進化のループは２つあり，外側のループはマッピングの進化に相当し，内側のループはプログラムの進化に相当する. プログラムの進化では，各マッピングが規定する仮想計算機を用いて，Amoeba に準じたディジタル生命実験を繰り返し，自己複製プログラムが出現するまでの世代数の平均の逆数をそれぞれのマッピングの適応度とする.

2.1　プログラムの進化

2.1.1 世界

世界は 22×80 のトーラス状２次元グリッド構造をしている（Fig. 3）. 各グリッドには命令数最大 30 個までの個体（プログラム）を一個だけ格納できる．生物は全体で 1000 個まで存在できる. 仮想計算機が実行できる処理内容の一覧を Table 1 に示す（Amoeba と同様）. 各生物は，プログラムカウンタと４つのレジスタ（ax：コピーレジスタ，bx：サイズレジスタ，cx：命令コードレジスタ，dx：アドレスレジスタ）を持つ. 初期状態で， ax=1，bx=生物のサイズ，cx=空，dx=1 である.

アドレッシングには，Tierra でも類似のものが使われているテンプレートアドレッシングを採用する．分岐命令を実行して別のプログラムコード部分の実行にジャンプする場合，一般の計算機では，アドレスを直接指定したり，現在の実行アドレスとの相対的な位置を指定して，目的のアドレス位置に飛ぶ方式などが用意されているが，これは，指定したコードをメモリ上で探し，一致したところにジャンプするものである. たとえば，PNOP の実行により，PNOP の次にある命令コードの全ビットを反転し，それを cx に格納する. その後，分岐命令を実行すると cx に格納された命令コードをメモリ上で探索し，その命令の実行に移る．もし，自グリッド内で探索できなかった場合は，距離３の近隣グリッド（最大近傍 24 グリッド）のプログラムを探索し，そこにジャンプして実行する. 本論文では，

ある命令コードの全ビットを反転した命令コードを**補コード**と呼ぶ（塩基対における対応関係を想定した設定である）. Fig. 2 の例では, BEQA の補コードは RESA である.

Table 1: 処理内容の一覧

子の複製	MALL	子の生物のために領域を確保.
	COPY	親の ax 番目の命令を子の同位置にコピーし．ax を+1.
	DIVD	子を切り離し，ポインタ等を設定して起動.
補コード格納	PNOP	次命令の補コードを cx に格納，その命令をスキップ.
	PYOP	次命令の補コードを cx に格納, その命令を実行.
分岐	JMPB	cx の命令を後方に探しジャンプ，cx が空なら先頭へ.
	JMPF	cx の命令を前方に探しジャンプ，cx が空なら最後へ.
	CALL	cx の命令を探しコール（JMPF，JMPB），次アドレスを dx に格納.
	RETN	dx のアドレスにリターン，空ならば先頭にジャンプ.
レジスタ設定	RESB	bx に生物サイズを格納.
	RESA	ax＝1
	BEQA	bx＝ax
	ADRF	cx の命令を前方に探しそのアドレスを bx に, cx が空なら生物サイズを格納.
	ADRB	cx の命令を後方に探しそのアドレスを ax に, cx が空なら 1 を格納.
条件	IFAG	ax＞bx ならば次命令をスキップ.
	IFAL	ax≦bx ならば次命令をスキップ.

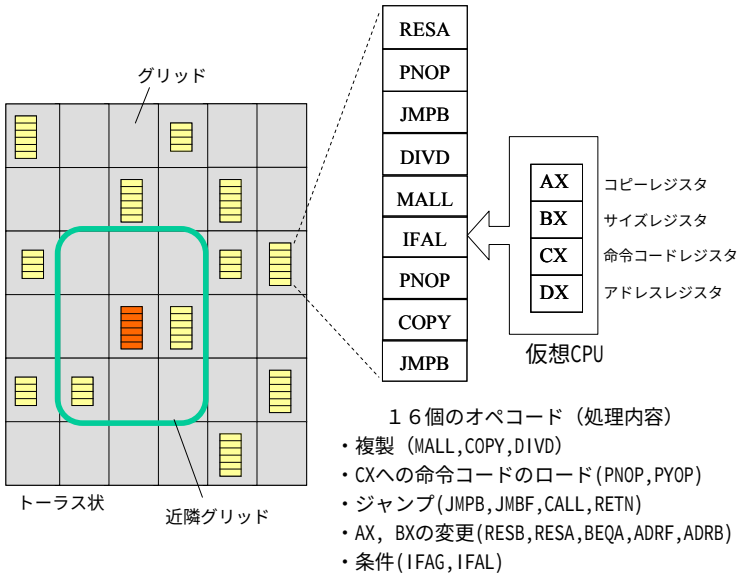

Fig. 3: プログラム進化の世界

2.1.2 進化

各世代の処理の流れを Fig. 4 に示す．まず，1 から 25 の範囲でランダムに命令コード列が作られた生物 500 個体が，2 次元平面上のランダムな位置に配置されて，実行が開始する．順に各生物でその生物の命令コード数に比例した数に 0.8 をべき乗したタイムスライスの間，CPU によって命令実行がなされる．なお，このタイムスライスは, 1) 領域確保前にコピー命令実行, 2) コールなしでリターン, 3) cx が空でジャンプまたはサーチ, 4) 領域を 2 度確保, 5) コピー元が空でコピー, 6) ジャンプ先発見できず，の６つの場合に，残りのタイムスライスの 3 分の 1 を失う．また，1) 分岐命令なし，2) 大きさ０の子の切り離し，3) 最後の命令でコール，の３つの場合に，残りの全タイムスライスを失う．1) の分岐失敗は分岐命令がなく命令最後尾に達した場合であり，その場合，距離３グリッド以内に配置されている近隣生物の先頭の命令に飛び，代わりに飛び先の生物が残りのタイムスライスの間，実行する．

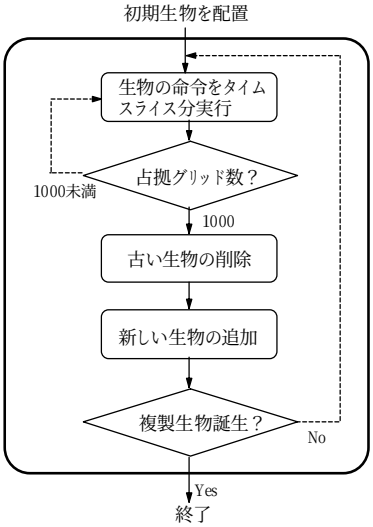

Fig. 4: 処理の流れ

生体外進化として有名な，大腸菌に感染するＱβウィルスがＱβレプリカーゼを酵素として，鋳型 RNA から RNA をコピーする[9]のと同様に，次のような４つのフェーズで自己複製は実現する：初期化）数値レジスタ, プログラムカウンタ, メモリブロックを子のために確保，複製）子のメモリ空間に親の命令コードをコピー，分離）子を分離して初期化，リセット）プログラムカウンタをスタート点に戻す（これがなくても一回は自己複製可能）．なお，MALL でメモリブロックを確保する際は，位置に関係なく空いているグリッドのいずれかがランダムに選ばれる．

1000 個のグリッドが占められたところで 1 世代の終了である．リーパが全体の 1/3 の生物を古いものから消す．その際切り離される前の未完成の生物も一緒に消される．その後，1 個から 25 個までのランダムな命令コード列からなる生物が 50 個ランダムな位置に配置される．

生物が複製されたときに， Amoeba と同様に個体単位の確率 10%で突然変異が起こる．そのうち，置換，削除，挿入が 60%，20%，20%である．Amoeba では，突然変異は等確率でランダムに命令コードが変異する

ものであった．塩基配列の遺伝コードでは，突然変異によって他の任意のアミノ酸に同等に変わるということではなく，塩基配列とアミノ酸の関係には局所性があり，その局所性が何らかの適応性を持つことが考えられるため，本研究では突然変異においてはコード４ビット中のどれか 1 ビットが反転するものとした．つまり，命令コードは突然変異でハミング距離１の命令コードに変わる．

2.2 マッピングの進化

集団は 20 個体（マッピング）で構成される．各個体は 16 個の命令コードをそれぞれ 16 個の処理内容のいずれかに対応づける．各個体の適応度を測るために，それぞれが規定するマッピングで独立に 2.1 で述べたプログラムの進化実験を行う．適応度は自己複製をする生物が初めて出現するまでの世代数の逆数とし，各個体ごとに５回そのような進化実験を行い，平均をとる．最大適応度の個体はエリートとして次世代に残し，19 個の個体はルーレット選択により適応度に応じた確率で選択する．

マッピングの突然変異としては，入れ替えとランダムの 2 種類がある．入れ替え突然変異は，命令コードと処理内容の対応関係を２つランダムに選び，それらの処理内容どうしを交換するものである．ランダム突然変異は，命令コードと処理内容の対応関係をひとつランダムに選び，その処理内容をランダムな処理内容に置き換えるものである．

３　実験

3.1 冗長性なしの進化

本実験では，命令コードのマッピングにおける局所性の影響，つまり，命令コード表現における処理内容の配置の局所的な影響を検討するため，命令コードと処理内容の関係を一対一対応に保持したまま進化させた．したがって，マッピングの進化における突然変異は入れ替えのみとなる．突然変異率は命令コードあたり 1.875%とした．

Fig. 5 に 100 世代までの個体の平均適応度の推移を示す．初期世代では適応度が 0.006 程度であるが，早いうちに適応度が上がり，0.008 程度で振動することがわかる．15 世代でもっとも高い適応度を記録している．この個体を取り出し，比較のために Amoeba で採用されている設定との両者を 100 回ずつプログラム進化の実験を行い，適応度の平均を算出した結果を Table 2 に示す．最良個体の適応度が約 2 割も高いことがわかり，命令コードのマッピングにおける局所性の影響が明確に示された．

このような適応度の差をもたらす代表的な原因として，補コードに基づくテンプレートアドレッシングが考えられる．補コードによるアドレッシングをうまく利用することによりコンパクトで，タイムスライスへのペナルティのない自己複製プログラムが構成可能であるマッピングほど，そのような自己複製生物の探索問題は容易となり，適応度が高くなると考えられるからである．

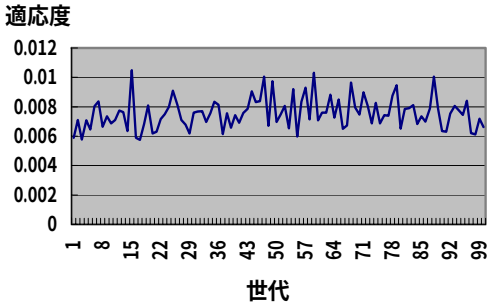

Fig. 5 冗長性なしの進化

Table 2 : Amoeba 設定と最高適応度個体の適応度
（カッコ内は自己複製生物発生世代数）

Amoeba	$5.06 * 10^{-3}$ （198）
最良個体	$6.11 * 10^{-3}$ （164）

Table 3: 上位 100 個体の補コード関係の頻度

	PNOP	MALL	COPY	DIVD	JMPB	CALL	RETN	JMPF	IFAG	PYOP	RESB	IFAL	ADRF	RESA	BEQA	ADRB
PNOP		4	2	1	0	56	3	6	0	7	1	3	0	1	4	7
MALL	4		3	5	0	0	6	21	2	7	2	20	3	14	5	8
COPY	2	3		4	2	0	3	0	46	3	0	0	4	5	22	6
DIVD	1	10	4		3	7	52	0	1	0	0	16	5	4	1	1
JMPB	0	0	2	3		9	9	4	0	23	8	14	11	10	0	7
CALL	56	0	0	7	9		5	0	8	0	3	5	1	5	1	0
RETN	3	12	3	52	9	5		9	0	3	2	2	0	2	1	3
JMPF	6	42	0	0	4	0	5		5	3	3	1	3	0	3	42
IFAG	0	4	46	1	0	8	4	5		10	19	0	3	1	3	2
PYOP	7	14	3	0	23	0	3	3	10		14	4	9	4	13	0
RESB	1	4	0	0	8	3	2	3	19	14		10	16	17	3	2
IFAL	8	40	0	16	14	5	2	1	0	4	10		5	1	2	12
ADRF	0	6	4	5	11	1	0	3	3	9	16	5		13	26	1
RESA	1	28	5	4	10	5	2	0	1	4	17	1	13		15	8
BEQA	4	10	22	1	0	1	1	3	3	13	3	2	26	15		1
ADRB	7	8	6	1	7	0	3	42	2	0	2	12	1	8	1	

そこで，本実験全体を通じての適応度の上位 100 個体のマッピングに関して，処理内容と，補コードの関係にある処理内容との相関を調べた（Table 3）．同表は各処理内容に対して補コード関係となっている処理内容の数をカウントしたものである．同表より，補コード関係には大きな偏りが見られることがわかる．たとえば，COPY と IFAG が補コード関係になっている個体は半数近い 46 個あるが，そのようなマッピングで誕生したのは，たとえば以下のようなコンパクトな生物である．まず，PYOP で IFAG の補コードである COPY が cx にロードされる．これが，JMPB の飛び先として

好都合となり，タイムペナルティなく複製が完了（ax＞bx になる）までループが実行され、DIVD で複製が完了する．

```
1011    MALL
1001    COPY
0000    PYOP
0110    IFAG
1101    JMPB
0011    DIVD
```

3.2 冗長性ありの進化

本実験では，複数の命令コードが同一の処理内容を持つことを許す進化を行った．マッピング進化の突然変異としては，ランダム突然変異，つまり，処理内容が 16 個のいずれかにランダムに変わる操作のみを採用した．突然変異率は命令コードあたり 0.3125%とした．これにより，処理内容の取捨選択の過程を観察することができる．

Fig. 6 に個体の平均適応度の 400 世代までの推移を示す．実験開始直後から大きく適応度は増加し，約 14 倍に達することがわかる．実験中での最高適応度の個体のマッピングでプログラム進化を 100 回実験したところ，平均適応度は 0.0709 であった．Table 2 と比べても 10 倍以上の適応度の向上が冗長性を許すことによってもたらされたことがわかる．

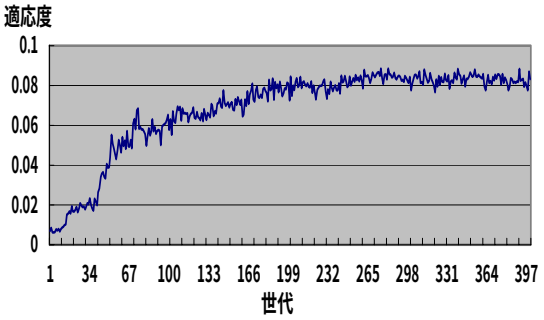

Fig. 6: 冗長性ありの進化

Fig. 7 に最終 400 世代における全 20 個体のマッピングを示す．20 個のうち 15 個が同一のマッピングに収束している．ここでの使用されている処理内容の出現頻度を Table 4 に示す. 処理内容に関する取捨選択が進み，10 個ほどに収束していることがわかる．その中でも COPY と IFAG の 2 つの出現頻度が突出して大きいことがわかる．これは，Pargellis による実験結果と同様である．

Fig. 7 の 15 個体が共有したマッピングに基づくプログラム進化の結果，生ずる生物の例を以下に示す．COPY と IFAG の補コード関係が補コードに基づくアドレッシングに利用されていることがわかる．

```
1001    MALL
1101    COPY
1110    IFAG
0110    COPY
0111    PYOP
0010    IFAG
0011    JMPB
1100    DIVD
```

0000	COPY	COPY	COPY	COPY
0001	IFAG	RESB	PNOP	IFAG
0010	IFAG	RESB	IFAG	IFAG
0011	JMPB	JMPB	JMPB	JMPB
0100	IFAG	IFAG	IFAG	IFAG
0101	ADRF	ADRF	ADRF	ADRF
0110	COPY	COPY	COPY	COPY
0111	PYOP	PYOP	PYOP	PYOP
1000	RETN	RETN	RETN	RETN
1001	MALL	MALL	MALL	MALL
1010	PNOP	PNOP	PNOP	PNOP
1011	IFAL	IFAL	IFAL	IFAL
1100	DIVD	DIVD	DIVD	DIVD
1101	COPY	COPY	COPY	COPY
1110	IFAG	IFAG	IFAG	IFAG
1111	COPY	COPY	COPY	IFAL
個数	15	3	1	1

Fig. 7: 400 世代におけるマッピング

Table 4: 400 世代における処理内容の出現頻度

COPY	24.7%
IFAG	22.8%
PNOP	6.6%
IFAL	6.6%
MALL	6.3%
DIVD	6.3%
JMPB	6.3%
ADRF	6.3%
PYOP	6.3%
RETN	6.3%
RESB	1.9%

処理内容が絞られていく際の（3.1 の実験で調べたような）マッピングにおける局所性の影響に関して調べると，以下のような興味深い傾向が見られた．それは，進化の中期までは，同じ処理内容をハミング距離 1（突然変異 1 回で移る距離）の近傍に配置する傾向が強まるが，後期になるとその傾向は弱められるということである．Fig. 8 左図は，中期（74 世代）における典型的なマッピングであり，右図は Fig. 7 に示した 400 世代に多数を占めたマッピングである．同図では，ハミング距離 1 の位置に同じ処理内容を割り振っているグループの処理内容は斜体で表わしている．左図のマッ

ピングでは同じ処理内容は１つを除いてすべてハミング距離１の位置に配置していることがわかる．そうすることで，突然変異率の影響を実質的に弱め，重要な命令コードを保護するという意味があると考えられる．ただし，次第にこの関係が弱くなる理由としては，補コードに基づくアドレッシングを確実に行う効果などのほうが強い（が，その関係を探索するのに時間がかかる）ということが考えられよう．

0000	*COPY*
0001	*COPY*
0010	*PNOP*
0011	JMPB
0100	IFAG
0101	ADRF
0110	RETN
0111	PYOP
1000	MALL
1001	IFAG
1010	*PNOP*
1011	IFAL
1100	DIVD
1101	*COPY*
1110	*PNOP*
1111	*COPY*

0000	COPY
0001	IFAG
0010	IFAG
0011	JMPB
0100	IFAG
0101	ADRF
0110	COPY
0111	PYOP
1000	RETN
1001	MALL
1010	PNOP
1011	IFAL
1100	DIVD
1101	*COPY*
1110	IFAG
1111	*COPY*

Fig. 8: 74 世代と 400 世代の典型的マッピング

４　おわりに

本論文では，自己複製プログラムの誕生における遺伝コード解釈の進化を対象とし，命令コードから処理内容へのマッピングの進化を検討した．自己複製プログラムが出現した後に関しては，その系には明示的，あるいは非明示的に選択圧が生じて適応進化が期待できるが，出現以前は事情が異なる．進化実験の結果，補コードに基づくアドレッシングの影響で特定の命令コード対の配置が自己複製プログラム出現に関して適応的であること，命令セットが縮小することにより適応度が飛躍的に増大すること，その際に重要な処理内容に対して実質的に突然変異を弱めるようなマッピングが生ずることなどが示された．

実験結果の詳細な解析が今後の課題である．たとえば，自己複製は自分のプログラムだけでなく，たまたま近所に存在する個体のプログラムを利用しても可能である．実験において，そのようなことが起こっている頻度は必ずしも多くないと考えられるが，厳密な解析が望まれる．本研究を進展させることにより，ディジタル生命システムの設計に寄与するだけでなく，遺伝コードの解釈の起源に関わる知見を生み出しうると考えている．

参考文献

[1] 有田隆也, "*人工生命*", 医学出版, 2002.

[2] A. N. Pargellis, "The Spontaneous Generation of Digital 'Life'", *Physica*, D91, 86-96, 1996.

[3] A. N. Pargellis, "The Evolution of Self-replicating Computer Organisms", *Physica*, D98, 111-127, 1996.

[4] A. N. Pargellis, "Digital Life Behavior in the Amoeba World", *Artificial Life*, 7 (1), 63-75, 2001.

[5] A. N. Pargellis, "Self-organizing Genetic Codes and the Emergence of Digital Life", *Complexity*, 8 (4), 69-78, 2003.

[6] K. Sugiura, H. Suzuki, T. Shiose, H. Kawakami and O. Katai, "Evolution of Rewriting Rule Sets Using String-based Tierra", *Proc. of 7th European Conference on Artificial Life*, 69-77, 2003.

[7] S. Matsuzaki, H. Suzuki and M. Osano, "An Approach to Describe the Tierra Instruction Set Using Microoperations: the First Result", *Proc. of 7th European Conference on Artificial Life*, 357-366, 2003.

[8] T. S. Ray, "An Approach to the Synthesis of Life", *Artificial Life II*, 371-408, 1992.

[9] M. Eigen, P. Schuster, W. Gardiner and R. Winkler -Oswatitsch, "The Origin of Genetic Information", *Scientific American*, 244 (4), 78-94, 1981.