

博士論文

色付き受理状態を持つオートマタとそれらの
非混色性問題に関する研究

(A Study on Automata with Colored Accepting States and Their Unmixedness Problems)

2022年11月

高橋 芳明

山口大学大学院創成科学研究科

A Study on Automata with Colored Accepting States and Their Unmixedness Problems

Abstract

Some textbooks of formal languages and automata theory implicitly state the structural equality of the binary n -dimensional de Bruijn graph and the state diagram of minimum state deterministic finite automaton which accepts regular language $(0+1)^*1(0+1)^{n-1}$. Although the isomorphism in binary case is relatively easy to prove, it is desirable to rigorously prove such an isomorphism in general k -ary case. To achieve this purpose, the author introduces a new computational model, called “colored finite automata (CFA),” and give a certain characterization of the general k -ary de Bruijn graphs by regular languages.

The second purpose of this study is to investigate the potential of this automaton with multi-colored accepting states. By the way, when CFA is nondeterministic (NCFA), it is desirable that the colors of accepting states are unmixed (i.e., there are no inputs that are accepted with differently colored accepting states) in order to pursuit the accurate identification. Thus, the author proposes the three decision problems (Unmixedness Verification problem, Unmixedness Partitioning problem, and Unmixedness Extension problem) concerning unmixedness and show that UV, UP, and UE problems are shown to be $NLOG$ -complete, P, and NP -complete, respectively. The author also illustrates the applications of colored finite automata, e.g. to existing regular expression engines and model checking tools for the purpose of improvement of their efficiency and conveniency.

Next, the author introduces “colored pushdown automaton (CPDA)” which is an ordinary pushdown automaton with colored accepting states. It is shown that while the computational complexity of the above-mentioned UV, UP, and UE problems of CPDA are all undecidable, restriction of CPDAs to unambiguous

ones simplifies some problems of them to the permanently true problems.

In this way, the concept of colored accepting states can be applied to a wide range of automata that have a set of accepting states and expected to be useful in a wide range of theoretical and practical field of automata applications in the future.

色付き受理状態を持つオートマタとそれらの非混色性 問題に関する研究

概要

オートマトン理論と形式言語理論のいくつかの文献では、2進 n 次元ドブリュイニンググラフと、正規言語 $(0+1)^*1(0+1)^{n-1}$ を受理する最小状態数の決定性有限オートマトン $D_{2,n}$ の状態遷移図が同型であることを暗に示している。この2進の場合の同型性は比較的容易に証明されるが、この結果を k 進に拡張した場合の同型性を厳密に証明することが望まれる。本研究ではこの第1の目的を達成するために、新たに“色付き有限オートマトン (CFA)”という計算モデルを導入し、正規言語によって一般の k 進ドブリュイニンググラフに対するある種の特徴づけを与える。

本研究の第2の目的は、受理状態を多色化することにより入力文字列を多種類に類別可能にしたこのオートマトンの性質を更に考察することにある。ところで、CFAが非決定性(NCFA)の場合には、的確な識別を可能にする為には受理状態の色が混色していない（すなわち、異なる2色以上の受理状態で受理されるような入力が存在しない）状況が望まれる。そこで、非混色性に関する計算量理論上の三つの決定問題（非混色性検証問題、非混色分割問題、非混色拡張問題）を新たに提案し、それぞれの計算複雑さが $NLOG$ 完全, P , ならびに NP 完全であることを明らかにする。なお、色付き有限オートマトンの正規表現エンジンやモデル検査への応用も例示する。

次に、色付き受理の概念をプッシュダウン・オートマトンに適用した“色付きプッシュダウン・オートマトン (CPDA)”を新たに導入し、上述の非混色性判定問題すべてが決定不能となる一方、対象を非曖昧なオートマトンに限定すると一部の問題が常に真な問題に容易化することを示す。

このように、色付き受理の概念は受理状態集合を持つような任意のオートマトンモデルに適用可能であり、その理論上ならびに応用上の幅広い有用性が今後期待される。

0 目次

1	序論	1
1.1	研究の背景	1
1.2	関連研究	3
1.3	研究の目的	6
1.4	本論文の構成	11
1.5	諸定義	12
1.5.1	非決定性（決定性）有限オートマトン	14
1.5.2	ドブリュイングラフ $DB_{k,n}$	14
2	ドブリュイングラフと正規言語	
	$(0+1)^*1(0+1)^{n-1}$ を受理する DFA との関係	17
2.1	正規言語 $(0+1)^*1(0+1)^{n-1}$ を受理する有限オートマトン $D_{2,n}$	18
2.2	2進 n 次元ドブリュイングラフ $DB_{2,n}$ と $D_{2,n}$ の関係	19
2.3	k 進への一般化の試み	25
2.4	まとめ	29
3	色付き有限オートマトン	31
3.1	非決定性色付き有限オートマトン (NCFA)	31
3.2	k 進 n 次元ドブリュイングラフ $DB_{k,n}$ と $D_{k,n}$ の関係	43
3.3	NCFA の非混色性に関する決定問題	52
3.4	ε 遷移付き NCFA とその適用例	62
3.5	色付き正規表現	64
3.6	非決定性色付きビュッヒオートマトン (NCBA)	66
3.7	色付き有限オートマトンの応用	69

3.7.1	正規表現エンジンへの応用	69
3.7.2	モデル検査への応用	75
3.8	まとめ	81
4	色付きプッシュダウン・オートマトン	84
4.1	非決定性色付きプッシュダウン・オートマトン (NCPDA)	85
4.2	NCPDA の非混色性に関する決定問題	88
4.3	無曖昧な NCPDA の非混色性に関する決定問題	91
4.4	まとめ	95
5	結論	97
5.1	研究の目的毎の総括	97
5.2	まとめ	98
5.3	今後の課題	100
	参考文献	104
	A 付録	110
A.1	倍化法によるドブリュイングラフの構成	110
A.2	倍化法の状態遷移表による解釈	112
A.3	状態数最小化 DCFA の一意性	114
	謝辞	118
	本論文に関する研究業績	119

1 序論

1.1 研究の背景

1930年代に数学者のアラン・チューリングは、計算に関して現代のコンピュータと同等の能力を持つチューリング機械を考察した。プログラムと入力データが与えられた時に、そのプログラムが停止するかどうかを事前に判定するアルゴリズムが存在するかどうかという停止性問題を、チューリング機械を用いて否定的に証明した [1, 2, 16]。更には、チューリングはあらゆるプログラムを実行できる万能チューリング機械が実現可能であることを示し、現代の汎用機の発展に大きな影響を与えた。チューリング機械は、今日の計算理論を支える重要な計算モデルとなっている。なお、チューリング機械を簡略化した有限オートマトンやプッシュダウン・オートマトンも、現代の計算機科学の多くの分野において、基本的かつ不可欠な計算モデルとなっており、現在までに、オートマトンの数多くの変形版が提案され研究されてきた [16, 17, 19, 21, 23, 26, 30, 31, 36, 38]。一方、グラフ理論で扱われるドブリュイングラフ（ドブリュイン系列）は、符号理論、コンピュータネットワークのトポロジー、ゲノムのアセンブリ等の応用領域で幅広く使用されている [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]。例えば、ゲノムのアセンブリは短いDNAの断片の集まりから元の長いDNAを再構成することが目的であるが、円環ゲノムのアセンブリに関して、従来のアルゴリズムはハミルトン閉路問題に還元して解いていたので時間がかかっていた。そこで、ドブリュイングラフを用いたアセンブリが採用されている [15]。

ところで、オートマトン理論を学ぶことは、計算機の計算の原理を理解することに繋がる。AI技術、機械学習技術が益々発展していく高度情報社会の中で、上

手く高度な情報技術を使いこなすためにも、計算の原理に関連する計算の限界や計算複雑さの理論を認識した上で、情報技術を扱うことが求められる。その基礎的理論を固めるための一つの科目に位置づけられるオートマトン理論には、いくつか必ず修得すべき重要項目があり、部分集合構成法、正規表現から有限オートマトンへの変換、有限オートマトンの最小化はその代表格である。それらの中の部分集合構成法および有限オートマトンの最小化に関連し、状態数 n の非決定性有限オートマトン (NFA) を決定性有限オートマトン (DFA) に変換した際に状態数が 2^n と指数関数的に増え、且つその状態数が最小であるような、オートマトン理論の学習者にとってなじみ深い有限オートマトンが存在する。それは正規言語 $L_{2,n} = (0 + 1)^*1(0 + 1)^{n-1}$ を受理する最小状態数の決定性有限オートマトン $D_{2,n}$ であり、よく知られたオートマトン理論の教科書 [16] に登場する。オートマトン理論と形式言語理論のいくつかの文献では、2進 n 次元ドブリュイングラフ $DB_{2,n}$ と、 $L_{2,n}$ を受理する最小状態数の $D_{2,n}$ の状態遷移図が同型であることを暗に示しているが、この同型性の確かな証拠は与えられておらず、高等教育の教育者はこの同型性について自信を持って公言できない状況であった。頻出するなじみ深いオートマトンに関して、不確かな事実が存在することは、教育的観点および理論的観点からも見過ごすことができず、その同型性を明確にすることによって不明瞭な状況を解消することが望まれる。また、古典的疑問でもあるこの同型性は、グラフ理論とオートマトン理論という異なる専門分野間での強い関連性を示すものであり、その同型性の証明が2つの違う分野を繋げ、更には新しい知見の創出に繋がる可能性を秘めている。

このような状況において、文献 [44] では、2進の場合におけるこの同型性が証明された。しかしながら、この2進の結果をより高い基数 k に拡張しようとする、正規言語の k 個組み $(0 + 1 + \dots + k - 1)^*1(0 + 1 + \dots + k - 1)^{n-1}, \dots, (0 + 1 + \dots + k - 1)^*(k - 1)(0 + 1 + \dots + k - 1)^{n-1}$ で表される言語 $L_{k,n}$ を受理する非決定性有限オートマトン $N_{k,n}$ を決定性に変換した $D_{k,n}$ は状態数が最小ではなくなる。すなわち、穴埋めアルゴリズムや Hopcroft のアルゴリズムといったよく知られた

最小化アルゴリズムを使用すると、状態が削減され辺のラベルを除いて $k = 2$ の場合のオートマトン $D_{2,n}$ と同型になる。つまり、 k 進 n 次元ドブリュイングラフ $DB_{k,n}$ と $L_{k,n}$ を受理する最小状態数の決定性有限オートマトンの状態遷移図の同型性の証明は未解決のままである。

1.2 関連研究

$DB_{k,n}$ と $L_{k,n}$ を受理する最小状態数の決定性有限オートマトンの状態遷移図の同型性は、教育的観点および理論的観点から重要と言えるにも関わらず、筆者が知る限りこれまで誰も取り組んでいない。誰も取り組んでいないゆえ、この未解決な課題は追求する価値がないとは断言できない。例えば、2 進の場合の上述の同型性により、グラフ理論で用いられる構成法がオートマトン理論の概念において解釈できることを本論文の付録 A.2 で紹介している。これは異なる理論分野間での同型性により見出された知見の例であり、 k 進においてもこの同型性を証明することにより、教育的観点における重要度以外に更なる新しい知見や概念の創出が期待される所以である。

ここにおいて、2 進における最小状態決定性有限オートマトン $D_{2,n}$ とドブリュイングラフ $DB_{2,n}$ の同型対応を k 進へ拡張するための必要条件としては以下が挙げられる。

1. 通常の部分集合構成法を自然に拡張した、非決定性有限オートマトン $N_{k,n}$ を $D_{k,n}$ へ変換するような構成法が存在する。
2. 得られた $D_{k,n}$ の状態数が最小であるかどうかを論証可能である。
3. 言語 $L_{k,n}$ を受理する状態数最小の決定性有限オートマトンの一意性が保障可能である。

上記の条件1における拡張の自然さ（2進との連続性）は k 進での同型性の証明を2進の場合の自然な拡張にすることで、その蓋然性（成立の必然性）を高めるために必要である。

ところで、 k 進の場合に同型性を主張できない理由は、上述したように k 進に拡張した有限オートマトンが最小化アルゴリズムにより簡略化され、 $DB_{k,n}$ と同型でなくなるからであった。この状況を回避するためには、有限オートマトンが従来持つ受理および非受理を判断する機能から、入力文字列を2つ以上の言語に分類する機能を有限オートマトンに持たせたオートマトンにより簡略化を防ぐ必要性が生じる。そのような機能を持つオートマトンとして、Castiglione ら [20] によって導入された非決定性ムーアオートマトン (NMA) が存在する。NMA は以下のように定義されている。

- 非決定性ムーアオートマトン NMA は決定性ムーアオートマトン DMA $A = (\Sigma, \Gamma, q_0, \delta, \lambda)$ の拡張として定義される。ここに、 $\Gamma = \{\gamma_1, \dots, \gamma_k\}$ は出力記号（すなわち色）の集合、 $\delta: Q \times \Sigma \rightarrow Q$ は（必ずしも全域でない）遷移関係、 $\lambda: Q \rightarrow \Gamma$ は全域 (total) な出力関数である。NMA では、初期状態 q_0 は初期状態の集合 I に、 δ は遷移関係 $\Delta \subseteq Q \times \Sigma \times Q$ に、色付け関数 λ は部分関数に緩和される。色無しの要素が存在する ($Q - \text{dom}(\lambda) \neq \emptyset$) とき、 A は出力不完全と呼ばれる。
- 以下の2つの性質を持つ NMA をコヒーレントな NMA と言う。
 - **性質1**：初期状態から到達可能なすべての語 $v \in \Sigma^*$ に対して、 v でラベル付けされた少なくとも一つの色付きパスが存在する（色が付かない v が存在しない）。
 - **性質2**：初期状態から到達可能なすべての語 $v \in \Sigma^*$ に対して、 v でラベル付けされた全てのパスは同じ色で色付けされている。

上記性質2のみを満たす NMA を準コヒーレントな (semi-coherent) NMA と

言う。例えば、準コヒーレントな NMA のクラスはコヒーレントな NMA のクラスより計算能力が高くなる。

- 初期状態から到達可能な語からなる言語 $L(A)$ は A が DMA あるいはコヒーレントな NMA ならば各 $w \in L(A)$ に付随する一意な色 γ_i により直和分割 $\sum_{i=1}^k L_i(A)$ される（性質 1 を満たさない場合は $L(A) - \sum_{i=1}^k L_i(A) \neq \emptyset$ となり、性質 2 を満たさない場合は一意な色が定まらない）。

要するに、この NMA は形式上は順序機械理論に基づいており、有限オートマトンのような受理状態がなく、受理および非受理という概念がない。また、NMA A が表す言語 $L(A)$ は、初期状態から到達可能な全てのパスを表す文字列を表しており、有限オートマトンにおける言語の扱い、即ち初期状態から受理状態へ到達可能なパスを表す文字列全体とは遠くかけ離れている。詳しく言うと、Castiglione らのアプローチには以下の問題点がある。

- **他研究の問題点 1**：本研究においては受理言語と見なしたい、色づけされた語全体 $\bigcup_{i=1}^k L_i(A)$ の存在が無視されている。逆に、本研究の場合には非受理言語の一部であること以上の意味を見出せない到達可能な語全体が $L(A)$ として特別視されている。特に、2つのオートマトン A, B が同値であるための必要条件のひとつが $L(A) = L(B)$ である。
- **他研究の問題点 2**：非決定性有限オートマトンから決定性有限オートマトンを求める通常の部分集合構成法は、状態遷移関数に対して局所的な操作を施した結果を集約すれば自然に定まる、静的な定義と捉えるべきであるが、彼らの部分集合構成法は NMA を表す（ラベル付き有向）グラフから DMA を表すグラフへの（不要な部分グラフを生成しない）グラフ変換アルゴリズムとして与えられている。この動的な定義は、部分集合構成法で得られる決定性有限オートマトンの性質を詳しく調べることを困難にする。

従って、この NMA を用いて k 進 n 次元ドブリュイングラフ $DB_{k,n}$ と $L_{k,n}$ を受理

する最小状態数のオートマトンの状態遷移図が同型であることを証明することは難しく、この要求を満たすようなオートマトンの新たな定式化が課題として残る。

1.3 研究の目的

2進の結果を k 進に拡張することはグラフ理論では極めて自然なことであり、実際的にドブリュイングラフが応用されている3D Network on Chip [60], ゲノムのアセンブリ [49], ピアツーピアネットワークである Koorde [50] のトポロジー等では k 進のドブリュイングラフで議論がなされている。一方上述したように、 k 進における $DB_{k,n}$ と $L_{k,n}$ を受理する最小状態数の決定性有限オートマトンの状態遷移図の同型性を証明するためには、入力文字列を2つ以上の言語に分類する機能を有限オートマトンに持たせたオートマトンの必要性が生じる。それを満たすはずの既存のオートマトンである NMA は有限オートマトンと土台が違い、素直に適用できない状況である。

そこで、本研究では $DB_{k,n}$ と $L_{k,n}$ の同型性を証明する課題を解決するために、NMA のように順序機械理論ではなく従来の有限オートマトン理論に基づいた“色付きオートマトン”を提案する。この色付きオートマトンは受理状態を多色化することにより、入力文字列を2つ以上の言語に分類することができる。この色付きの概念を用いることにより、 k 進の場合つまり k 進 n 次元ドブリュイングラフ $DB_{k,n}$ と $L_{k,n}$ を受理する最小状態数の決定性色付き有限オートマトン (DCFA) $D_{k,n}$ の状態遷移図の同型性を証明することを目指す。また、非決定性有限オートマトンから決定性有限オートマトンに変換する部分集合構成法を自然に拡張した非決定性色付き有限オートマトン (NCFA) に対する部分集合構成法を導入する。

更に、この NCFA 版の部分集合構成法を用いることにより、NCFA から DCFA に変換すると共に、変換した DCFA に元々の NCFA には存在しない合成色があるかどうかを確認することすなわち準コピーレンシーと同じ概念である、NCFA の

非混色性 (Unmixedness) を明らかにすることができる。例えば、図1の色付き有限オートマトンは、 q_1 と q_2 が色付き受理状態であり、受理状態 q_1 は最後に入力文字 1 を読み込んで受理し、受理状態 q_2 は最後に入力文字 2 を読み込んで受理する。色付き有限オートマトンは、色付けされた受理状態を導入することによって、受理する言語を識別できることが特徴である。図1では、入力文字列を最後に1か2を読んだかで識別している。このように、NCFA が入力文字列を的確に識別するためには、受理状態の色が混色していないことが望まれる。即ち、色付き有限オートマトンを考察するにあたり、非混色性の概念が実用上の課題となり (図1参照)、色付きオートマトンを考察する重要性がこの非混色性に表れていると言える。

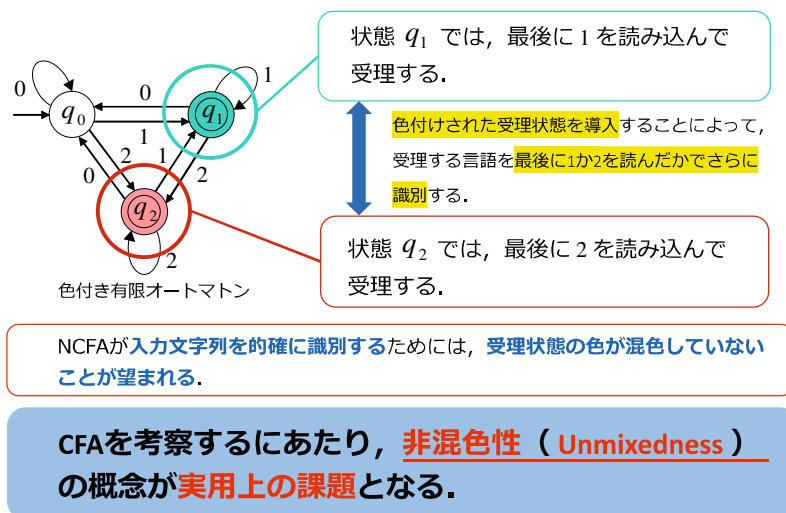


図 1: 非混色性の概念の重要性.

以上より、本研究の目的を以下の2つとする。この2つの目的は、同時並行的に取り組むものではなく、第1段階の目的を解決した後に第2段階の目的に取り組んでいくというように、段階的に設定する目的である。本論文もこの設定の流れに則り、構成されている。

- **第1段階の目的**: k 進 n 次元ドブリュイングラフ $DB_{k,n}$ と、正規言語の k 個組み $(0 + 1 + \dots + k - 1)^*1(0 + 1 + \dots + k - 1)^{n-1}, \dots, (0 + 1 + \dots + k -$

$1^*(k-1)(0+1+\dots+k-1)^{n-1}$ を受理する最小状態数の決定性色付き有限オートマトン (DCFA) $D_{k,n}$ の状態遷移図の同型性を証明することにより、ドブリュイングラフに対して正規言語によってある種の特徴づけを行うこと。

- **第2段階の目的**：2進 n 次元ドブリュイングラフと $D_{2,n}$ の同型性を k 進に拡張するための手段として得られた、受理状態を多色化することにより入力文字列を多種類に類別可能にした色付きオートマトンについて考察すること。

ここに、正規言語による有向グラフの特徴づけとは、ある特定のグラフの族が、ある特定の正規言語の族を受理する有限オートマトンの状態遷移図のグラフ構造と完全に一致することを意味する。このとき、正規言語と有限オートマトンは通常一対多で対応するので、そのような特徴づけを得るためにはグラフの族とその正規言語を受理する DFA の族の状態遷移図が構造的に同型であることを示すことのみならず、そのような DFA が一意に定まらなければならない。すなわち、各正規言語 L に対して状態数が最小な（かつ一意な）一つの DFA M が対応し、逆もまた言える。これらの間には1対1の対応関係が存在する。ただし、 M の状態遷移図を表す有向グラフの (1) 各辺には文字ラベルが付いており、(2) 初期状態として一つの頂点、(3) 受理状態集合として頂点のある集合が指定されている。従って、一つの（辺ラベル等の無い）有向グラフに対応する最小化 DFA（≡正規言語）は2つ以上存在する可能性があると言える。このような現状が在るが、本論文ではドブリュイングラフを $L_{k,n}$ によってある種の特徴づけを行うことを目指し、既知の正規言語以外での対応関係は今後の課題として位置づけることとする。

以下に本研究の特徴について述べる。ここで、関連研究の Castiglione らのアプローチには準コヒーレント（非混色）な有限オートマトンの扱いにおいて、以下の2つの問題点があると考えられる。

- **他研究における非混色性に関する問題点1**：非決定性有限オートマトンから部分集合構成法によって変換された決定性有限オートマトンの各受理状態からの出力色は、準コヒーレントな場合にのみ定義されている（それ以外の場

合は未定義である)。

- **他研究における非混色性に関する問題点 2**：非決定性有限オートマトンの準コヒーレンシー性は，決定性に変換した後にのみ確認することができる。これは最悪の場合，計算が指数関数時間かかってしまう。

他研究における非混色性に関する問題点 1 と本研究の対比を示す図 2 のように，この NCFA 版の部分集合構成法の特徴は，**他研究における非混色性に関する問題点 1** で表される準コヒーレント（非混色）な場合にのみ出力色が定義されるというものとは対照的で，色付きオートマトンのモデルは非混色かどうかに関わらず受理状態の色が定義され，NCFA から DCFA に変換後も混色状況が正しく反映されている。

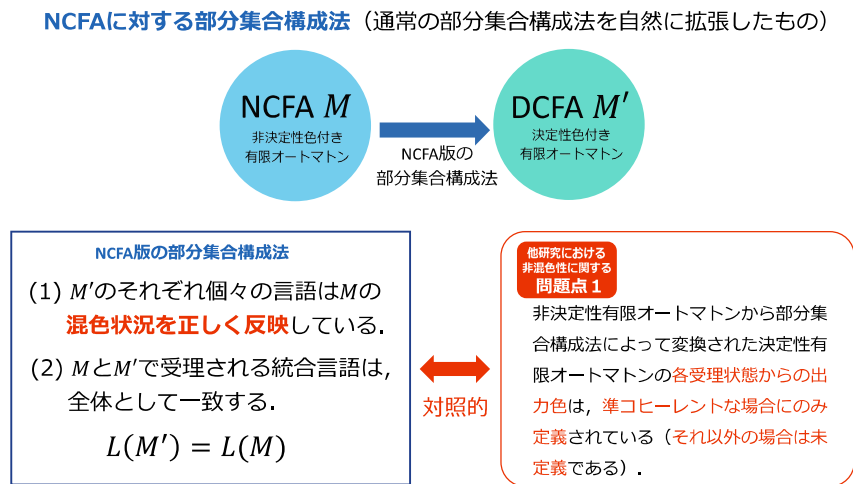


図 2: 他研究における非混色性に関する問題点 1 と本研究の対比.

また，本研究では色付き有限オートマトンを考察するにあたり，NCFA の“非混色性”に関する計算量理論上の下記 3 つの決定問題を提起し，それらの計算複雑さについて議論を行っている。

- 非混色性検証問題 (Unmixedness verification problem: UV 問題)

- 非混色分割問題 (Unmixedness partitioning problem: UP 問題)
- 非混色拡張問題 (Unmixedness extension problem: UE 問題)

非混色性検証問題 (UV 問題) は, 与えられた NCFA が同一の入力を異なる色で受理することがないかどうかを問う問題である. 他の二つの非混色分割問題 (UP 問題) と非混色拡張問題 (UE 問題) は与えられた色無し有限オートマトンを多色化しようとする際に生ずる問題であり, 前者は既存の受理状態の多色化, 後者は既存の非受理状態の多色受理状態化に関連する. 最終的にそれぞれの問題が $NLOG$ 完全, P , NP 完全であることを示しているが, UV 問題と UP 問題が P に属することは, 非決定性 CFA を決定性のものに変換せずとも多項式時間内に非混色性を確認できる確証を得たことであり, 他研究における非混色性に関する問題点 2 である準コヒーレンシー性 (混色性) は決定性に変換した後にのみ確認することができるとは, 図 3 に示すように良い意味で対照的である.

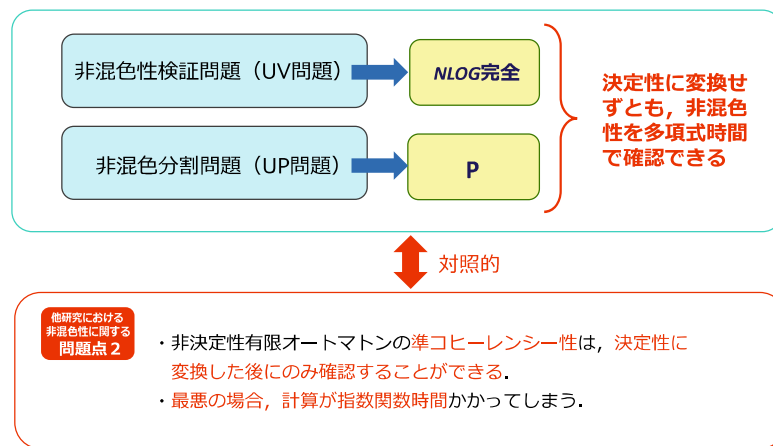


図 3: 他研究における非混色性に関する問題点 2 と本研究の対比.

ところで, 読者は既出の NMA と準コヒーレンシーという概念があるのに, なぜ NCFA と非混色性という概念を新たに持ち出す必要があるのかと疑問に思うかもしれない. 非混色性は, 自己検証型有限オートマトン (self-verifying finite automata)

[27] の前提条件である：Yes で色付けされた受理状態と No（拒否）で色付けされた状態に同時に入ることにはできない。混色性を回避するための対策としては、色の集合に束 [28] や環 [30, 29] などの特定の順序構造を予め与えておき、受理状態の色の中から最も優先度の高い色が自動的に選択されるようにし向けることが考えられる。この自己検証オートマトンは出力アルファベットが $\Gamma = \{\text{No}, \text{Yes}\}$ であるようなコヒーレントな NMA である。以下の定理が成り立つことが知られている。

定理 1 [56] 受理状態と拒否状態を持つ有限状態機械 A が与えられたとき、 A が自己検証オートマトンかどうかを問う問題は $PSPACE$ -完全である。

従って、次の事実が得られる。

事実 1 受理状態集合 F_1, F_2 を持つ有限状態機械 A が与えられたとき、 A がコヒーレントな NMA かどうかを問う問題は $PSPACE$ -困難である。

この事実 1 はコヒーレントな NMA の構成自体に困難さを伴いこの計算モデルには実用性が乏しいことを意味する。コヒーレントな NMA が“整合”しようとする相手は決定性ムーア機械 (DMA) である。DMA は全ての状態が受理状態であるような DCFA と見なせる。

1.4 本論文の構成

本論文は大きく下記 5 章により構成される。

1. 序論
2. ドブリュイングラフと正規言語 $(0+1)^*1(0+1)^{n-1}$ を受理する DFA との関係
3. 色付き有限オートマトン

4. 色付きプッシュダウン・オートマトン

5. 結論

本論文の章ごとの関連性を図4に示す.

第2章では, 色付きオートマトンを導入する背景となったドブリュイングラフの正規言語によるある種の特徴づけについて説明する.

第3章では, 非決定性色付き有限オートマトン (NCFA) の定義を与え, k 進 n 次元ドブリュイングラフ $DB_{k,n}$ と $D_{k,n}$ の状態遷移図の同型性の証明を行い, $DB_{k,n}$ を正規言語 $(0+1+\dots+k-1)^*1(0+1+\dots+k-1)^{n-1}, \dots, (0+1+\dots+k-1)^*(k-1)(0+1+\dots+k-1)^{n-1}$ により特徴づける. また NCFA の非混色性に関する三つの決定問題に関して議論を行い, 最後に, NCFA の正規表現エンジンやモデル検査への応用を例示する.

第4章では, 色付き受理状態の概念をプッシュダウン・オートマトンへ適用させた非決定性色付きプッシュダウン・オートマトン (NCPDA) の定義を与え, NCFA の場合と同様に非混色性に関する三つの決定問題に関して議論を行う. 更に, 無曖昧という制約を加えた場合の NCPDA の非混色性に関する決定問題について検討する.

最後の第5章では, 本論文の結論を与え, 色付き受理の概念は受理状態集合を持つような任意のオートマトンモデルに適用可能であることを示し, 色付きオートマトンの今後の理論上ならびに応用上の幅広い有用性について述べる.

1.5 諸定義

本節では, 必要な諸定義及び, 諸記法を与える [5, 8, 10, 11, 16, 17, 23].

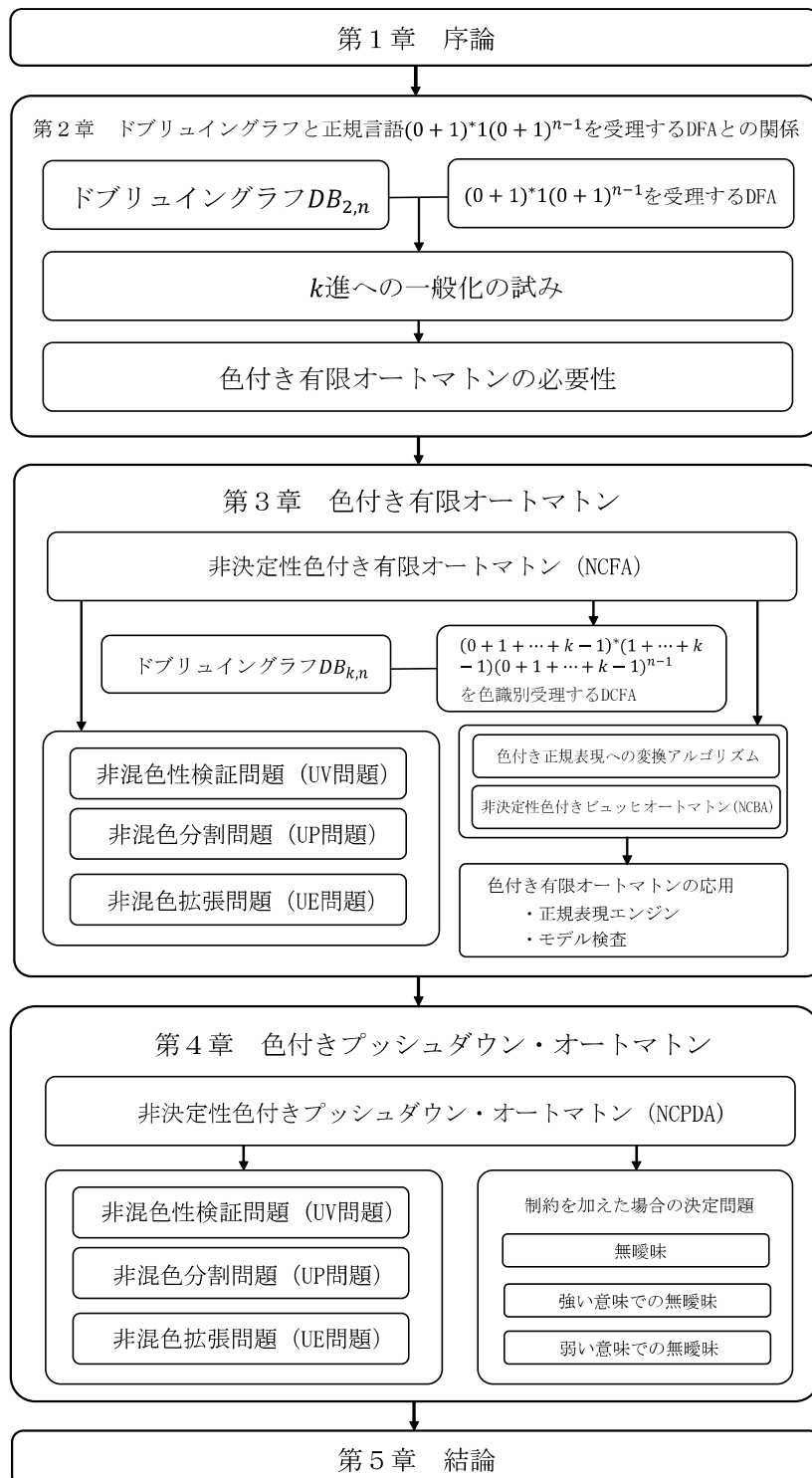


図 4: 本論文の章ごとの関連図.

1.5.1 非決定性（決定性）有限オートマトン

定義 1 [16, 17, 23] 次のように定まる 5 項組 $M = (Q, \Sigma, \delta, q_0, F)$ を非決定性有限オートマトン (nondeterministic finite automata) と呼び、NFA と略記する。

1. Q は状態の有限集合.
2. Σ は入力文字列の有限集合.
3. δ は遷移関数で, $Q \times \Sigma$ から 2^Q への関数.
4. $q_0 \in Q$ は初期状態.
5. $F \subseteq Q$ は受理状態の集合.

各 $q \in Q, a \in \Sigma$ に対して $\delta(q, a)$ が単項集合の場合, M は決定性 (deterministic) であるといい, DFA と略記する. M が初期状態 q_0 から出発し, 入力列を x を読み終えたとき, その最終状態が受理状態集合 F に入っているならば, x は M で受理されたという. M で受理される言語を $L(M) \triangleq \{x \in \Sigma^* \mid x \text{ が } M \text{ で受理される}\}$ と定義する.

事実 2 [部分集合構成法] [16, 17, 23] NFA $M = (Q, \Sigma, \delta, q_0, F)$ に対して, DFA $M' = (2^Q, \Sigma, \delta', \{q_0\}, \{S \subseteq Q \mid S \cap F \neq \emptyset\})$ とおくと, $L(M') = L(M)$. ここに, 各 $S \subseteq Q, a \in \Sigma$ に対して, $\delta'(S, a) = \bigcup_{p \in S} \delta(p, a)$.

1.5.2 ドブリュイングラフ $DB_{k,n}$

定義 2 [5, 8, 10, 11] 次のように定まる有向グラフ (V, E) を k 進 n 次元ドブリュイングラフ (k -ary n -dimensional de Bruijn graph) といい, $DB_{k,n}$ と略記する.

$$\left\{ \begin{array}{l} V = \{b_1 b_2 \cdots b_n \mid b_i \in \{0, 1, \dots, k-1\}, i = 1, \dots, n\}, \\ E = \{(b_1 b_2 \cdots b_n, b'_1 b'_2 \cdots b'_n) \mid \\ \quad b_i, b'_i \in \{0, 1, \dots, k-1\}, i = 1, \dots, n, \\ \quad b_2 = b'_1, b_3 = b'_2, \dots, b_n = b'_{n-1}\}. \end{array} \right.$$

$DB_{2,1}$ を図5に示す. 図において各辺の2進表示ラベルは, その左端 n ビットが表す点が始点であり, 右端 n ビットが表す点が終点であることを意味する. なお, $n = 1$ のときは $b_2 = b'_1, b_3 = b'_2, \dots, b_n = b'_{n-1}$ の制限がなく, 無制限となることに注意されたい. ドブリュイングラフは多段シフトレジスタの完全状態図に等しく, ドブリュイン系列はそのドブリュイングラフを一筆書き巡りすることにより求められる [3].

例 1 ドブリュイングラフの定義より, ドブリュイングラフ $DB_{2,1}(n = 1)$ を求める. $n = 1$ なので, 定義2における $b_2 = b'_1, b_3 = b'_2, \dots, b_n = b'_{n-1}$ なる制限は無視できる. よって,

$$\left\{ \begin{array}{l} V = \{b_i \mid b_i \in \{0, 1\}, i = 1\}, \\ \quad = \{0, 1\}, \\ E = \{(b_i, b'_i) \mid b_i, b'_i \in \{0, 1\}, i = 1\}, \\ \quad = \{00, 01, 10, 11\}. \end{array} \right.$$

各辺の2進表示ラベルは, その左端1ビットが表す点が始点であり, 右端1ビットが表す点が終点であることより, 各辺にラベルを付けると, 図5で表される $DB_{2,1}$ を求めることができる.

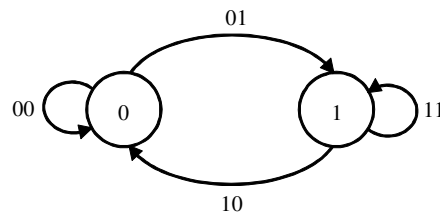


図 5: ドブリュイングラフ $DB_{2,1}$.

定義 3 [3] 1 周期中に 0 から $k^n - 1$ までのすべての n ビット k 進数表示を丁度 1 回部分列として含むような長さ k^n の系列を k 進 n 次元のドブリュイン系列 (k -ary n -dimensional de Bruijn sequence) という.

例 2 00111010(00) は, 2 進 3 次元のドブリュイン系列の 1 つである.

符号理論で知られている M 系列は, 規則性と不規則性を合わせ持つ系列であり, 0 のみからなるビットパターンを含まないという点を除いて, 2 進ドブリュイン系列の一部である [4]. なお, $DB_{2,n}$ の増加的構成法である倍化法については, 付録 A.1 に記述する.

2 ドブリュイングラフと正規言語

$(0+1)^*1(0+1)^{n-1}$ を受理する DFA との関係

本章では、正規言語 $(0+1)^*1(0+1)^{n-1}$ を受理する非決定性有限オートマトン $N_{2,n}$ を決定性に変換した $D_{2,n}$ の状態遷移図が、2進 n 次元ドブリュイングラフ $DB_{2,n}$ と同型になることを $n = 4$ の場合まで確認する。また、図6に示すように、一般化した場合も同型となることを証明するために、ドブリュイングラフがシフトレジスタの完全状態図であることと、 $N_{2,n}$ から $D_{2,n}$ に変換する部分集合構成法の分析結果がシフトレジスタの動きと一致することを利用する。更には、2進の場合の同型性の結果を k 進に拡張することを試みる。

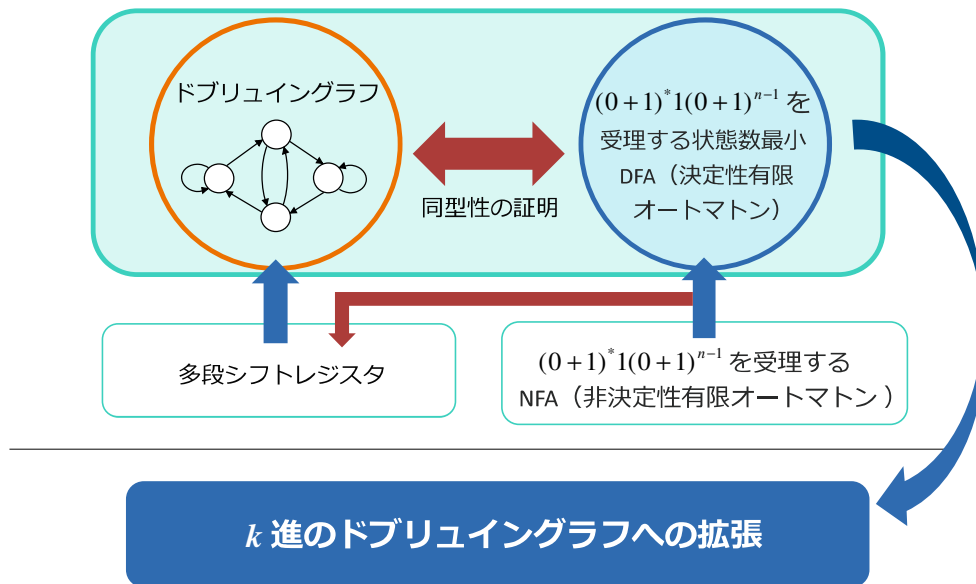


図 6: 第 2 章のアウトライン.

2.1 正規言語 $(0+1)^*1(0+1)^{n-1}$ を受理する有限オートマトン $D_{2,n}$

ここで、 $L_{2,n} = \{x \in \{0,1\}^* \mid \text{文字列 } x \text{ の後ろから } n \text{ 番目が } 1\}$ すなわち、右端から n 番目の文字が 1 であるような $\{0,1\}$ の文字列の集合を考える。 $L_{2,n}$ を受理する NFA $N_{2,n}$ は次のように表せる。

各 $i = 1, \dots, n-1$, $a \in \{0,1\}$ に対して、

$$\begin{aligned} N_{2,n} &= (\{r_0, r_1, \dots, r_n\}, \{0,1\}, \delta, r_0, \{r_n\}), \\ \delta(r_0, 0) &= \{r_0\}, \delta(r_0, 1) = \{r_0, r_1\}, \\ \delta(r_i, a) &= \{r_{i+1}\}, i = 1, \dots, n-1, a \in \{0,1\}. \end{aligned}$$

図 7, 8, 9, ならびに 10 はそれぞれ $N_{2,1}$, $N_{2,2}$, $N_{2,3}$, ならびに $N_{2,4}$ の状態遷移図である。 $L_{2,n}$ を受理する DFA を $D_{2,n}$ とする。部分集合構成法により $N_{2,1}$, $N_{2,2}$, $N_{2,3}$, ならびに $N_{2,4}$ からそれぞれ $D_{2,1}$, $D_{2,2}$, $D_{2,3}$, ならびに D_4 が得られる。これらの DFA をそれぞれ図 11, 12, 13, ならびに 14 に示す。

例 3 部分集合構成法により、NFA $N_{2,1} = (\{r_0, r_1\}, \{0,1\}, \delta, r_0, \{r_1\})$ から DFA $D_{2,1} = (2^Q, \Sigma, \delta', F')$ を構成する。

ここに、

$$\delta(r_0, 0) = \{r_0\}, \delta(r_0, 1) = \{r_0, r_1\}, \delta(r_1, 0) = \delta(r_1, 1) = \emptyset.$$

F' は Q の部分集合 S で、 $S \cap F \neq \emptyset$ を満たす S から成る集合である。従って、 $S = \{r_0, r_1\}, F = \{r_1\}$ より、 $F' = \{\{r_1\}, \{r_0, r_1\}\}$ となる。

各 $S \subseteq Q$ と各入力記号 $a \in \{0,1\}$ に対して、 $\delta'(S, a) = \bigcup_{p \in S} \delta(p, a)$ であるから、

$$\begin{aligned} \delta'(\{r_0\}, 0) &= \delta(r_0, 0) = \{r_0\}, \delta'(\{r_0\}, 1) = \delta(r_0, 1) = \{r_0, r_1\}, \\ \delta'(\{r_1\}, 0) &= \delta(r_1, 0) = \emptyset, \delta'(\{r_1\}, 1) = \emptyset, \\ \delta'(\{r_0, r_1\}, 0) &= \delta(r_0, 0) \cup \delta(r_1, 0) = \{r_0\} \cup \emptyset = \{r_0\}, \\ \delta'(\{r_0, r_1\}, 1) &= \delta(r_0, 1) \cup \delta(r_1, 1) = \{r_0, r_1\} \cup \emptyset = \{r_0, r_1\}, \end{aligned}$$

2.2 2進 n 次元ドブリュイングラフ $DB_{2,n}$ と $D_{2,n}$ の関係

ここで, $\{r_0\} = q_0, \{r_0, r_1\} = q_1$ とおくと, 図 11 に示す $L_{2,1}$ を受理する DFA $D_{2,1}$ が得られる.

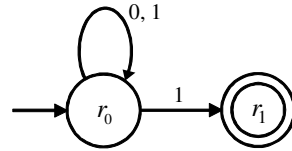


図 7: $L_{2,1}$ を受理する NFA $N_{2,1}$.

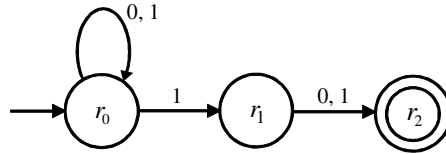


図 8: $L_{2,2}$ を受理する NFA $N_{2,2}$.

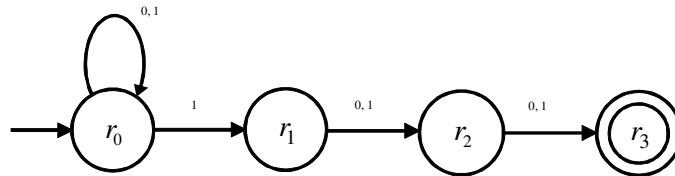


図 9: $L_{2,3}$ を受理する NFA $N_{2,3}$.

2.2 2進 n 次元ドブリュイングラフ $DB_{2,n}$ と $D_{2,n}$ の関係

本節では, $L_{2,n}$ を受理する決定性有限オートマトン $D_{2,n}$ の状態遷移図は, 2進 n 次元のドブリュイングラフ $DB_{2,n}$ と同型であることを示す.

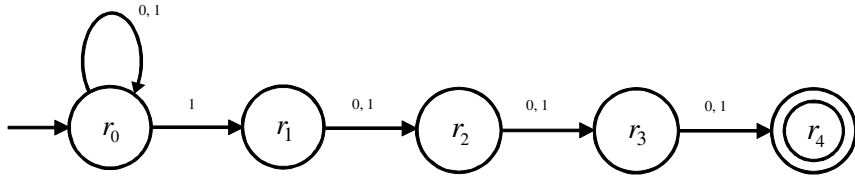


図 10: $L_{2,4}$ を受理する NFA $N_{2,4}$.

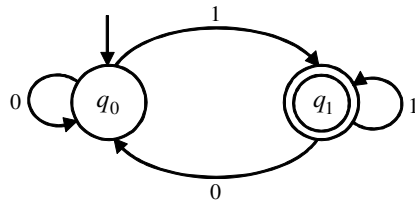


図 11: $L_{2,1}$ を受理する DFA $D_{2,1}$.

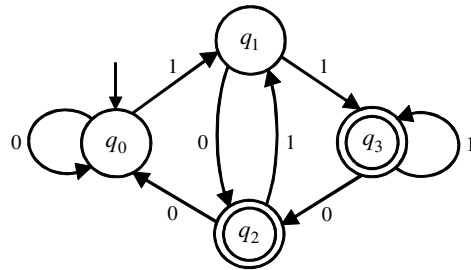


図 12: $L_{2,2}$ を受理する DFA $D_{2,2}$.

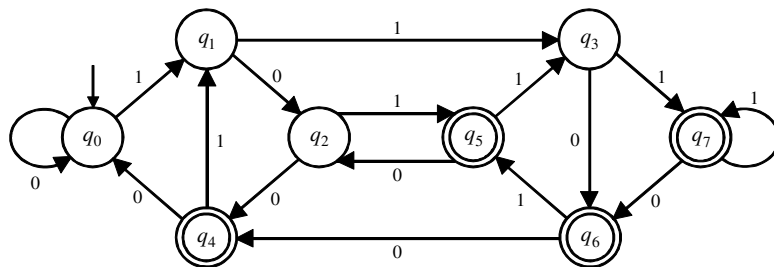


図 13: $L_{2,3}$ を受理する DFA $D_{2,3}$.

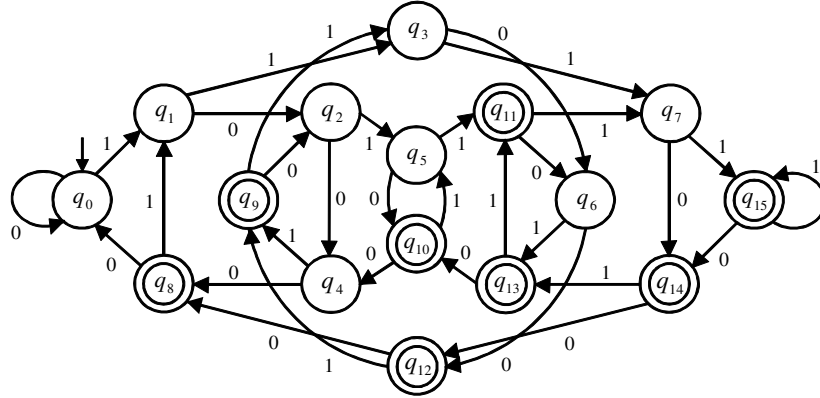


図 14: $L_{2,4}$ を受理する DFA $D_{2,4}$.

定理 2 部分集合構成法によって $N_{2,n}$ から求めた $D_{2,n}$ は, $n \geq 1$ に対して $DB_{2,n}$ と同型である.

(証明) 上述の $N_{2,n}$ に対して, 部分集合構成法を適用すると, 次のような DFA $D_{2,n}$ が得られる.

$$\begin{aligned}
 D_{2,n} &= (Q', \{0, 1\}, \delta', q'_0, F'), \\
 Q' &= \{[1x_1 \cdots x_n] \mid x_i \in \{0, 1\}, i = 1, \dots, n\}, \\
 q_0 &= [10 \cdots 0], \\
 F' &= \{[1x_1 \cdots x_{n-1}1] \mid x_i \in \{0, 1\}, i = 1, \dots, n\}.
 \end{aligned}$$

各 $b_1 \cdots, b_n \in \{0, 1\}, a \in \{0, 1\}$ に対し,

$$\delta'([1b_1 \cdots b_n], a) = \begin{cases} [10b_1 \cdots b_{n-1}], & a = 0 \text{ のとき,} \\ [11b_1 \cdots b_{n-1}], & a = 1 \text{ のとき.} \end{cases}$$

ここに, $[b_0, b_1 \cdots b_n] \in \{0, 1\}, a \in \{0, 1\}$ は $\{r_0, r_1, \dots, r_n\}$ の部分集合を表すためのビット列 (特性関数) である. b_0 の後の $[b_1 \cdots b_n]$ を左右反転させて得られる 2

進値を 10 進数に変換し、状態の名前を変更すると、以下のように書き直せる。

$$\begin{aligned} Q' &= \{q_0, \dots, q_{2^n-1}\}, \\ q'_0 &= q_0, \\ F' &= \{q_{2^{n-1}}, \dots, q_{2^n-1}\}. \end{aligned}$$

各 $i = 0, \dots, 2^n - 1, a \in \{0, 1\}$ に対して、

$$\delta(q_i, a) = \begin{cases} q_{2i \bmod 2^n}, & a = 0 \text{ のとき,} \\ q_{(2i+1) \bmod 2^n}, & a = 1 \text{ のとき.} \end{cases}$$

$D_{2,n}$ のこの記述は、定義 2 で述べた $DB_{2,n}$ の以下の定義に一致する。

$$\begin{cases} V = \{0, 1, \dots, 2^n - 1\}, \\ E = \{(x, 2x \bmod 2^n), (x, (2x + 1) \bmod 2^n) \mid x \in V\}. \end{cases}$$

□

$L_{2,n}$ を受理する DFA は状態数が 2^n 個以上必要であることが知られている [16]. 定理 2 の証明で $|Q'| = 2^n$ であることに着目すると、 $D_{2,n}$ は $L_{2,n}$ を受理する最小状態数の DFA であると言える。

例 4 $N_{2,2} = (\{r_0, r_1, r_2\}, \{0, 1\}, \delta, r_0, \{r_2\})$ から $D_{2,2}$ を求める際の部分集合構成法を分析し、 $D_{2,2}$ の状態遷移図は $DB_{2,2}$ と同型であることを示す。

ここで、図 8 で表される $N_{2,2}$ の状態集合 $\{r_0, r_1, r_2\}$ の r_0 を必ず含む部分集合は、 $\{\{r_0\}, \{r_0, r_1\}, \{r_0, r_2\}, \{r_0, r_1, r_2\}\}$ である。 r_0 を含まないと状態集合は $D_{2,2}$ において初期状態から到達不能であることに注意されたい。これらの部分集合をビット列 (特性関数) で表す。つまり、状態 r_0, r_1, r_2 を順番に並べて、状態が存在する場合は 0 で表し、状態が存在しない場合は 1 で表す。上述の部分集合を特性関数で表したものは次のようになる。

$$\{[100], [110], [101], [111]\}.$$

上記のそれぞれの特性関数が $D_{2,2}$ における各状態になる。次に、 $D_{2,2}$ における各状態の遷移について考える。まず、状態 $\{r_0, r_1\} = [110]$ に着目する。 $N_{2,2}$ において、 $\{r_0, r_1\} = [110]$ からの行先は、入力が 0 のとき、 r_0 から r_0 , r_1 から r_2 となる。よって、 $D_{2,2}$ での遷移先は r_0 と r_2 両方を含む $\{r_0, r_2\} = [101]$ となる。また、 $\{r_0, r_1\} = [110]$ における入力が 1 のときの $N_{2,2}$ での遷移先は、 r_0 からは r_0, r_1 へ且つ r_1 から r_2 へなので、 $D_{2,2}$ での遷移先は r_0, r_1, r_2 すべてを含む $\{r_0, r_1, r_2\} = [111]$ となる。 $\{r_0, r_1\} = [110]$ における入力 0, 1 に対する遷移の様子を図 15 に示す。残りの 3 つの状態についても同様に入力 0, 1 における遷移先を求めると、図 12 に示す $D_{2,2}$ を得ることができる。

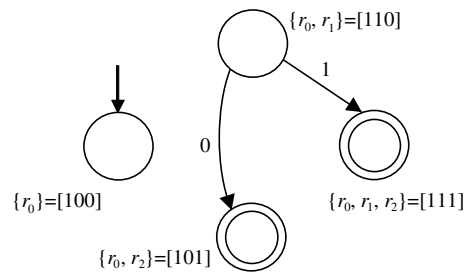


図 15: $N_{2,2}$ から $D_{2,2}$ に変換する過程。

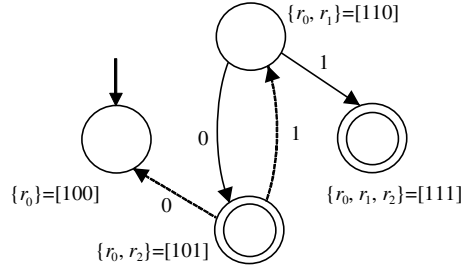
上記の遷移を求める作業を遷移関数で表すと次のようになる。

各 $b_1, b_2 \in \{0, 1\}, a \in \{0, 1\}$ に対し、

$$\delta'([1b_1b_2], a) = \begin{cases} [10b_1], & a = 0 \text{ のとき,} \\ [11b_1], & a = 1 \text{ のとき.} \end{cases}$$

例えば、 $b_1b_2 = 01$ とすると、 $a = 0$ のとき $[100]$, $a = 1$ のとき $[110]$ となる。その遷移の様子を図 16 の点線の矢印で示す。

ここで、 $[1b_1b_2]$ の $[b_1b_2]$ 部分を左右反転させると、次のような $D_{2,2}$ の定義が得ら


 図 16: $b_1b_2 = 01$ のときの遷移の様子.

れる.

$$D_{2,2} = (\{00, 10, 01, 11\}, \{0, 1\}, \delta', 00, \{01, 11\}).$$

ここに, 各 $b_1, b_2 \in \{0, 1\}, a \in \{0, 1\}$ に対し,

$$\delta'([b_2b_1], a) = \begin{cases} b_10, & a = 0 \text{ のとき,} \\ b_11, & a = 1 \text{ のとき.} \end{cases}$$

上記の $D_{2,2}$ の定義は, 下記の $DB_{2,2}$ の定義に一致する.

$$\begin{cases} V = \{00, 10, 01, 11\}, \\ E = \{(00, 00), (00, 01), (01, 10), (01, 11), (10, 00), (10, 01), (11, 10), (11, 11)\}. \end{cases}$$

□

上記の遷移関数

$$\delta'([1b_1b_2], a) = \begin{cases} [10b_1], & a = 0 \text{ のとき,} \\ [11b_1], & a = 1 \text{ のとき.} \end{cases}$$

は入力 $0, 1$ を読み込むと, 遷移元の左端の 1 は固定したままで 2 ビット目から右に 1 ビットシフトし, 2 ビット目に入力の $0, 1$ が挿入される. この入力に対する右へのシフトと入力の挿入は, 図 17 に示す 2 段シフトレジスタへ値 $0, 1$ が入力されたときの状態遷移に対応している. つまり $N_{2,2}$ に対する部分集合構成法自体が 2 段シフトレジスタと同じ動きを基に構成されていると言える. ドブリュイングラフは多段シフトレジスタの完全状態図に等しいのであるから, 定理 2 が主張する

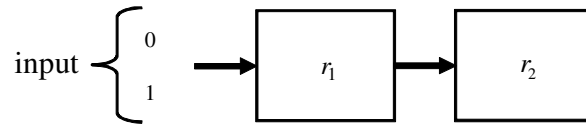


図 17: $N_{2,2} - D_{2,2}$ 変換により得られた遷移関数の動きに対応する 2 段シフトレジスタ。

ところの $D_{2,n}$ と $DB_{2,n}$ の同型性の根本的理由がここに現れている。

なお、 $DB_{2,n}$ と $D_{2,n}$ の同型性が証明されたことを受け、倍化法の考え方に基づいて、 $D_{2,n-1}$ から $D_{2,n}$ を構成するアルゴリズムである増加的部分集合構成法を付録 A.2 に示している。このことはグラフ理論の概念がオートマトン理論に適用できる有用な例と言える。この構成法は [44] で発表した。

2.3 k 進への一般化の試み

前節において 2 進の場合つまり $DB_{2,n}$ と $D_{2,n}$ の状態遷移図の同型性の証明を行った。次は、その結果を k 進へ一般化することを試みる。ここで、 $\{0, 1, \dots, k-1\}$ 上の文字列の集合 $L_{k,n} = \{w \in \{0, 1, \dots, k-1\}^* \mid x \text{ の末尾から } n \text{ 番目の文字が } 1, 2, \dots, \text{ または } k-1 \text{ である}\}$ について検討する。 $L_{3,1}, L_{3,2}, L_{4,1}$ を受理する NFA $N_{3,1}, N_{3,2}, N_{4,1}$ の状態遷移図をそれぞれ図 18, 19, 20 に示し、それらから事実 2 の部分集合構成法を用いて求めた DFA $D_{3,1}, D_{3,2}, D_{4,1}$ をそれぞれ図 21, 22, 23 に示す。

これらの状態遷移図と、ドブリュイングラフ $DB_{3,1}, DB_{3,2}, DB_{4,1}$ を比較すると、ラベルを無視した構造は同一であることを確認できる。ただし、これらの DFA の状態数は最小ではない。すなわち、穴埋めアルゴリズムや Hopcroft のアルゴリズムといった良く知られている最小化アルゴリズム [16, 17] により最小化すると簡略化されてしまう。詳しく言うと、各 DFA $D_{3,1}, D_{3,2}, D_{4,1}$ は各図 24, 25, 26 に示す

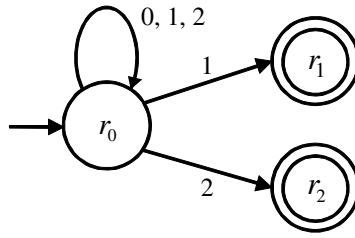


図 18: $L_{3,1}$ を受理する NFA $N_{3,1}$.

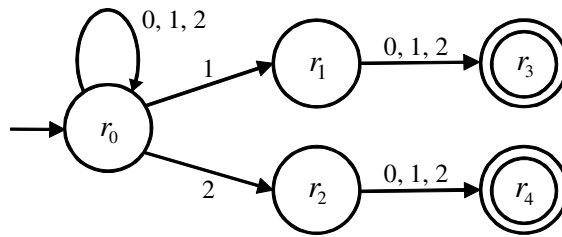


図 19: $L_{3,2}$ を受理する NFA $N_{3,2}$.

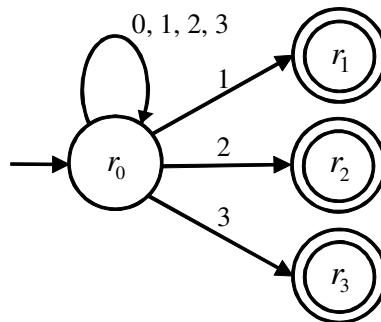


図 20: $L_{4,1}$ を受理する NFA $N_{4,1}$.

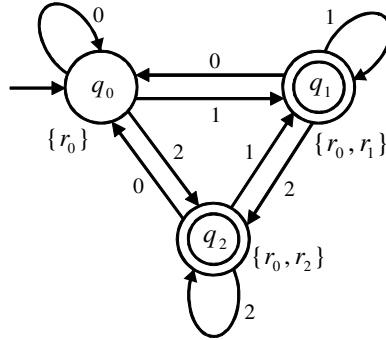


図 21: $L_{3,1}$ を受理する DFA $D_{3,1}$.

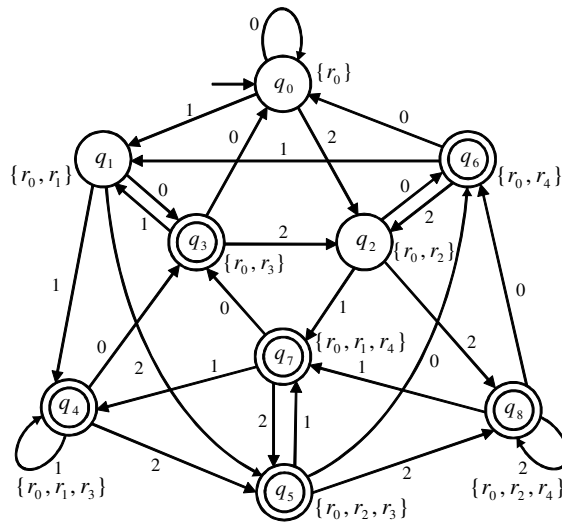


図 22: $L_{3,2}$ を受理する DFA $D_{3,2}$.

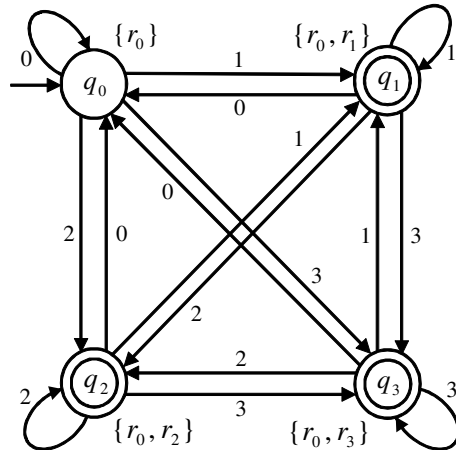


図 23: $L_{4,1}$ を受理する DFA $D_{4,1}$.

DFA $D'_{3,1}, D'_{3,2}, D'_{4,1}$ に簡略化される. つまり, 辺のラベルを除いて $k = 2$ の場合のオートマトン $D_{2,n}$ と同型になる. このように 2 進の同型性の結果を k 進に拡張し, DFA の状態数の最小性を主張するためには, 状態同士の等価性の新たな定義が必要となる. それは, オートマトンが行う一つの言語に対する従来の受理, 非受理の認識から, 数種類の言語を識別する機能を持たせることが必要になる. そのようなオートマトンを, 我々は色付き有限オートマトン (Colored Finite Automata) と呼び, 次章においてその定義を与える.

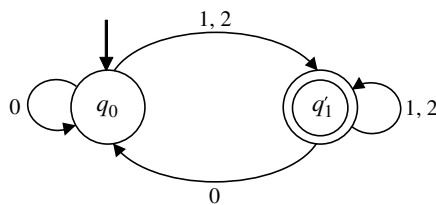
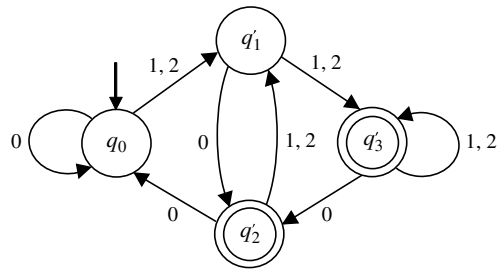
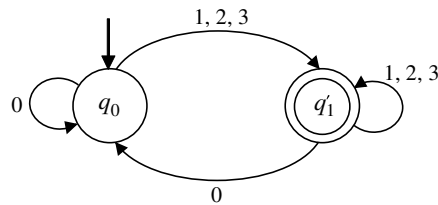


図 24: DFA $D_{3,1}$ を最小化した DFA $D'_{3,1}$.

図 25: DFA $D_{3,2}$ を最小化した DFA $D'_{3,2}$.図 26: DFA $D_{4,1}$ を最小化した DFA $D'_{4,1}$.

2.4 まとめ

- 2進 n 次元ドブリュイングラフ $DB_{2,n}$ と $D_{2,n}$ の同型性の証明：右端から n 番目が1であるような文字列の集合 $L_{2,n}$ を受理する非決定性有限オートマトン $N_{2,n}$ は，決定性 $D_{2,n}$ に変換した際に状態数が指数関数的に増え，且つ DFA の状態数が最小（これ以上状態数を減らせない）であるような性質を持つ有限オートマトンである．この $N_{2,n}$ から $D_{2,n}$ に変換する際の部分集合構成法を分析し，その分析結果が2進 n 次元ドブリュイングラフ $DB_{2,n}$ の定義と一致することを示すことにより，同型性を示した．（証明の詳細は [42] で発表した．）
- k 進への一般化： $DB_{2,n}$ と $D_{2,n}$ が同型であるという2進での結果を k 進に拡張することが期待されるが，3進以上の DFA に最小化アルゴリズムを適用す

ると状態数が削減されてしまい, k 進の場合の同型性を主張することができないことを明らかにした. (この主張は [42] で発表した.)

3 色付き有限オートマトン

本章では、2進 n 次元ドブリュイングラフと、正規言語 $(0+1)^*1(0+1)^{n-1}$ を受理する最小状態数の決定性有限オートマトン $D_{2,n}$ の同型性の証明結果を k 進に拡張するために、手段として必要となった色付き有限オートマトンの定義を行う。そして、この色付き有限オートマトンを用いて k 進 n 次元ドブリュイングラフ $DB_{k,n}$ と $(k-1)$ 項の正規言語 $(0+1+\dots+k-1)^*1(0+1+\dots+k-1)^{n-1}, \dots, (0+1+\dots+k-1)^*(k-1)(0+1+\dots+k-1)^{n-1}$ を受理する最小状態数の色付き有限オートマトン $D_{k,n}$ の状態遷移図が同型であることを証明する。また、NCFAの非混色性に関する決定問題の計算複雑さについて議論を行い、NCFAの適用例を示すことにより色付き受理状態を導入することのメリットを紹介する。この章の最後には、NCFAの応用として正規表現エンジンとモデル検査への応用例を例示する。

3.1 非決定性色付き有限オートマトン (NCFA)

定義 4 L_i をアルファベット Σ 上の言語とする ($i = 1, \dots, k, k \geq 1$)。 (1) 言語の k 個組 (L_1, L_2, \dots, L_k) を、 Σ 上の k 色の**色付き言語** (ベクトル) と呼ぶ。 (2) これらの言語の直和¹ $\sum_{i=1}^k L_i$ によって言語 L が表現される場合、 L を k 色で**純色** (あるいは**明瞭**に) **色付けされた** Σ 上の言語と呼ぶ。

色付き言語ベクトルの例を図 27 に示す。図 27 左では、 $k = 4$ つまり色が 4 色の場合の Σ 上の色付き言語を表している。図 27 中央は、この色付き言語が重なりあ

¹集合 X と Y に対して、直和 $X + Y$ は条件 $X \cap Y = \emptyset$ を満たす和集合 $X \cup Y$ である。従って、上記の (単一言語) $\sum_{i=1}^k L_i$ は L_i が互いに素であるときのみ定義される。正規表現 $r+s$ はまた言語 $L_1 \cup L_2$ を表すことに注意されたい。ここに、 L_1 と L_2 はそれぞれ正規表現 r と s で表される言語である。

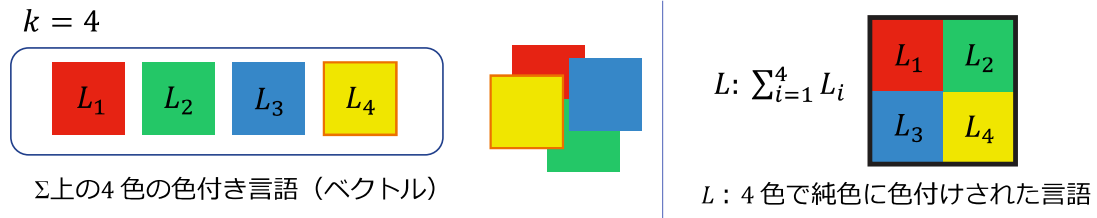


図 27: 色付き言語ベクトルの例.

ている様子を表しており、図 27 右では混色していない 4 色で純色に色付けされた言語の様子を表している.

上記の言語ベクトルや言語の組という用語は奇妙に思われるかもしれないが、同じ概念が形式文法の分野では暗に使用されている [18, 24, 25]. 例えば、文脈自由文法のわずかな拡張である多重文脈自由文法 (multiple context-free grammars) [24] における終端文字列の導出過程の中では、文形式の多次元ベクトルが異なる次元の別のベクトルに書き換えられている. また、Brzozowski [51] は多出力順序回路のモデルとして、正規表現ベクトル $R = (R_1, R_2, \dots, R_r)$ を導入し、各 R_i は独立しているもののそれらが表す言語ベクトル $L = (L_1, L_2, \dots, L_R)$ が一つの DFA として構成できることを示している.

定義 5 次のように表される 5 項組 $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^k F_i)$ を非決定性色付き有限オートマトン (nondeterministic colored finite automata) と呼び、NCFA と略記する.

1. Q は状態の有限集合.
2. Σ は入力文字列の有限集合.
3. δ は遷移関数で $Q \times \Sigma$ から 2^Q への関数.
4. $q_0 \in Q$ は初期状態.

5. $\sum_{i=1}^k F_i \subseteq Q$ は色付き受理状態の集合であり, F_i は i 色の受理状態の集合.

次の例は, ここで導入する諸概念を容易に理解し, 我々の結果の有用性を示すことを目的としている. その非自明性にとまどう読者は, 本稿の概念を標準的な文献にある他の見慣れた NFA の例に適用することを勧める [16, 17, 26].

例 5 NCFA の例 $M_0 = (Q, \Sigma, \delta, q_0, \sum_{i=1}^3 F_i)$ を図 28 に示す. ここに

$$Q = \{0, 1_R, 2, 3_G, 4, 5_B\},$$

$$\Sigma = \{0, 1\},$$

$$q_0 = 0,$$

$$\sum F_i = F_R + F_G + F_B, F_R = \{1_R\}, F_G = \{3_G\}, F_B = \{5_B\}. \quad \blacksquare$$

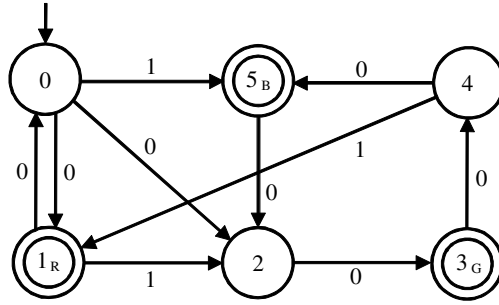


図 28: NCFA の例 M_0 .

もし $\delta(q, a)$ が単項集合の場合, M を決定性色付き有限オートマトン (deterministic colored finite automata) と呼び, DCFA と略記する.

M が状態 q から始まり, 入力文字列 x を読み取った後に終了するときの到達可能な状態の集合を $\hat{\delta}(q, x)$ と表す. もし $\hat{\delta}(q, x) \cap F_i \neq \emptyset$ であるならば, M は入力文字列 x を i 色で受理するという.

$$L_i(M) \triangleq \{x \in \Sigma^* \mid \hat{\delta}(q_0, x) \cap F_i \neq \emptyset\}$$

は i 色で M によって受理される言語と呼ばれる.

$$L(M) \triangleq \bigcup_{i=1}^k L_i(M)$$

M によって i 色で受理される言語 :

$$L_i(M) \triangleq \{x \in \Sigma^* \mid \hat{\delta}(q_0, x) \cap F_i \neq \emptyset\}$$

M によって受理される (統合された) 言語 : $L(M) \triangleq \bigcup_{i=1}^k L_i(M)$

$L(M) = \sum_{i=1}^k L_i(M)$ が成り立つとき,
 $L(M)$ は**非混色**であると言い,
 M は $L(M)$ を**色識別受理**するという.

$L(M) = \sum_{i=1}^4 L_i(M)$

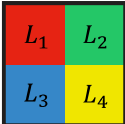


図 29: i 色で受理される言語および統合された言語ならびに非混色の例.

は M によって受理される (統合された) 言語と呼ばれる. 特に,

$$L(M) = \sum_{i=1}^k L_i(M)$$

が成り立つならば $L(M)$ は**非混色**であると言い, M は純色で $L(M)$ を受理すると言う. M が決定性であるかあるいは $k = 1$ の場合 $L(M)$ は元々非混色である. 図 29 は, i 色で受理される言語, M によって受理される (統合された) 言語および $L(M)$ が直和で表される場合は非混色となることを視覚的に分かり易くまとめた図である. また図 29 右下は, $k = 4$ における $L(M)$ の非混色の場合を表す例である.

例 6 例 5 と同じ NCFA について以下が成り立つ.

$$\begin{aligned}
 L_R(M_0) &= (00 + (01 + 0 + 10)(0000 + 0011)^*0010)^* \\
 &\quad \cdot ((01 + 0 + 10)(000 + 0011)^*001 + 0), \\
 L_G(M_0) &= (00 + (01 + 0 + 10)(0000 + 0011)^*0010)^* \\
 &\quad \cdot (01 + 0 + 10)(0000 + 0011)^*0, \\
 L_B(M_0) &= (00 + (01 + 0 + 10)(0000 + 0011)^*0010)^* \\
 &\quad \cdot ((01 + 0 + 10)(0000 + 0011)^*000 + 1), \\
 L(M_0) &= L_R(M_0) \cup L_G(M_0) \cup L_B(M_0) \\
 &= (00 + (01 + 0 + 10)(0000 + 0011)^*0010)^* \\
 &\quad \cdot ((01 + 0 + 10)(0000 + 0011)^* \\
 &\quad \cdot (000 + 0 + 001) + 0 + 1).
 \end{aligned}$$

これらは元の M_0 をそれぞれ F_R, F_G, F_B , ならびに $F_R \cup F_G \cup F_B$ を受理状態とする個別の NFA M_0^R, M_0^G, M_0^B , ならびに M_0^{RGB} と見なし, NFA を正規表現に変換するよく知られた方法 [16] を適用することにより得られる. ■

任意の $x \in \Sigma^*$ について, $x \in L_i(M), i \in I, x \notin L_j(M), j \notin I$ のような一意の $I \subseteq \{1, \dots, k\}$ が存在する. 言い換えると,

$$I(x) \triangleq \{i \in \{1, \dots, k\} \mid \hat{\delta}(q_0, x) \cap F_i \neq \emptyset\}$$

は写像 $I : \Sigma^* \rightarrow 2^{\{1, \dots, k\}}$ である.

次の事実は, 定義から明らかである.

事実 3 (1) $x \in L_i(M) \iff i \in I(x)$.

(2) $x \in L(M) \iff \exists i \in \{1, \dots, k\} [i \in I(x)] \iff I(x) \neq \emptyset$.

上述の事実は以下の命題に利用されている.

命題 1 $L(M)$ が非混色である \iff 任意の $x \in \Sigma^*$ について, $|I(x)| \leq 1$.

(証明) $L(M)$ が混色している

$$\Leftrightarrow \exists i, j, i \neq j \exists x \in \Sigma^* [x \in L_i(M) \cap L_j(M)]$$

$$\Leftrightarrow \exists i, j \exists x \in \Sigma^* [i, j \in I(x), i \neq j]$$

$$\Leftrightarrow \exists x \in \Sigma^* [|I(x)| \geq 2].$$

□

各 $I \subseteq \{1, \dots, k\}$ について、次のように定義する.

$$F'_I \triangleq \{S \subseteq Q \mid S \cap F_i \neq \emptyset, i \in I, S \cap F_j = \emptyset, j \notin I\}.$$

つまり、 F'_I は状態の部分集合の集合であり、それぞれ I に属さない j 色の受理状態を含まないが、 I に属する i 色の受理状態を含んでいる.

命題 2 (1) $I \neq J$ ならば、 $F'_I \cap F'_J = \emptyset$.

$$(2) \bigcup_{I \subseteq \{1, \dots, k\}, I \neq \emptyset} F'_I = \bigcup_{i=1}^k \{S \subseteq Q \mid S \cap F_i \neq \emptyset\}.$$

(証明)

(1) 逆に $S \in F'_I \cap F'_J$ と仮定する. 一般性を失うことなく、 $i_0 \in I, i_0 \notin J$ とすると、

$$\begin{aligned} S \in F'_I, S \in F'_J \\ \Leftrightarrow S \cap F_i \neq \emptyset, i \in I, S \cap F_j = \emptyset, j \notin I, \\ S \cap F_j \neq \emptyset, j \in J, S \cap F_i = \emptyset, i \notin J \\ \Rightarrow S \cap F_{i_0} \neq \emptyset, S \cap F_{i_0} = \emptyset. \end{aligned}$$

これは矛盾である.

(2) 任意の $S \subseteq Q$ について、 $S \cap F_i \neq \emptyset, i \in I, S \cap F_j = \emptyset, j \notin I$ なる一意の $I \subseteq \{1, \dots, k\}$ が存在する. 言い換えると、

$$I(S) \triangleq \{i \in \{1, \dots, k\} \mid S \cap F_i \neq \emptyset\}$$

は写像 $I : 2^Q \rightarrow 2^{\{1, \dots, k\}}$ である.

$$F'_I = \{S \subseteq Q \mid I(S) = I\}$$

より次が得られる.

$$\begin{aligned} & \bigcup_{i=1}^k \{S \subseteq Q \mid S \cap F_i \neq \emptyset\} \\ &= \{S \subseteq Q \mid \exists i \in \{1, \dots, k\} [S \cap F_i \neq \emptyset]\} \\ &= \{S \subseteq Q \mid \exists I \subseteq \{1, \dots, k\} I \neq \emptyset, [I(S) = I]\} \\ &= \bigcup_{I \subseteq \{1, \dots, k\}, I \neq \emptyset} F'_I. \end{aligned}$$

□

例 7 例 5 と同じ NCFA M_0 について考える. 入力文字列 x を 0000 とする. $I(x)$ の定義より, q_0 から 0000 を読み終わって到達可能な状態の集合 $\{3_G, 5_B\}$ と, G 色および B 色の受理状態の集合 F_G, F_B との共通集合は空ではない. つまり,

$$\hat{\delta}(q_0, 0000) \cap F_G \neq \emptyset, \hat{\delta}(q_0, 0000) \cap F_B \neq \emptyset$$

より,

$$G, B \in I(0000), \text{ すなわち } |I(0000)| \geq 2.$$

従って, $L(M_0)$ は混色である.

$$\begin{aligned}
 F'_\emptyset &= \{S \subseteq Q \mid S \cap F_R = \emptyset, S \cap F_G = \emptyset, S \cap F_B = \emptyset\} \\
 &= \{\{0\}, \{2\}, \{4\}, \emptyset, \{0, 2\}, \{0, 4\}, \{2, 4\}, \{0, 2, 4\}\}, \\
 F'_{\{R\}} &= \{S \subseteq Q \mid S \cap F_R \neq \emptyset, S \cap F_G = \emptyset, S \cap F_B = \emptyset\} \\
 &= \{\{1_R\}, \{1_R, 0\}, \{1_R, 2\}, \{1_R, 4\}, \{1_R, 0, 2\}, \\
 &\quad \{1_R, 0, 4\}, \{1_R, 2, 4\}, \{1_R, 0, 2, 4\}\}, \\
 F'_{\{G\}} &= \{S \subseteq Q \mid S \cap F_G \neq \emptyset, S \cap F_R = \emptyset, S \cap F_B = \emptyset\} \\
 &= \{\{3_G\}, \{3_G, 0\}, \{3_G, 2\}, \{3_G, 4\}, \{3_G, 0, 2\}, \\
 &\quad \{3_G, 0, 4\}, \{3_G, 2, 4\}, \{3_G, 0, 2, 4\}\}, \\
 F'_{\{B\}} &= \{S \subseteq Q \mid S \cap F_B \neq \emptyset, S \cap F_R = \emptyset, S \cap F_G = \emptyset\} \\
 &= \{\{5_B\}, \{5_B, 0\}, \{5_B, 2\}, \{5_B, 4\}, \{5_B, 0, 2\}, \\
 &\quad \{5_B, 0, 4\}, \{5_B, 2, 4\}, \{5_B, 0, 2, 4\}\}, \\
 F'_{\{RG\}} &= \{S \subseteq Q \mid S \cap F_R \neq \emptyset, S \cap F_G \neq \emptyset, S \cap F_B = \emptyset\} \\
 &= \{\{1_R, 3_G\}, \{1_R, 3_G, 0\}, \{1_R, 3_G, 2\}, \\
 &\quad \{1_R, 3_G, 4\}, \{1_R, 3_G, 0, 2\}, \{1_R, 3_G, 0, 4\}, \\
 &\quad \{1_R, 3_G, 2, 4\}, \{1_R, 3_G, 0, 2, 4\}\}, \\
 F'_{\{RB\}} &= \{S \subseteq Q \mid S \cap F_R \neq \emptyset, S \cap F_B \neq \emptyset, S \cap F_G = \emptyset\} \\
 &= \{\{1_R, 5_B\}, \{1_R, 5_B, 0\}, \{1_R, 5_B, 2\}, \{1_R, 5_B, 4\}, \\
 &\quad \{1_R, 5_B, 0, 2\}, \{1_R, 5_B, 0, 4\}, \{1_R, 5_B, 2, 4\}, \{1_R, 5_B, 0, 2, 4\}\}, \\
 F'_{\{GB\}} &= \{S \subseteq Q \mid S \cap F_G \neq \emptyset, S \cap F_B \neq \emptyset, S \cap F_R = \emptyset\} \\
 &= \{\{3_G, 5_B\}, \{3_G, 5_B, 0\}, \{3_G, 5_B, 2\}, \{3_G, 5_B, 4\}, \\
 &\quad \{3_G, 5_B, 0, 2\}, \{3_G, 5_B, 0, 4\}, \{3_G, 5_B, 2, 4\}, \{3_G, 5_B, 0, 2, 4\}\}, \\
 F'_{\{RGB\}} &= \{S \subseteq Q \mid S \cap F_R \neq \emptyset, S \cap F_G \neq \emptyset, S \cap F_B \neq \emptyset\} \\
 &= \{\{1_R, 3_G, 5_B\}, \{1_R, 3_G, 5_B, 0\}, \{1_R, 3_G, 5_B, 2\}, \\
 &\quad \{1_R, 3_G, 5_B, 4\}, \{1_R, 3_G, 5_B, 0, 2\}, \\
 &\quad \{1_R, 3_G, 5_B, 0, 4\}, \{1_R, 3_G, 5_B, 2, 4\}, \{1_R, 3_G, 5_B, 0, 2, 4\}\},
 \end{aligned}$$

より次が得られる.

$$\begin{aligned} \bigcup_{I \subseteq \{R,G,B\}, I \neq \emptyset} F'_I &= \sum_{I \subseteq \{R,G,B\}, I \neq \emptyset} F'_I \\ &= F'_{\{R\}} + F'_{\{G\}} + F'_{\{B\}} + F'_{\{RG\}} + F'_{\{RB\}} + F'_{\{GB\}} + F'_{\{RGB\}}. \end{aligned}$$

ここに, 例えば $F'_{\{R\}}$ の要素は F_R との共通集合が空でなく, F_G との共通集合および F_B との共通集合が空であるような Q の部分集合である. 従って, $F'_{\{R\}}$ は 1_R を必ず含み, 且つ $3_G, 5_B$ を含まない Q の部分集合となる. ■

定理 3 (NCFA に対する部分集合構成法)

NCFA

$$M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^k F_i)$$

に対して DCFA を次のように定める.

$$M' = (2^Q, \Sigma, \delta', \{q_0\}, \sum_{I \subseteq \{1, \dots, k\}, I \neq \emptyset} F'_I).$$

ここに

$$\delta'(S, a) = \bigcup_{p \in S} \delta(p, a), S \subseteq Q, a \in \Sigma.$$

更に,

$$L_I(M') \triangleq \{x \in \Sigma^* \mid \hat{\delta}'(\{q_0\}, x) \in F'_I\},$$

$$F'[\{q_0\}] \triangleq \{S \subseteq Q \mid \exists x \in \Sigma^* [S = \hat{\delta}'(\{q_0\}, x), S \in \sum_{I \subseteq \{1, \dots, k\}, I \neq \emptyset} F'_I]\}$$

と定義することにより, 以下が得られる.

$$(1) L_I(M') = \bigcap_{i \in I} L_i(M) - \bigcup_{j \notin I} L_j(M), I \subseteq \{1, \dots, k\},$$

つまり, M' のそれぞれ個々の言語は M の混色状況を正しく反映している.

$$(2) L(M') = \sum_{I \subseteq \{1, \dots, k\}, I \neq \emptyset} L_I(M') = L(M),$$

つまり, M と M' で受理される統合言語は, 全体として一致する.

(3) $L(M)$ が非混色である

$$\iff \text{各 } i \in \{1, \dots, k\} \text{ に対して } L_i(M) = L_{\{i\}}(M'),$$

$$\iff F'[\{q_0\}] \subseteq \sum_{i=1}^k F'_{\{i\}}.$$

つまり、非混色の場合、 M' の色付き言語自体と色の数は M と同じである。

(証明)

(1) $x \in \bigcap_{i \in I} L_i(M) - \bigcup_{j \notin I} L_j(M)$

$$\iff x \in L_i(M), i \in I, x \notin L_j(M), j \notin I$$

$$\iff \hat{\delta}(q_0, x) \cap F_i \neq \emptyset, i \in I, \hat{\delta}(q_0, x) \cap F_j = \emptyset, j \notin I.$$

ここで、部分集合構成法の NCFA 版は受理状態を除いて通常の NFA 版と同じであるので、 $\hat{\delta}(q_0, x) = \hat{\delta}'(\{q_0\}, x)$ が成り立つ。この等式の左側の部分は NFA の状態の集合を表し、右側の部分は DFA の状態の 1 つを表すことに注意されたい。従って、

$$x \in \bigcap_{i \in I} L_i(M) - \bigcup_{j \notin I} L_j(M)$$

$$\iff \hat{\delta}'(\{q_0\}, x) \cap F_i \neq \emptyset, i \in I, \hat{\delta}'(\{q_0\}, x) \cap F_j = \emptyset, j \notin I$$

$$\iff \hat{\delta}'(\{q_0\}, x) \in F'_I$$

$$\iff x \in L_I(M').$$

(2) 逆に $x \in L_I(M') \cap L_J(M'), I \neq J$ と仮定すると、 $\hat{\delta}'(\{q_0\}, x) \in F'_I \cap F'_J, I \neq J$ が得られる。これは $\bigcup F'_I = \sum F'_I$ と矛盾する。それ故に、 $\bigcup L_I(M') = \sum L_I(M')$ 。

$$\begin{aligned}
 x \in L(M') & \\
 \Leftrightarrow \exists I \subseteq \{1, \dots, k\}, I \neq \emptyset [x \in L_I(M')] & \\
 \Leftrightarrow x \notin L_\emptyset(M') & \\
 \Leftrightarrow \hat{\delta}'(\{q_0\}, x) \notin F'_\emptyset & \\
 \Leftrightarrow \overline{\forall j \in \{1, \dots, k\} [\hat{\delta}'(\{q_0\}, x) \cap F_j = \emptyset]} & \\
 \Leftrightarrow \exists j \in \{1, \dots, k\} [\hat{\delta}'(\{q_0\}, x) \cap F_j \neq \emptyset] & \\
 \Leftrightarrow \exists j \in \{1, \dots, k\} [x \in L_j(M)] & \\
 \Leftrightarrow x \in L(M). &
 \end{aligned}$$

ここに、 \bar{P} は P の否定を表す。

(3) 本定理の(1)から、

$L(M)$ が非混色である

$$\Leftrightarrow \text{任意の } i, j \text{ に対して, } L_i(M) \cap L_j(M) = \emptyset \ (i \neq j)$$

$$\Leftrightarrow \text{任意の } i \text{ に対して,}$$

$$\begin{aligned}
 L_{\{i\}}(M') &= L_i(M) - \cup_{j \neq i} L_j(M) \\
 &= L_i(M) - \cup_{j \neq i} (L_i(M) \cap L_j(M)) \\
 &= L_i(M) .
 \end{aligned}$$

最後の同等性については,

$L(M)$ が混色である

$$\Leftrightarrow \exists x \in \Sigma^* [|I(x)| \geq 2]$$

$$\Leftrightarrow \exists x \in \Sigma^* \exists I \subseteq \{1, \dots, k\}, |I| \geq 2$$

$$[\hat{\delta}(q_0, x) \cap F_i \neq \emptyset, i \in I, \hat{\delta}(q_0, x) \cap F_j = \emptyset, j \notin I]$$

$$\Leftrightarrow \exists x \in \Sigma^* \exists I \subseteq \{1, \dots, k\}, |I| \geq 2 [\hat{\delta}'(\{q_0\}, x) \in F'_I]$$

$$\Leftrightarrow \exists I \subseteq \{1, \dots, k\}, |I| \geq 2 \exists x \in \Sigma^* [\hat{\delta}'(\{q_0\}, x) \in F'_I].$$

従って,

$L(M)$ が非混色である

$$\Leftrightarrow \forall I \subseteq \{1, \dots, k\}, |I| \geq 2, \forall x \in \Sigma^* [\hat{\delta}'(\{q_0\}, x) \notin F'_I]$$

$$\Leftrightarrow \text{初期状態 } \{q_0\} \text{ から到達可能な } M' \text{ の任意の受理状態は}$$

$$|I| \leq 1 \text{ なる } F'_I \text{ に属する}$$

$$\Leftrightarrow F'[\{q_0\}] \subseteq \sum_{i=1}^k F'_{\{i\}} \subseteq \sum_{I \subseteq \{1, \dots, k\}, I \neq \emptyset} F'_I.$$

□

非決定性を決定性に変換した場合, 状態数の指数関数的な増加に加えて, M' の色の数も指数関数的に増加する, すなわち k から $2^k - 1$ に増加する可能性があることに注意されたい.

例 8 図 30 は NCFA に対する部分集合構成法によって, NCFA M_0 から変換された DCFA M_1 を示している. ここに,

$$M_1 = (Q, \Sigma, \delta, q_0, \sum_{I \subseteq \{RGB\}, I \neq \emptyset} F'_I),$$

$$Q = \{A, B_R, C_G, D_R, E_{GB}, F_B, G, H_G, I, J_R, K\},$$

$$\Sigma = \{0, 1\},$$

$$q_0 = A,$$

$$\sum F'_I = F'_{\{R\}} + F'_{\{G\}} + F'_{\{B\}} + F'_{\{GB\}}, F'_{\{R\}} = \{B_R, D_R, J_R\},$$

$$F'_{\{G\}} = \{C_G, H_G\}, F'_{\{B\}} = \{F_B\}, F'_{\{GB\}} = \{E_{GB}\}.$$

M_1 には4つの異なる色があり, M_0 の元の3色 R, G, Bから合成色 GBが一つ増加していることに注意されたい. ■

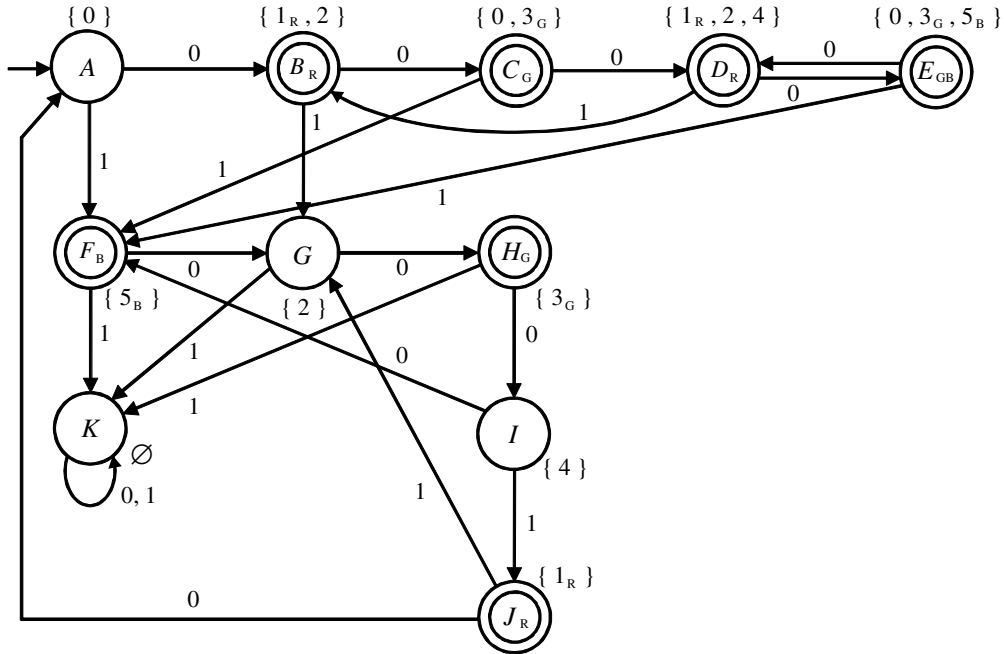


図 30: NCF A M_0 から構成された DCFA M_1 .

3.2 k 進 n 次元ドブリュイングラフ $DB_{k,n}$ と $D_{k,n}$ の関係

この節では, ある特定の決定性色付き有限オートマトンの状態遷移図が, k 進 n 次元ドブリュイングラフ $DB_{k,n}$ と同型であることを示す.

次のように定義する.

$$N_{k,n} = (Q, \{0, 1, \dots, k-1\}, \delta, r_0, \sum_{i=1}^{k-1} F_i).$$

ここに

$$\begin{aligned}
 Q &= \{r_0, r_{11}, \dots, r_{1n}, \dots, r_{(k-1)1}, \dots, r_{(k-1)n}\}, \\
 \delta(r_0, 0) &= \{r_0\}, \\
 \text{各 } a \in \{0, 1, \dots, k-1\} \text{ について, } \delta(r_0, a) &= \{r_0, r_{a1}\}, \\
 \text{各 } i = 1, \dots, k-1, a \in \{0, 1, \dots, k-1\} \text{ について, } \delta(r_{ij}, a) &= \{r_{ij+1}\}, \\
 \text{各 } i = 1, \dots, k-1 \text{ について, } F_i &= \{r_{in}\}.
 \end{aligned}$$

図 31 に一般的な $N_{k,n}$ の状態遷移図を示す. $N_{k,n}$ が非混色であることは明らかであり, 以下が成り立つ.

$$\begin{aligned}
 L(N_{k,n}) &= \{x \in \{0, 1, \dots, k-1\}^* \mid x \text{ の末尾から } n \text{ 番目} \\
 &\quad \text{の文字は } 1, 2, \dots, \text{ または } k-1 \text{ のいずれかである}\} \\
 &= (0+1+\dots+k-1)^*(1+\dots+k-1)(0+1+\dots+k-1)^{n-1} \\
 &= \sum_{i=1}^{k-1} L_i(N_{k,n}).
 \end{aligned}$$

ここに, 各 $i = 1, \dots, k-1$ に対して,

$$\begin{aligned}
 L_i(N_{k,n}) &= \{x \in \{0, 1, \dots, k-1\} \mid x \text{ の末尾から } n \text{ 番目の文字は } i \text{ である}\} \\
 &= (0+1+\dots+k-1)^* i (0+1+\dots+k-1)^{n-1}.
 \end{aligned}$$

以下では, 次のように略記する. 各 $i = 1, \dots, k-1$ について,

$$L_{k,n} \triangleq L(N_{k,n}) \text{ かつ } L_{k,n}^{(i)} \triangleq L_i(N_{k,n})$$

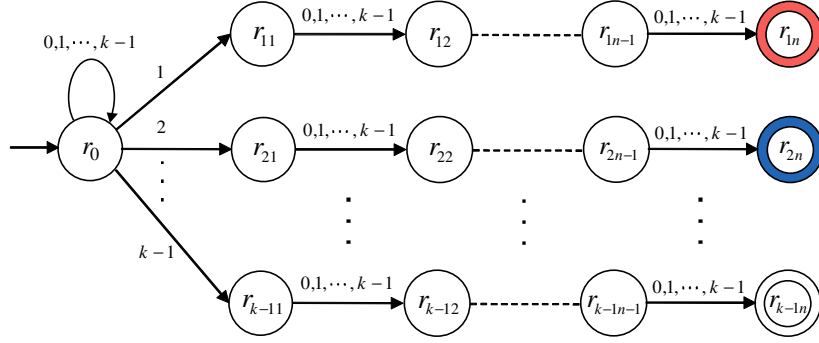
定理 4 任意の $k \geq 2, n \geq 1$ に対して, NCFA $N_{k,n}$ に部分集合構成法を使用して構築された DCFA $D_{k,n}$ の状態遷移図は, $DB_{k,n}$ と同型である.

(証明) 以下では, 次のように表記する.

$$\mathbf{B}^{(k)} = \{\mathbf{0}, \mathbf{1}_1, \mathbf{1}_2, \dots, \mathbf{1}_{k-1}\}.$$

ここに, $\mathbf{0}$ は $00 \cdots 0$ の転置であり, すなわち“縦ベクトル”を表している.

$$\mathbf{0} = [0 \cdots 0]^T, \mathbf{1}_i = [0 \cdots 0 \overset{\leftarrow i}{1} 0 \cdots 0]^T, i = 1, \dots, k-1.$$


 図 31: $L_{k,n}$ を受理する NCFA $N_{k,n}$.

$N_{k,n}$ へ部分集合構成法の NCFA 版を適用すると、以下の DCFA $D_{k,n}$ が得られる。

$$D_{k,n} = (Q', \{0, 1, \dots, k-1\}, \delta', q'_0, \sum_{i=1}^{k-1} F'_{\{i\}}).$$

ここに

$$\begin{aligned} Q' &= \{[1\mathbf{x}_1 \cdots \mathbf{x}_n] \mid \mathbf{x}_j \in \mathbf{B}^{(k)}, j = 1, \dots, n\}, \\ q'_0 &= [1\mathbf{0} \cdots \mathbf{0}], \\ F'_{\{i\}} &= \{[1\mathbf{x}_1 \cdots \mathbf{x}_{n-1}\mathbf{1}_i] \mid \mathbf{x}_j \in \mathbf{B}^{(k)}, j = 1, \dots, n-1\}. \end{aligned}$$

各 $\mathbf{b}_1, \dots, \mathbf{b}_n \in \mathbf{B}^{(k)}$, $a \in \{0, 1, \dots, k-1\}$ に対して、

$$\delta'([1\mathbf{b}_1 \cdots \mathbf{b}_n], a) = \begin{cases} [1\mathbf{0}\mathbf{b}_1 \cdots \mathbf{b}_{n-1}], & a = 0 \text{ のとき,} \\ [1\mathbf{1}_1\mathbf{b}_1 \cdots \mathbf{b}_{n-1}], & a = 1 \text{ のとき,} \\ \vdots \\ [1\mathbf{1}_{k-1}\mathbf{b}_1 \cdots \mathbf{b}_{n-1}], & a = k-1 \text{ のとき.} \end{cases}$$

ここに、 $[1\mathbf{b}_1 \cdots \mathbf{b}_n]$ は $Q = \{r_0, r_{11}, \dots, r_{(k-1)1}, \dots, r_{1n}, \dots, r_{(k-1)n}\}$ の部分集合を表すためのビット列 (特性関数) である。

$D_{k,n}$ の状態遷移は、その入力が記号 $0, 1, \dots$, または $k-1$ の $k-1$ ビット単進表現であるような垂直に並べた $k-1$ 個の n 段シフトレジスタに対応する。その垂直に並べた $k-1$ 個の n 段シフトレジスタを図 32 に示す。

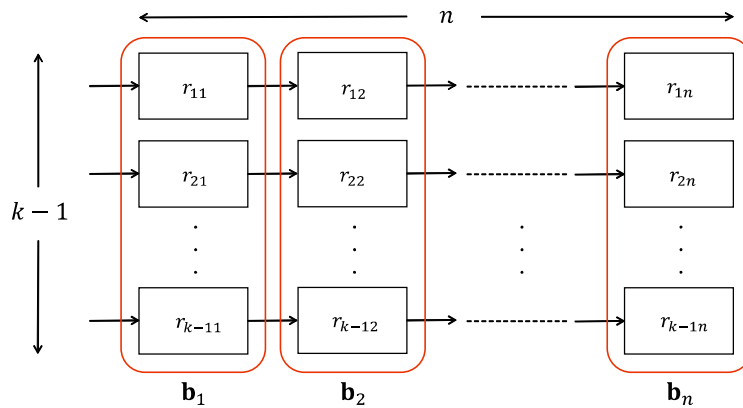


図 32: $N_{k,n} - D_{k,n}$ 変換により得られた遷移関数に対応する垂直に $k-1$ 個並べた n 段シフトレジスタ。

例えば、入力が $a = 0$ または $a = 1$ のとき遷移関数に対応する垂直に $k-1$ 個並べた n 段シフトレジスタはそれぞれ図 33, 34 のようになる。このように NCFA 版の部分集合構成法を分析すると、 $D_{k,n}$ の遷移関数の動きは、垂直に $k-1$ 個並べた n 段シフトレジスタに対応していることがわかる。また、図 35 に示すようにドブリュイングラフは、垂直に $k-1$ 個並べた n 段シフトレジスタの完全状態図であるので、ドブリュイングラフ $DB_{k,n}$ と $L_{k,n}$ を受理する有限オートマトンの状態遷移図が同型であることの理由がここに表れていると言える。

上記の導出は、次の主張によって明確になる。

主張 1 本部分集合構成法は初期状態から探索を開始し、距離の長いものより短い距離にある状態を優先して探索するような幅優先探索を行う。部分集合構成法の j ステップ目では、状態ベクトルは次のように表される。

$$[1\mathbf{b}_1 \cdots \mathbf{b}_j \mathbf{0} \cdots \mathbf{0}], j = 0, 1, \dots, n.$$

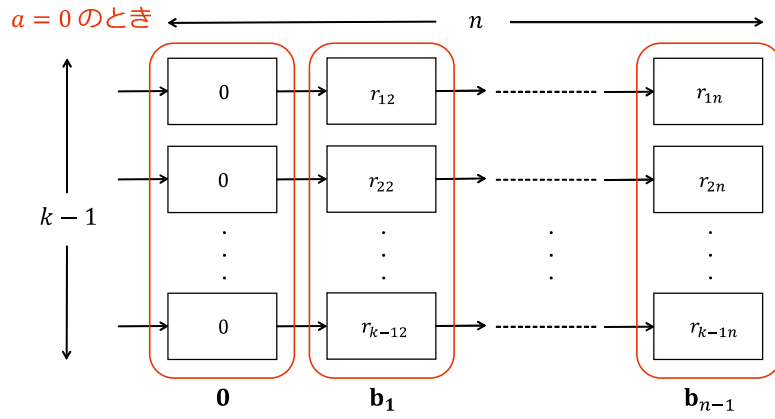


図 33: $a = 0$ の場合の垂直に $k - 1$ 個並べた n 段シフトレジスタ.

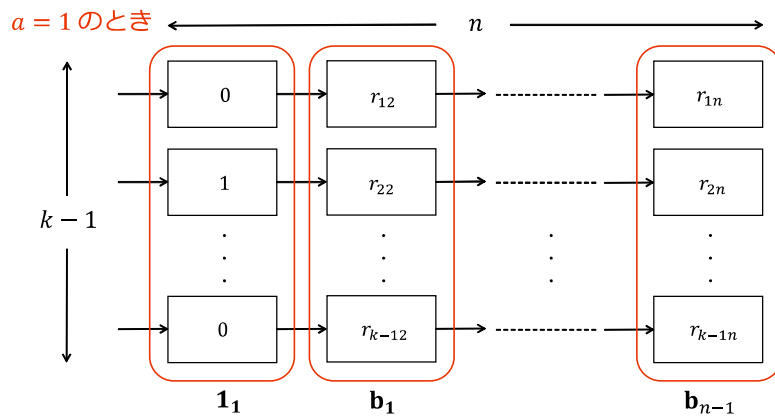


図 34: $a = 1$ の場合の垂直に $k - 1$ 個並べた n 段シフトレジスタ.

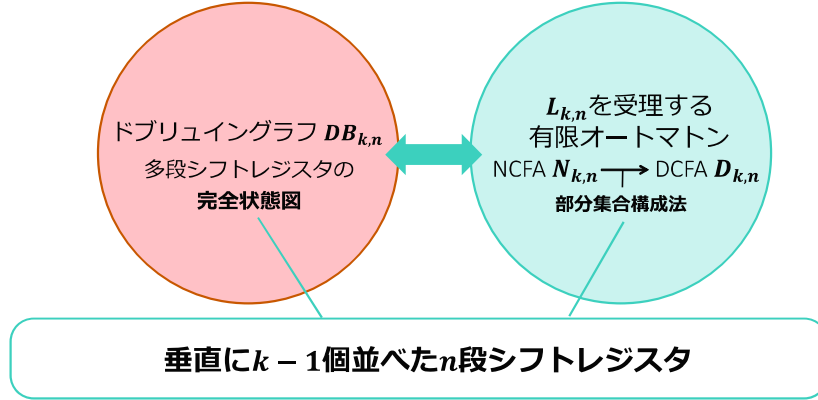


図 35: $DB_{k,n}$ と $D_{k,n}$ の状態遷移図が同型であることのイメージ図.

(証明) 初期状態 $q'_0 = \{q_0\} = [10 \cdots 0]$ は明らかである. もし j ステップで生成される M' のベクトルの集合が $[1b_1 \cdots b_j 0 \cdots 0]$ であるならば, 各 $a \in \{0, 1, \dots, k-1\}$ について,

$$\delta'([1b_1 \cdots b_j 0 \cdots 0], a) = \begin{cases} [10b_1 \cdots b_j 0 \cdots 0], & a = 0 \text{ のとき,} \\ [11b_1 \cdots b_j 0 \cdots 0], & a = 1 \text{ のとき,} \\ \vdots \\ [11_{k-1}b_1 \cdots b_j 0 \cdots 0], & a = k-1 \text{ のとき} \end{cases}$$

より, $(j+1)$ ステップ目のベクトルは, $[1b'_1 b'_2 \cdots b'_{j+1} 0 \cdots 0]$ となる. ここに, $b'_1 \in \mathbf{B}^{(k)}$, $b'_2 = b_1, \dots, b'_{j+1} = b_j$. 特に, $j = n$ のとき, $[1b_1 b_2 \cdots b_n]$ と表される. \square

(定理 4 の証明の続き) $b_1 \cdots b_n$ の左右反転と左端 1 の省略により, $D_{k,n}$ の記述を次のように書き換えることができる.

$$\begin{aligned} Q' &= \{[x_n \cdots x_1 \mid x_j \in \mathbf{B}^{(k)}, j = 1, \dots, n]\}, \\ q'_0 &= [0 \cdots 0], \\ F'_{\{i\}} &= \{[1_i x_{n-1} \cdots x_1 \mid x_j \in \mathbf{B}^{(k)}, j = 1, \dots, n-1]\}, \end{aligned}$$

$$\delta'([\mathbf{b}_n \cdots \mathbf{b}_1], a) = \begin{cases} [\mathbf{b}_{n-1} \cdots \mathbf{b}_1 \mathbf{0}], & a = 0 \text{ のとき,} \\ [\mathbf{b}_{n-1} \cdots \mathbf{b}_1 \mathbf{1}_1], & a = 1 \text{ のとき,} \\ \vdots \\ [\mathbf{b}_{n-1} \cdots \mathbf{b}_1 \mathbf{1}_{k-1}], & a = k-1 \text{ のとき.} \end{cases}$$

更に、垂直ベクトル $\mathbf{0}, \mathbf{1}_1, \mathbf{1}_2, \dots, \mathbf{1}_{k-1}$ をそれぞれ $0, 1, 2, \dots, k-1$ の k 進数字とみなすと、 $D_{k,n}$ の記述を以下のように書き換えることができる。

$$\begin{aligned} Q' &= \{[x_n \cdots x_1]_k \mid 0 \leq x_j < k, j = 1, \dots, n\} \\ &= \{q_0, \dots, q_{k^n-1}\}, \\ q'_0 &= q_0 = [0 \cdots 0]_k. \end{aligned}$$

各 $i = 1, \dots, k-1$ について、

$$\begin{aligned} F'_{\{i\}} &= \{[ix_{n-1} \cdots x_1]_k \mid 0 \leq x_j < k, j = 1, \dots, n-1\} \\ &= \{q_{ik^{n-1}}, \dots, q_{(i+1)k^{n-1}-1}\}. \end{aligned}$$

各 $i = 0, \dots, k^n - 1, a \in \{0, 1, \dots, k-1\}$ について、

$$\delta'(q_i, a) = \begin{cases} q_{ki \bmod k^n}, & a = 0 \text{ のとき,} \\ q_{(ki+1) \bmod k^n}, & a = 1 \text{ のとき,} \\ \vdots \\ q_{(ki+k-1) \bmod k^n}, & a = k-1 \text{ のとき.} \end{cases}$$

上記の $D_{k,n}$ の記述は、定義 2 の $DB_{k,n}$ の記述と同じである：

$$\begin{cases} V = \{0, 1, \dots, k^n - 1\}, \\ E = \{(x, (kx + i) \bmod k^n) \mid x \in V, i = 0, \dots, k-1\}. \end{cases}$$

□

上記のように $D_{k,n}$ の遷移関数を変形すると、ドブリュイングラフ $DB_{k,n}$ の定義と一致することにより、 $N_{k,n}$ に部分集合構成法を使用して構築した $D_{k,n}$ は $DB_{k,n}$ と同型であることを証明した。図 36 に $D_{3,2}$ の状態遷移図を示す。

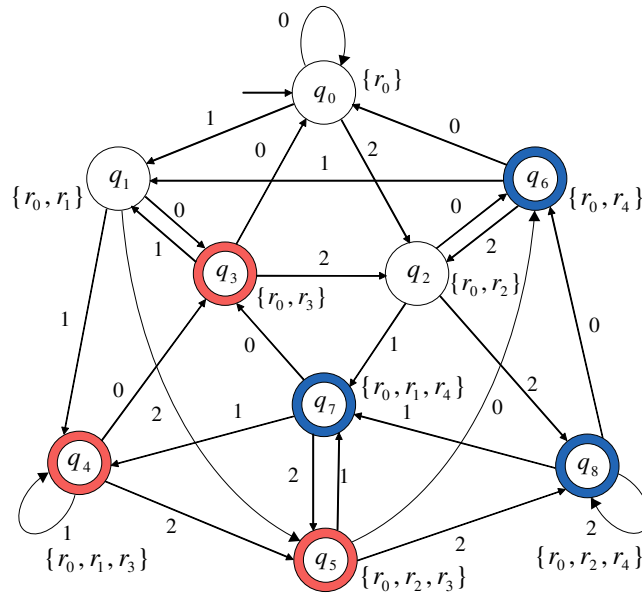


図 36: $L_{3,2}$ を受理する DFA $D_{3,2}$.

なお、非決定性有限オートマトン $N_{k,n}$ を決定性有限オートマトン $D_{k,n}$ に変換すると、その状態数は $|Q| = (k-1)n + 1$ から $|Q'| = k^n$ へ増加すること、ならびに、色無し DFA $D_{2,n}$ の $k-1$ 個の積オートマトンの状態数は、 k に関する指数関数である $(2^n)^{k-1} = 2^{(k-1)n}$ となることに注意されたい。

読者は複数の積オートマトンによって、我々が提案する色付きオートマトンの代用品とすることができるのではないかと思うかもしれない。このことは基本的に正しいが、上で示したようにこれらの記述計算量は莫大である。色付きオートマトンを使用する目的の一つは、そのような積オートマトンの機能を保持しながら与えられた非決定性オートマトンが持つ設計時に生ずる自然な冗長性を有効利用することにある。

次の事実は、部分集合構成法によって構築された DCFA $D_{k,n}$ の状態集合をこれ以上減らすことができないことを示している。

事実 4 純色で色付けされた言語 $L_{k,n} = \sum_{i=1}^k L_{k,n}^{(i)}$ を純色で受理する DCFA は、状態数が k^n 個以上必要である。ここに、各 $i = 1, \dots, k-1$ について、

$$L_{k,n}^{(i)} = \{x \in \{0, 1, \dots, k-1\}^* \mid \text{右端から } x \text{ の } n \text{ 番目の文字は } i \text{ である}\}.$$

(証明) 証明は、2進の場合 [16] を k 進の場合に単純に拡張したものである。逆に、状態数が k^n より少ないような $L_{k,n}$ を純色で受理する DCFA M が存在するとする。すると、長さ n の2つの異なる文字列

$$x = a_1 a_2 \cdots a_n, \quad y = b_1 b_2 \cdots b_n$$

に対して、 M は文字列の右端を読んだ後に全く同じ状態、例えば、 q に入る。なぜなら、 $\{0, 1, \dots, k-1\}$ 上の長さ n の異なる文字列の数は k^n 個だからである。一般性を失うことなく、ある $i = 1, \dots, n$ について、

$$(1) \quad a_i = 0, b_i = j, j \neq 0,$$

または

$$(2) \quad a_i = j_1, b_i = j_2, j_1 \neq j_2, j_1 \neq 0, j_2 \neq 0$$

が成り立つ。ここで、

$$x' = x0^{i-1}, \quad y' = y0^{i-1}$$

とおく。 M は決定性であるから、 x' と y' の両方に対して同じ状態例えば、 q' に入る。(1) の場合、

$$x' \notin L_{k,n}, \quad y' \in L_{k,n}$$

であるから、 M は文字列 x' を読んでも受理状態に入らず、且つ文字列 y' を読んだ場合はある色の受理状態に入らなければならない。(2) の場合、

$$x' \in L_{k,n}^{(j_1)}, \quad y' \in L_{k,n}^{(j_2)}, \quad j_1 \neq j_2$$

であるから、文字列 x' と y' を読んだ場合でそれぞれ異なる色の受理状態に入らなければならない。(1) と (2) のいずれの場合も、仮定と矛盾する。□

定理4がドブリュイングラフの言語による特徴づけであると主張するには、オートマトンと言語との間に更なる厳格な関係付けが必要である。ここに正規言語による有向グラフの特徴づけとは、ある特定のグラフの族が、ある特定の正規言語の族を受理する有限オートマトンの状態遷移図のグラフ構造と完全に一致することを意味する。このとき、正規言語と有限オートマトンは通常一対多で対応するので、そのような特徴づけを得たと言うためには、グラフの族とその正規言語を受理する DCFA の族の状態遷移図が構造的に等価であることを示すことのみならず、そのような DCFA が一意に定まらなければならない。状態数が最小化された DCFA が一意に定まることは、付録 A.3 に色付けされていない通常の DFA の場合と同じ方法で証明できることを記述している。

3.3 NCFA の非混色性に関する決定問題

前節において、NCFA から DCFA への変換は、通常の NFA から DFA への変換する部分集合構成法を自然に拡張した構成法により可能であることを示した。その際、 k 色の NCFA は最悪の場合 2^k 色の DCFA へ変換されてしまうことが明らかになった。従って、CFA を考察するにあたり、NCFA の“非混色性”という概念が重要になる。なぜなら、NCFA により入力文字列を的確に識別するために、受理状態の色が混色していない状況が望まれるからである。このような非混色性は NCFA を DCFA へ変換すれば自ずと明らかになるが、そのような NCFA から DCFA への変換自体に、指数関数的な時間を消費してしまう。

以上のような状況を踏まえて、非混色性に関する計算量理論上の三つの決定問題を提起し、それらの計算複雑さ [22, 31, 32] について検討する。

定義 6 非決定性色付き有限オートマトンの**非混色性検証問題** (Unmixedness Verification problem : UV 問題) は以下のように定義される。

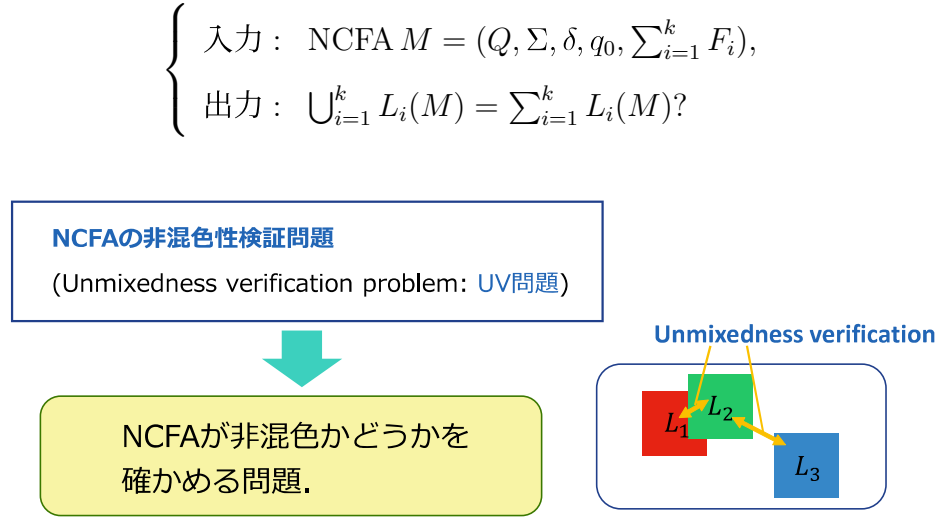


図 37: UV 問題の意味.

UV 問題は、上記の定義のように NCFA が与えられて、 $L_i(M)$ が直和になるかどうかという問題であるが、図 37 に示すように、簡潔に表現すると、NCFA が非混色かどうかを確かめる問題である。図 37 では、 L_1 と L_2 が重なりあっているため、この場合は、UV 問題の出力結果は偽ということになり、入力の NCFA は混色していることを表す。

定理 5 UV 問題は多項式時間で計算できる。

(証明) 最初に、対数コスト基準の下で、問題の補集合 \overline{UV} が非決定性対数領域計算可能な問題のクラス $NLOG$ に属することを示す。

$$\begin{aligned} \langle M \rangle \in \overline{UV} & \\ \Leftrightarrow \bigcup_{i=1}^k L_i(M) & \neq \sum_{i=1}^k L_i(M) \\ \Leftrightarrow \text{ある } i_1 \neq i_2 \text{ に対して、} & L_{i_1}(M) \cap L_{i_2}(M) \neq \emptyset \\ \Leftrightarrow \text{ある } x \in \Sigma^*, i_1 \neq i_2 \text{ に対して、} & x \in L_{i_1}(M), x \in L_{i_2}(M) \\ \Leftrightarrow \text{ある } q_{f_1} \in F_{i_1}, q_{f_2} \in F_{i_2}, x \in \Sigma^*, i_1 \neq i_2 \text{ に対して、} & q_{f_1}, q_{f_2} \in \hat{\delta}(q_0, x) \end{aligned}$$

に注意されたい。ここに、 $\langle M \rangle$ は NCFA M の適切な符号化を表す。 M の記述 $\langle M \rangle$ が与えられると、対数領域に制限されたチューリング機械 M' は 2 つのマーカーを

初期状態 q_0 に配置する．次に，入力文字列 $x \in \Sigma^*$ を推測しながら， M' は遷移関数 δ に従って 2 つの隣接する状態を非決定的に選択し，両方のマーカをこれらの次の状態に移動させる． M が文字列 x を読み終わると， M' は 2 つのマーカが配置されている状態 q_1 と q_2 が $q_1 \in F_{i_1}, q_2 \in F_{i_2}, i_1 \neq i_2$. を満たす場合にのみ受理状態に入る．

$$\overline{UV} \in NLOG$$

より，以下が得られる．

$$UV \in \text{co-NLOG} \subseteq \text{co-P} = P.$$

次に，一様コスト基準の下で，UV 問題の実際的な複雑さについて検討する．NCFA $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^k F_i)$ に関する UV 問題の入力は， M 自身の積オートマトン $M' = (Q \times Q, \Sigma, \delta', (q_0, q_0), F')$ に関して， $L(M') = \emptyset$ かどうかを問う空集合問題例と等価であることは明らかである．ここに，各 $(p, q) \in Q \times Q, a \in \Sigma$ に対して，

$$\delta'((p, q), a) = \delta(p, a) \times \delta(q, a),$$

$$F' = \{(p, q) \in Q \times Q \mid p \in F_{i_1}, q \in F_{i_2}, i_1 \neq i_2, i_1, i_2 \in \{1, 2, \dots, k\}\}.$$

以下では， M の遷移関数 δ は二次元配列の形で与えられると仮定する．その各成分は，現在の状態から次に遷移可能な状態の一覧と， Q の各要素の色などを示すための補助的な一次元タグ配列である． M の状態遷移表のサイズ

$$\sum_{q \in Q} \sum_{a \in \Sigma} |\delta(q, a)|$$

と比較すると，対応する M' の状態遷移表のサイズは以下を満たしている．

$$\begin{aligned} & \sum_{q \in Q} \sum_{p \in Q} \sum_{a \in \Sigma} |\delta(q, a) \times \delta(p, a)| \\ & \leq \sum_{q \in Q} \sum_{p \in Q} \sum_{a \in \Sigma} \sum_{b \in \Sigma} |\delta(q, a)| \cdot |\delta(p, b)| = \left(\sum_{q \in Q} \sum_{a \in \Sigma} |\delta(q, a)| \right)^2. \end{aligned}$$

それ故に、サイズ N の UV 問題の入力は、サイズ $O(N^2)$ の空集合問題に変換される。なお、 M' の空集合問題は、初期状態 (q_0, q_0) から受理状態 F' に決して到達しないことを確認することに相当する。この処理は、通常の線形時間グラフ探索アルゴリズムを使用して達成できる。□

例 9 図 38 は、例 5 で定義された M_0 の積オートマトン M'_0 の状態遷移図を示している。■

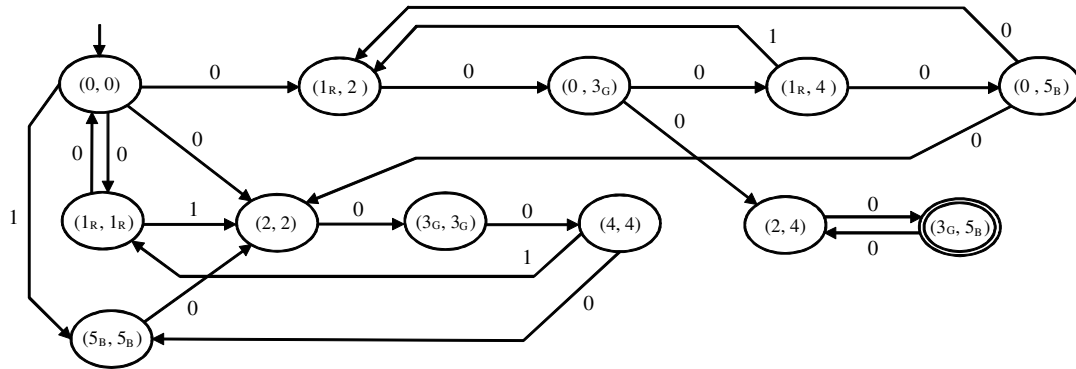


図 38: M_0 の積オートマトン M'_0 .

系 1 UV 問題は $NLOG$ 完全である。

(証明) $NLOG$ が補集合演算について閉じていることは良く知られている [31, 32]. このことと定理 5 の最初の部分から、以下が得られる。

$$UV = \overline{\overline{UV}} \in \text{co-}NLOG = NLOG.$$

次に UV が $NLOG$ 困難であることを示す。NFA の空集合問題例 $L(M) = \emptyset?$ は、以下のようにして UV 問題例 $\langle M' \rangle$ に還元できる。一般性を失うことなく、 M は丁度一つの受理状態 q_f を持つと仮定する。 M' と M の唯一の違いは、 q_f が非受理

状態 q_n に変更され、同じ入力文字例えば、 $a \in \Sigma$ を読むことによって、 q_n から直接到達可能な 2 つの異なる色付き受理状態 q_{f_1}, q_{f_2} が追加されることである。要するに、

$$\begin{aligned}
 L(M) \neq \emptyset &\Leftrightarrow \exists x \in \Sigma^*[x \in L(M)] \\
 &\Leftrightarrow \exists x \in \Sigma^*[q_f \in \delta(q_0, x)] \\
 &\Leftrightarrow \exists x \in \Sigma^*[q_{f_1}, q_{f_2} \in \delta'(q'_0, xa)] \\
 &\Leftrightarrow \exists x \in \Sigma^*[|I(xa)| = 2] \\
 &\Leftrightarrow M' \text{ は混色} \\
 &\Leftrightarrow \langle M' \rangle \in \overline{UV}
 \end{aligned}$$

が成り立つ。ここに、 q_0, q'_0 はそれぞれ M, M' の初期状態である。

M から M' への変更は、決定性対数領域チューリング機械によって成されることは明らかである。すなわち、

$$\begin{aligned}
 \forall P \in NLOG [P \leq_{\log} \overline{UV}] \\
 \Leftrightarrow \forall \overline{P} \in \text{co-NLOG} = NLOG [\overline{P} \leq_{\log} \overline{\overline{UV}} = UV]. \quad \square
 \end{aligned}$$

UV 問題は、与えられた NCFA が同一の入力を異なる色で受理することがないかどうかを問う問題であった。以下では、与えられた色無し有限オートマトンを多色化しようとする際に生ずる問題を扱う。

定義 7 非決定性有限オートマトンの**非混色分割問題** (Unmixed Partitioning problem: UP 問題) は以下のように定義される。

$$\left\{ \begin{array}{l}
 \text{入力: NFA } M = (Q, \Sigma, \delta, q_0, F) \text{ と整数 } k \geq 2, \\
 \text{出力: } F = \sum_{i=1}^k F_i \text{ であるような非混色の NCFA } N = (Q, \Sigma, \delta, q_0, \sum_{i=1}^k F_i) \\
 \text{が存在するか?}
 \end{array} \right.$$

UP 問題は、図 39 に示すように、通常の色無しの NFA が与えられて、その NFA の受理状態を非混色の受理状態に分割した NCFA が存在するかという問題である。図 39 では、 L_1, L_2, L_3 および L_4 が重なりあっておらず混色していないので、ここ

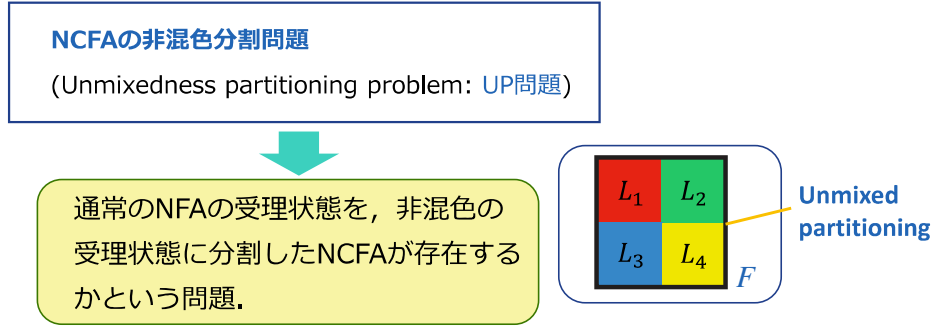


図 39: UP 問題の意味.

で与えられた NFA は $k = 4$ つまり 4 色の非混色の色付き受理状態で分割できるということになる.

事実 5 NCFA $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^k F_i)$ が混色である

$$\begin{aligned}
 &\iff \exists i, j [L_i(M) \cap L_j(M) \neq \emptyset] \\
 &\iff \exists i, j, x [x \in L_i(M) \cap L_j(M)] \\
 &\iff \exists i, j, x [\hat{\delta}(q_0, x) \cap F_i \neq \emptyset, \hat{\delta}(q_0, x) \cap F_j \neq \emptyset] \\
 &\iff \exists i, j, x, p, q [p \in F_i, q \in F_j, p, q \in \hat{\delta}(q_0, x)].
 \end{aligned}$$

定義 8 ある NCFA を $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^k F_i)$ とする. 定理 5 の証明における M の積オートマトン M' から得られる以下の無向グラフ $G = (V, E)$

$$\begin{cases}
 V = Q \\
 E = \{(p, q) \in Q \times Q \mid \exists x \in \Sigma^* [(p, q) \in \hat{\delta}'((q_0, q_0), x)]\} \\
 \quad = \{(p, q) \in Q \times Q \mid \exists x \in \Sigma^* [p, q \in \hat{\delta}(q_0, x)]\}
 \end{cases}$$

は, M の同時到達可能グラフ (simultaneously reachable graph) と呼ばれ, $G_{sr}(M)$ と表される.

M が決定性である場合, $G_{sr}(M)$ には枝がないことに注意されたい. NFA の同時到達可能グラフは, 以下の議論で重要な役割を果たす. 明らかに, 次のことが成り立つ.

命題 3 NCFA M が混色である $\iff G_{sr}(M)$ においてある $i \neq j$ に対して $p \in F_i, q \in F_j$ であるような $(p, q) \in E$ が存在する.

例 10 図 40 に例 5 で定義した M_0 の同時到達可能グラフ $G_{sr}(M_0)$ を示す. このグラフには $3_G \in F_G, 5_B \in F_B$ なる枝 $(3_G, 5_B)$ が存在する. これは M_0 が混色であることを意味する. ■

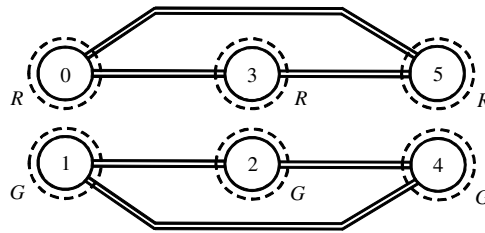


図 40: M_0 の同時到達可能グラフ $G_{sr}(M_0)$.

補題 1 UP 問題の入力 (M, k) が真であるのは, Q の頂点部分集合 F から誘導された $G_{sr}(M)$ の部分グラフ $G_{sr}(M)[F]$ が k 個以上の連結成分を持つときそのときのみである. ここに, F は M の受理状態の集合である.

(証明) もし $G_{sr}(M)[F]$ が k 個以上の連結成分を持つならば, それらの中からある k 個の成分の状態をそれぞれ異なる色で色付けすることができる. M の初期状態から異なる成分の 2 つの状態に同時に到達することはないので, これによって色の混色が発生することはない.

逆に, もし $G_{sr}[F](M)$ の連結成分が k 未満ならば, 色の混色を避けるために同じ成分の全ての状態を同じ色で着色する必要があるため, それらを k 色で着色することはできない. □

例 11 例 5 の M_0 を修正した通常の NFA を $M'' = (Q, \Sigma, \delta, q_0, F)$ とする. ここに, 状態集合は $Q = \{0, 1, 2, 3, 4, 5\}$ で受理状態は $F = \{1, 3, 5\}$ である. 図 40 に示すよ

うに $G_{sr}(M_0'')[F]$ には2つの連結成分しかないため、UP 問題の入力 $(M_0'', 3)$ は偽である。 ■

定理 6 UP 問題は多項式時間で計算することができる。

(証明) 定理5の証明に示されているように、NCFA M の積オートマトンは多項式時間で構築できる。このオートマトンを線形探索することにより、同時到達可能グラフ $G_{sr}(M)$ と誘導グラフ $G_{sr}(M)[F]$ を取得できる。補題1で採用された連結成分の列挙は、通常のグラフ探索アルゴリズムを使用して線形時間で実行できる。 □

続いて、元の受理状態集合 F_0 の他に M の非受理状態から k 個の非混色な状態集合 F_1, \dots, F_k を選択できるかどうかを問う問題について検討する。この場合、これらの追加された k 色は、受理状況を意味するのではなく、非受理状態で停止したときに入力に関する補足的な（例えば、エラーに関連する）情報を提供する。

定義 9 非決定性有限オートマトンの**非混色拡張問題** (Unmixed Extension problem: UE 問題) は次のように定義される (図 41 参照)。

$$\left\{ \begin{array}{l} \text{入力: NFA } M = (Q, \Sigma, \delta, q_0, F_0) \text{ と整数 } k \geq 1, \\ \text{出力: 非混色の NCFA } N = (Q, \Sigma, \delta, q_0, \sum_{i=0}^k F_i) \text{ が存在するか?} \end{array} \right.$$

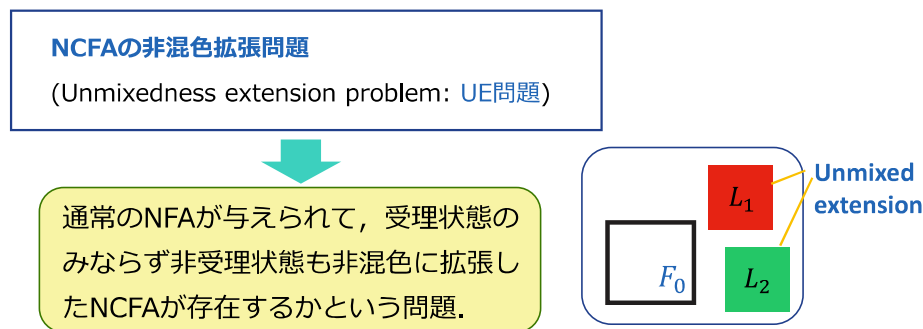


図 41: UE 問題の意味.

図 41 に示すように、UE 問題は通常の色無しの NFA が与えられて、受理状態のみ

ならず非受理状態も非混色に拡張した NCFA が存在するかという問題である。図 41 では、受理状態集合 F_0 を除く非受理状態を赤色の色付き言語 L_1 と緑色の色付き言語 L_2 で非混色に拡張できている状態を表している。

定義 10 [10, 33, 34] グラフ $G = (V, E)$ について、もし $\forall u, v \in I[(u, v) \notin E]$ ならば、 $I \subseteq V$ を G の独立集合と呼ぶ。

定義 11 無向グラフの独立集合問題 (IS と略記) は次のように定義される。

$$\left\{ \begin{array}{l} \text{入力: グラフ } G \text{ と整数 } k \geq 2, \\ \text{出力: } G \text{ にサイズ } k \text{ の独立集合が存在するか?} \end{array} \right.$$

補題 2 UE 問題の入力 (M, k) は、IS 問題の入力 $(G_{sr}(M)[\overline{F_0 \cup N(F_0)}], k)$ が真のときそのときのみ真になる。ここに、 $G_{sr}(M)[\overline{F_0 \cup N(F_0)}]$ は、 $G_{sr}(M)$ の F_0 とその隣接点集合 $N(F_0)$ の和集合の補集合 $Q - (F_0 \cup N(F_0))$ から誘導された $G_{sr}(M)$ の部分グラフである。

(証明) F_0 の頂点とその隣接する頂点 $N(F_0)$ が元の $G_{sr}(M)$ から削除された M の同時到達可能グラフが k 個の独立点を持っていると仮定する。このことはそれらが互いに直接隣接しているわけではなく、 F_0 の頂点とも隣接していないことを意味する。従って、色の混色なしで、これらの k 状態を k 色の異なる色で色付けし、 F_0 の状態を他の 1 つの色で色付けすることができる。

逆に、 $G_{sr}(M)$ の頂点から F_0 以外の集合 $F_1 + \dots + F_k$ を選択するとき、 $i \neq j$ なる任意の $p \in F_i$ と $q \in F_j$ を隣接させてはいけない。このように、頂点 F_0 とその隣接点が削除された $G_{sr}(M)$ は、少なくとも k 個の要素で構成される独立集合を持つ必要がある。 \square

定理 7 UE 問題は NP 完全である。

(証明) 最初に、IS 問題を多項式時間でこの問題に還元することにより、問題の NP 困難性を示す。

IS 問題の入力を (G, k) とする. ここに $G = (\{v_1, \dots, v_n\}, E)$ である. (G, k) を UE 問題の入力 $(M, k + 1)$ へ変換する. ここに, $M = (\{p_0, q_1, \dots, q_n\}, \Sigma, \delta, p_0, \emptyset)$, $\Sigma = \{a_1, \dots, a_n\} \cup \{a_{ij} \mid (i, j) \in I_E\}$, $I_E \triangleq \{(i, j) \mid (v_i, v_j) \in E\}$, $\delta(p_0, a_i) = \{q_i\}, i = 1, \dots, n$, ならびに $\delta(p_0, a_{ij}) = \{q_i, q_j\}, (i, j) \in I_E$ である.

$G_{sr}(M)$ は, M の初期状態 p_0 に対応する 1 つの孤立した頂点が追加された G であり, M の入力アルファベットのサイズは $O(|G|)$ で抑えられることは明らかである. ここに, $|G|$ は G のサイズを表す.

UE 問題の非決定性多項式時間での可解性は, 補題 2 から得られる. □

例 12 図 42 は, サイズ 2 の独立集合を持つグラフ G から, それに対応する NFA M への変換例を示している. M の同時到達可能グラフ $G_{sr}(M)$ は, 初期状態の頂点が追加されていることを除いて, G と同じであることを注意されたい. ■

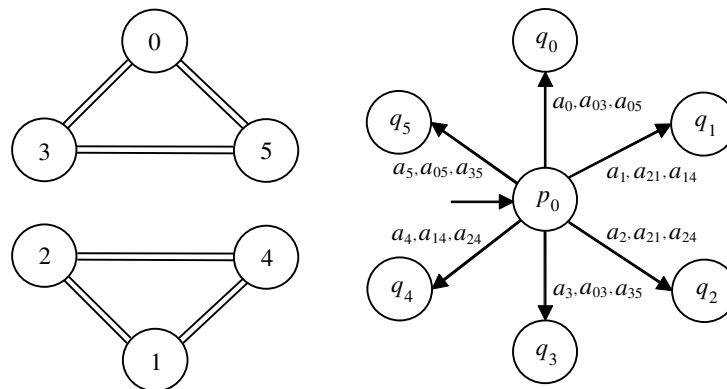


図 42: グラフ G の NFA への変換.

3.4 ε 遷移付き NCFA とその適用例

本節では、 ε 遷移付きの NCFA を定義し、通常の色無し有限オートマトンの色付き有限オートマトンに変換することにより生じる一度に多数の言語が識別可能というメリットを例示する。

定義 12 次のように表される 5 項組 $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^k F_i)$ を ε 遷移付き非決定性色付き有限オートマトンと呼び、 ε -NCFA と略記する。

1. Q は状態の有限集合.
2. Σ は入力文字列の有限集合.
3. δ は遷移関数で $Q \times (\Sigma \cup \{\varepsilon\})$ から 2^Q への関数.
4. $q_0 \in Q$ は初期状態.
5. $\sum_{i=1}^k F_i \subseteq Q$ は色付き受理状態の集合で、 F_i は i 色の受理状態の集合.

q から ε 遷移を経て到達可能な状態の集合を ε -Closure(q) と表す. また、 $p, q \in Q$ に対して、 ε -Closure(p, q) $\triangleq \{(p', q') \mid p' \in \varepsilon$ -Closure(p), $q' \in \varepsilon$ -Closure(q)\} とする.

定義 13 ε -NCFA $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^k F_i)$ とし、5 項組 $M' = (Q', \Sigma, \delta', q'_0, F')$ を M 自身の直積オートマトンと呼ぶ. ここに、

$$\begin{aligned} Q' &= \{\varepsilon\text{-Closure}(p, q) \mid (p, q) \in Q^2\}, \\ \delta'(S, a) &= \{\varepsilon\text{-Closure}(p', q') \mid p' \in \delta(p, a), q' \in \delta(q, a), (p, q) \in S\}, S \in Q', a \in \Sigma, \\ q'_0 &= \varepsilon\text{-Closure}(q_0, q_0), \\ F' &= \{S \in Q' \mid \text{ある } (p, q) \in S \text{ に対して, } (p, q) \in S, p \in F_{i_1}, \\ &\quad q \in F_{i_2}, i_1 \neq i_2, i_1, i_2 \in \{1, 2, \dots, k\}\}. \end{aligned}$$

上記の直積オートマトンの定義は、定義 5 における色付き有限オートマトンに対して ε 遷移を許しているため、原型のものより僅かに複雑になっている。

$|Q'| \leq |Q|^2$ より、 M' が M から多項式時間で構成できることは容易に分かる。

定義 14 ε -NCFA $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^k F_i)$ とする. M の直積オートマトン M' から得られる無向グラフ $G = (V, E)$

$$\begin{cases} V = Q \\ E = \{(p, q) \in Q \times Q \mid \exists x \in \Sigma^* [(p, q) \in \hat{\delta}'(q'_0, x)]\} \\ \quad = \{(p, q) \in Q \times Q \mid \exists x \in \Sigma^* [p, q \in \hat{\delta}(q_0, x)]\} \end{cases}$$

は M の同時到達可能グラフと呼ばれ, $G_{sr}(M)$ と表記する.

例 13 図 43 は, 良く知られた定番の教科書 [16] に登場する (色付きでない) 非決定性有限オートマトン M_d の例である. M_d は以下の (1), (2), 及び (3) を満たす 10 進数の数字列を受理する. (1) 正または負の符号を持っても良い. (2) 整数部または少数部を持たなくても良いが, どちらかは持つ必要がある. (3) 小数点を必ず持つ. M_d から図 44 に示す積オートマトン M'_d を構成した後, 図 45 に示すような M_d の同時到達可能グラフ $G_{sr}(M_d)$ を取得できる.

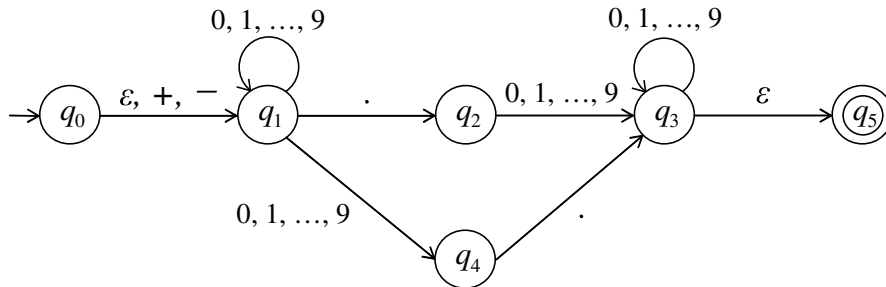


図 43: 10 進数を受け取る ε -NFA M_d [16].

図 45 から, M_d の状態 q_0, q_4 および q_5 にそれぞれ対応する 3 つの独立した頂点 v_0, v_4 および v_5 を選択できる. それらをそれぞれ R, G, B の 3 つの異なる色で色付けすると, 対応する言語 $L_R(M_d), L_G(M_d)$ および $L_B(M_d)$ は, 互いに非混色となる. ここに, $L_R(M_d) = \{\varepsilon\}$, $L_G(M_d)$ は小数点無しの 1 桁以上の整数であり, $L_B(M_d) = L(M_d)$ である.

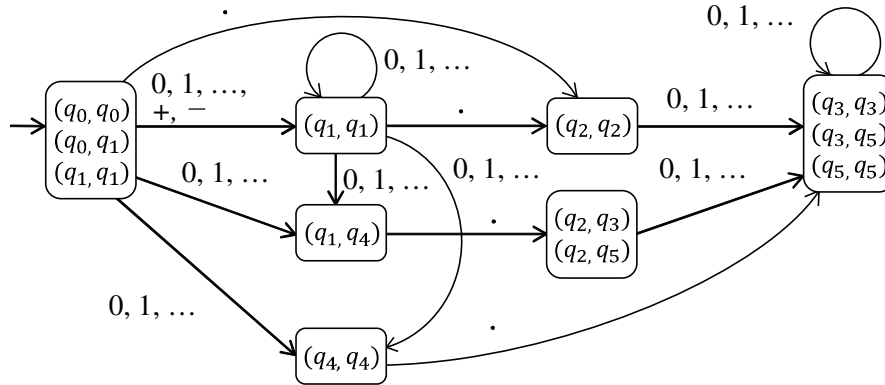


図 44: M_d (図 43) の積オートマトン M'_d .

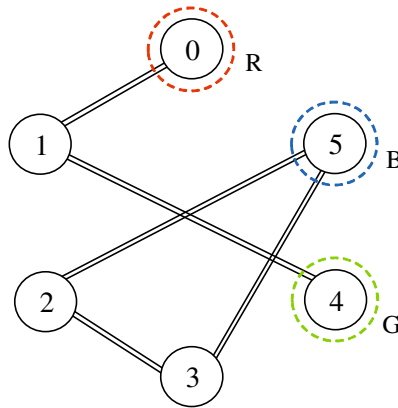


図 45: M_d (図 43) の同時到達可能グラフ $G_{sr}(M_d)$.

3.5 色付き正規表現

後述するように、NCFA を正規表現エンジンへ応用するためには、NCFA を機械的に色が付いた正規表現に変換する必要がある。そこで本研究では、色付き正規表現、色付き正規表現が表す色付き言語、ならびに NCFA を色付き正規表現に変換するアルゴリズムを新たに導入する。以下にそれらの定義と第 3.7.1 項にアルゴリズムを示す。

定義 15 アルファベット Σ 上の色数 k の色付き正規表現 (Colored Regular Express-

sions) は次のように定義される.

1. $\varepsilon, \triangleleft_1, \dots, \triangleleft_k, \emptyset, a \in \Sigma$ はそれぞれ色付き正規表現.
2. r, s が色付き正規表現ならば, $r + s, rs, r^*$ はそれぞれ色付き正規表現.

記法 1 色付き正規表現 r に含まれる (出現位置が異なる) すべての終端受理状態記号を $\triangleleft_1, \dots, \triangleleft_K$ とする. $\triangleleft_j (1 \leq j \leq K)$ 以外のすべての終端受理状態記号を削除した (単色の) 正規表現を $r|_j$ と記す. 更に, すべての終端受理状態記号を削除した (無色の) 正規表現を $r|_0$ と記す.

定義 16 終端受理状態記号を $K = l_1 + \dots + l_k$ 個含む k 色の正規表現 r が表す色付き言語 (ベクトル) は $(L_1(r), \dots, L_k(r)) = (\bigcup_{j=1}^{l_1} L(r|_j), \dots, \bigcup_{j=1}^{l_k} L(r|_j))$ と定義される. ここに, $r|_j, 1 \leq j \leq K$ が表す言語 $L(r|_j)$ は以下のように定まる.

- $r|_j = s + t$ ならば, $L(r|_j) = L(s|_j) \cup L(t|_j)$.
- $r|_j = st, t \neq \varepsilon$ ならば, $L(r|_j) = \begin{cases} L(s|_j), \triangleleft_i \notin t \text{ のとき,} \\ L(s), \triangleleft_i = t \text{ のとき.} \end{cases}$
- $r|_j = s^*$ ならば, $L(r|_j) = L(s^*|_0) \cdot L(s|_j)$.
- $r|_j = \varepsilon$ ならば, $L(r|_j) = \emptyset$.

例 14 $r = \triangleleft_R(\oplus + \varepsilon) d^*(\bullet d + d \triangleleft_G \bullet) d^* \triangleleft_B$ のとき, $\sum_{i=R,G,B} r|_i = r|_R + r|_G + r|_B$
 $= \triangleleft_R(\oplus + \varepsilon) d^*(\bullet d + d \bullet) d^* + (\oplus + \varepsilon) d^*(\bullet d + d \triangleleft_G \bullet) d^* + (\oplus + \varepsilon) d^*(\bullet d + d \bullet) d^* \triangleleft_B$
 $= \triangleleft_R + (\oplus + \varepsilon) d^*(d \triangleleft_G) + (\oplus + \varepsilon) d^*(\bullet d + d \bullet) d^* \triangleleft_B$. 従って,
 $L(r|_R) = \varepsilon, L(r|_G) = (\oplus + \varepsilon) d^* d, L(r|_B) = (\oplus + \varepsilon) d^*(\bullet d + d \bullet) d^*$.

3.6 非決定性色付きビュッヒオートマトン (NCBA)

NCFA のモデル検査への応用の可能性を後述するが、モデル検査を考察する上で、有限オートマトンや色付き有限オートマトンで考えていた有限言語だけでなく、無限言語を考える必要性が生じる。無限言語を考えるにあたり、それらを受理対象とするビュッヒオートマトンを用いる必要がある。そこで我々は、色付きの概念をビュッヒオートマトンへ拡張させた色付きビュッヒオートマトン (NCBA) を新たに導入する。以下に、NCBA の定義を与える。

定義 17 [39] アルファベット Σ 上の無限語とは、記号 $A_i \in \Sigma$ の無限系列 $A_0A_1A_2\cdots$ である。 Σ^ω は Σ 上の全ての無限語の集合を表す。ギリシャ文字 σ が無限語を記すのに使われる。 Σ^ω の部分集合は無限言語、あるいはまた ω 言語と呼ばれる。以下では、言語とは $\Sigma^* \cup \Sigma^\omega$ の部分集合を言う。

正規表現における Kleene 閉包操作 (有限回の繰り返し接続演算) は無限回に拡張され、 ω で表される。例えば、有限語 AB の無限繰り返し操作は無限語 $ABABABAB\cdots$ (無限に続く) を生成し、 $(AB)^\omega$ と記される。特に、空列の場合には $\varepsilon^\omega = \varepsilon$ 。

定義 18 [39] 原始命題集合 AP 上の線形時間特性 (property) とは、 $(2^{AP})^\omega$ の部分集合を言う。

定義 19 [39] 遷移系 (transition system) とは、6 個組 $TS = (S, Act, \rightarrow, I, AP, L)$ である。ここに、

1. S は状態の集合,
2. Act はアクションの集合,
3. $\rightarrow \subseteq S \times Act \times S$ は遷移関係,
4. $I \subseteq S$ は初期状態の集合,
5. $L : S \rightarrow 2^{AP}$ はラベル付け関数である。

利便性のため, $(s, \alpha, s') \in \rightarrow$ の代わりに $s \xrightarrow{\alpha} s'$ と記す. ラベル付け関数 L は原始命題 $L(s) \in 2^{AP}$ を状態 s に結び付ける. 直観的には, $L(s)$ は状態 s で充足されるような原始命題 $a \in AP$ の集合を表す. TS の (無限あるいは有限) パス断片 π とは $s_i \xrightarrow{\alpha} s_{i+1}, i \geq 0$ を満たす状態系列 $s_0 s_1 s_2 \dots$ を言う. $L(s_0)L(s_1)L(s_2)\dots$ を π の軌跡 ($trace(\pi)$) と呼ぶ.

定義 20 [39] TS を AP 上の遷移系とし, P を AP 上の特性とする. $Traces(TS) \triangleq \bigcup_{s \in I} trace(Paths(s)) \subseteq P$ が成り立つときそのときに限り, TS は P を充足すると言う ($TS \models P$ と記す).

例 15 図 46 はセマフォを用いて 2 つのプロセス Proc1 と Proc2 の相互排除問題を解決する遷移系 $TS_{sm\&pr} = (S, Act, \rightarrow, I, AP, L)$ である. ただし, 両者ともに待機状態にある場合には, Proc1 が臨界区域に入ることを優先させている. ここに,

- $S = (\{n_1, w_1, c_1\} \times \{n_2, w_2, c_2\} - \{(c_1, c_2)\}) = \{(n_1, n_2), (w_1, n_2), (c_1, n_2), (n_1, w_2), (w_1, w_2), (c_1, w_2), (n_1, c_2), (w_1, c_2)\}$
- $Act = \{req_1, ent_1, rel_1, req_2, ent_2, rel_2\},$
 $I = \{(n_1 n_2)\},$
 $AP = \{n_1, w_1, c_1, n_2, w_2, c_2\},$
- $(n_1, n_2) \xrightarrow{req_1} (w_1, n_2), \dots, (w_1, c_2) \xrightarrow{rel_2} (w_1, n_2),$
- $L(n_1, n_2) = \{n_1, n_2\}, L(w_1, n_2) = \{w_1, n_2\}, \dots, L(w_1, c_2) = \{w_1, c_2\}$

例えば, $\pi \triangleq ((n_1, n_2)(n_1, w_2)(n_1, c_1))^\omega$ とおくと, $trace(\pi) = (\{n_1, n_2\}\{n_1, w_2\}\{n_1, c_2\})^\omega$.

また, $\bar{P}_{mutex} \triangleq True^* \{c_1, c_2\} True^\omega$ とおくと, $Traces(TS_{sm\&pr}) \subseteq P_{mutex}$ すなわち, $TS_{sm\&pr} \models P_{mutex}$ が成り立つ. 従って, 2 つのプロセスが同時に臨界区域に入ることは決してない.

■

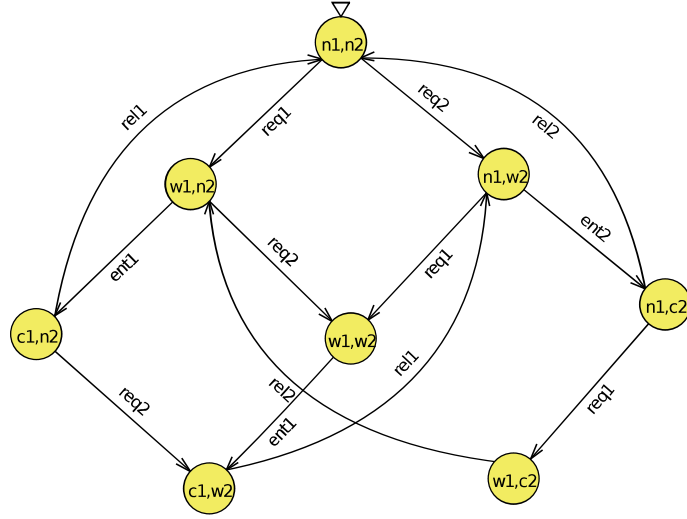


図 46: Proc1 と Proc2 の相互排除問題を解決するモデルの遷移系 TS .

定義 21 色付き非決定性ビュッヒオートマトン (NCBA) とは, 5 個組 $M = (Q, \Sigma, \delta, Q_0, \Sigma_{i=1}^k F_i)$ である. ここに,

1. Q は状態の有限集合,
2. Σ はアルファベット,
3. $\delta: Q \times \Sigma \rightarrow 2^Q$ は遷移関数,
4. $Q_0 \subseteq Q$ は初期状態の集合,
5. $\Sigma_{i=1}^k F_i \subseteq \Sigma$ は受理状態の集合である.

入力 $\sigma = A_0 A_1 A_2 \dots \in \Sigma^\omega$ に対する計算系列 (あるいは走行) とは, $q_0 \in Q_0$ かつ $\delta(q_i, A_i) \ni q_{i+1}, i \geq 0$ を満たすような M の無限状態系列 $q_0 q_1 q_2 \dots$ を言う. 計算系列 $q_0 q_1 q_2 \dots$ は無数に多くの $j \in \mathbb{N}$ に対して $q_j \in F_i$ ならば i 色で受理していると言う. M により i 色で受理される言語とは,

$$L_{\omega i}(M) = \{ \sigma \in \Sigma^\omega \mid \sigma \text{ に対する } i \text{ 色の受理計算系列が存在する} \}$$

である. M で受理される (統合された) 言語とは

$$L_\omega(M) \triangleq \bigcup_{i=1}^k L_{\omega i}(M)$$

である。 $L_\omega(M) = \Sigma_{i=1}^k L_{\omega_i}(M)$ が成り立つとき、 $L_\omega(M)$ は非混色であると言い、 M は $L_\omega(M)$ を色識別受理すると言う。この NCBA の具体的な適用例については、第 3.7.2 項で示す。

3.7 色付き有限オートマトンの応用

この節では、色付き有限オートマトンの応用の可能性を示す。

3.7.1 正規表現エンジンへの応用

正規表現 [16, 17] は文字列の集合を一つの文字列で表現することができ、文字列検索などの文字列操作が必要なプログラミングでは無くてはならない存在になっている [52]。

図 47 はプログラミング言語 Nim [40, 54] の標準ライブラリーである正規表現モジュール `re` をインポートすることで使用可能となるテンプレート関数 `s =~ p` の使用例である。検査対象文字列 `s` が正規表現パターン `p` にマッチした場合にはキャプチャされた部分文字列が配列 `matches` に格納されるが、この関数はあくまで真理値関数であり、特定の文字列 `s` が様々な異なるパターン `p, p', ...` のどれと照合するか調べるためには、この例のように `if else` 文を繰り返すしか方法が無い。一方、我々が提案する色付き有限オートマトンの機能が追加された場合には、プログラム 1 のように `case` 文によって一つの正規表現パターンによって他種類の判別が可能となるはずである。

ところで、正規表現を解釈し実際にパターンマッチを行う正規表現エンジンの実装においては、正規表現をそれと等価な ϵ 付き非決定性有限オートマトンに変換した後に（更に決定性のものに変換しないで）そのまま全探索シミュレートし

```

template `=~`(s: string; pattern: Regex): untyped

This calls match with an implicit declared matches array that can be used in the scope of the =~
call:

Example:

proc parse(line: string): string =
  if line =~ re"\s*(\w+)\s*=\s*(\w+)": # matches a key=value pair:
    result = $(matches[0], matches[1])
  elif line =~ re"\s*(\#.*)": # matches a comment
    # note that the implicit `matches` array is different from 1st branch
    result = $(matches[0],)
  else: doAssert false
  doAssert not declared(matches)
doAssert parse("NAME = LENA") == ""("NAME", "LENA")""
doAssert parse(" # comment ... ") == ""("# comment ... ",)""

Source Edit

```

図 47: Nim 言語におけるテンプレート関数 `s =~ p` の使用例 [40].

ている場合が多い [41]. このような現状から考えると、我々が提案する色付き有限オートマトンを既存のシステムで実現すること自体は決して困難ではない。なぜなら、システム内で各状態が受理状態かどうかを表しているフラグ情報を、何色の受理状態であるのかを表すタグ情報に変えれば済むからである。

プログラム 1: 照合関数 `s =~ p` での case 文の実現

```

1 case line =~ re "_R[+-]?[0-9]*([\.[0-9]| [0-9]_G\.) [0-9]*_B"
2 of R: echo "empty string"
3 of G: echo "an integer number"
4 of B: echo "a fractional number"
5 else: echo "error"

```

一方、ユーザーインターフェースの観点から言えば、既存システムへ投入できるのは有向グラフとしての色付き有限オートマトンではなく、文字列としての正規表現に限られている。そこで、与えられた色付き有限オートマトンをプログラ

ム 1 に例示したような“色付き正規表現”に変換する必要がある。そのアルゴリズムを以下に示す。

色付き有限オートマトンから色付き正規表現への変換法

1. 与えられた色付き有限オートマトン M を以下 (図 49 参照) のように変形色付き遷移図 M' に変換する: 各色付き受理状態を非受理状態化し, かつ対応する終端受理状態への ε 遷移を追加する. なお, 色付き受理状態へ入る ε 遷移が存在する場合には, その ε 遷移の始点となっている状態も同色の色付き受理状態とする.
2. 変形色付き遷移図 M' に対応する連立正規表現方程式を立てる.
3. Arden の補題 [$X \rightarrow rX + s$ の解は $X = r^*s$] を用いて, 状態変数を消去していく. ただし, 必要に応じて式 $X \rightarrow s + \triangleleft$ を式 $X \rightarrow \triangleleft s$ (遷移式) に置き換えることができる (変形消去法).
4. 変数 q の解は状態 q を初期状態と見なしたときに M で受理される色付き言語の正規表現を表す.

本変換アルゴリズムの正当性は色付き正規表現から色付き有限オートマトンへの逆変換 (たとえば, Thompson 構成法 [53]) によって元の有限オートマトンと等価なオートマトンが再現できることにより言える。

図 43 に示す 10 進数を受け取る ε -NFA M_d は, 図 48 に示すように UE 問題により q_0, q_4, q_5 を非混色の色付き受理状態に拡張することができる。ここで, q_3 と q_5 は ε で繋がっているため, 議論を簡単化するために q_5 を省略し, q_3 を色付き受理状態とし, また, $0, 1, \dots, 9$ は一括して d と表している。

例 16 図 48 に示す UE 問題により非混色の受理状態を拡張させた 10 進数を受け取る NFA M を色付き正規表現へ変換する。

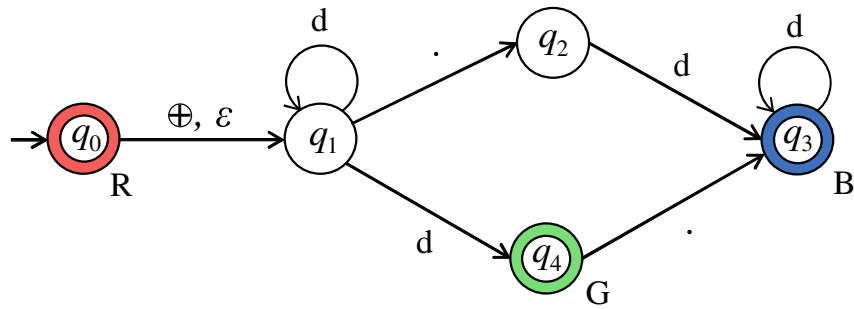


図 48: 非混色の受理状態を拡張させた 10 進数を受理する NCFA M .

ステップ 1 : 与えられた図 48 の NCFA M を図 49 に示す変形 NCFA M' に変換する. 図 48 各色付き受理状態が ε 遷移によって終端受理状態を表す \triangleleft に遷移していることに注意されたい.

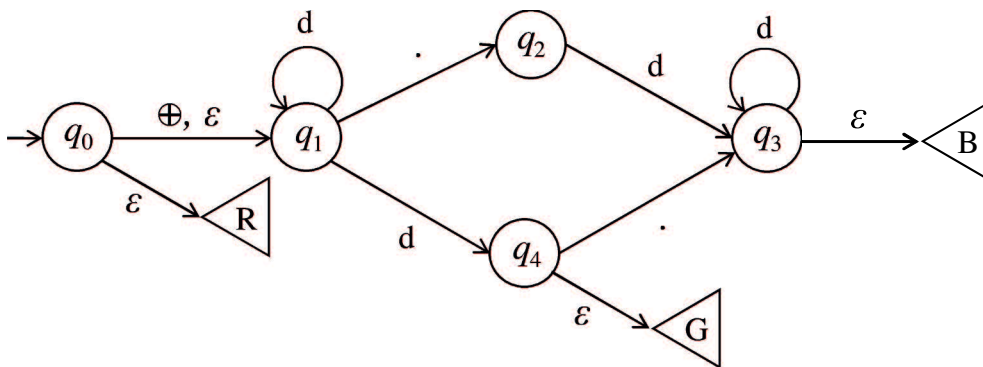


図 49: M の変形色付き遷移図 M' .

ステップ 2 : 変形色付き遷移図 M' の連立方程式を立てる.

$$\left\{ \begin{array}{l} q_0 \rightarrow (\oplus + \varepsilon) q_1 + \varepsilon \triangleleft_R \\ q_1 \rightarrow d q_1 + \cdot q_2 + d q_4 \\ q_2 \rightarrow d q_3 \\ q_3 \rightarrow d q_3 + \varepsilon \triangleleft_B \\ q_4 \rightarrow \cdot q_3 + \varepsilon \triangleleft_G \end{array} \right. = \left\{ \begin{array}{l} q_0 \rightarrow (\oplus + \varepsilon) q_1 + \triangleleft_R \quad \cdots (1) \\ q_1 \rightarrow d q_1 + \cdot q_2 + d q_4 \quad \cdots (2) \\ q_2 \rightarrow d q_3 \quad \cdots (3) \\ q_3 \rightarrow d q_3 + \triangleleft_B \quad \cdots (4) \\ q_4 \rightarrow \cdot q_3 + \triangleleft_G \quad \cdots (5) \end{array} \right.$$

ただし, q_0 が初期状態である.

ステップ3及び4：変形消去法を用いて、状態変数を消去していく。

式(4)に Arden の補題を適用して、 $q_3 = d^* \triangleleft_B \cdots (4')$.

式(4')を式(3)に代入して、 $q_2 = d d^* \triangleleft_B = d^+ \triangleleft_B \cdots (3')$.

式(4')を式(5)に代入して、 $q_4 \rightarrow \cdot d^* \triangleleft_B + \triangleleft_G$

従って、遷移式への置き換えにより、 $q_4 = \triangleleft_G \cdot d^* \triangleleft_B \cdots (5')$.

式(5'),(3')を式(2)に代入して、

$$q_1 \rightarrow d q_1 + \cdot d d^* \triangleleft_B + d \triangleleft_G \cdot d^* \triangleleft_B$$

従って、Arden の補題により、

$$q_1 = d^*(\cdot d d^* \triangleleft_B + d \triangleleft_G \cdot d^* \triangleleft_B) = d^*(\cdot d + d \triangleleft_G \cdot) d^* \triangleleft_B \cdots (2')$$

式(2')を式(1)に代入して、

$$q_0 \rightarrow (\oplus + \varepsilon) d^*(\cdot d + d \triangleleft_G \cdot) d^* \triangleleft_B + \triangleleft_R$$

従って、遷移式への置き換えにより、

$$q_0 = \triangleleft_R (\oplus + \varepsilon) d^*(\cdot d + d \triangleleft_G \cdot) d^* \triangleleft_B \quad \blacksquare$$

なお、上記の変換例において、遷移式への置き換えを使用しない場合には、他の色付き正規表現 $q_0 = (\oplus + \varepsilon) \{d^*(\cdot d + d \cdot) d^* \triangleleft_B + d^+ \triangleleft_G\} + \triangleleft_R = (\oplus + \varepsilon) d^*(\cdot d + d \cdot) d^* \triangleleft_B + (\oplus + \varepsilon) d^+ \triangleleft_G + \triangleleft_R$ が得られる。

赤色の受理状態 q_0 は空列、緑色の受理状態 q_4 は整数、青色の受理状態 q_3 は小数点を含む実数を受理することを意味している。従って、求めた色付き正規表現 $\triangleleft_R (\oplus + \varepsilon) d^*(\cdot d + d \triangleleft_G \cdot) d^* \triangleleft_B$ を用いると、図47のような従来の照合関数をプログラム1のように case 文が使用可能なものに拡張することができ、一度に多数の情報を得ることができるようになる。ここで例えば、 $_R$ は終端受理状態 \triangleleft_R を表す。

例 17 例5の NCFA M_0 に色付き有限オートマトンから色付き正規表現への変換

法を適用して、 M_0 の色付き正規表現を求める。

$$\left\{ \begin{array}{l} q_0 \rightarrow 0(q_{1R} + q_2) + 1q_{5B} \cdots (1) \\ q_{1R} \rightarrow 0q_0 + 1q_2 + \triangleleft_R \cdots (2) \\ q_2 \rightarrow 0q_{3G} \cdots (3) \\ q_{3G} \rightarrow 0q_4 + \triangleleft_G \cdots (4) \\ q_4 \rightarrow 0q_{5B} + 1q_{1R} \cdots (5) \\ q_{5B} \rightarrow 0q_2 + \triangleleft_B \cdots (6) \end{array} \right.$$

式(2), (6) を (1) に代入して,

$$q_0 \rightarrow 0(0q_0 + 1q_2 + \triangleleft_R + q_2) + 1(0q_2 + \triangleleft_B) = 00q_0 + (01 + 0 + 10)q_2 + 0\triangleleft_R + 1\triangleleft_B \cdots (1')$$

式(2), (6) を (5) に代入して,

$$q_4 \rightarrow 0(0q_2 + \triangleleft_B) + 1(0q_0 + 1q_2 + \triangleleft_R) = 10q_0 + (00 + 11)q_2 + 0\triangleleft_B + 1\triangleleft_R \cdots (5')$$

式(5') を (4) に代入して,

$$q_{3G} \rightarrow 0(10q_0 + (00 + 11)q_2 + 0\triangleleft_B + 1\triangleleft_R) + \triangleleft_G = 010q_0 + 0(00 + 11)q_2 + 00\triangleleft_B + 01\triangleleft_R + \triangleleft_G \cdots (4')$$

式(4') を式(3) に代入して,

$$q_2 \rightarrow 0(010q_0 + 0(00 + 11)q_2 + 00\triangleleft_B + 01\triangleleft_R + \triangleleft_G) = 0010q_0 + 00(00 + 11)q_2 + 000\triangleleft_B + 001\triangleleft_R + 0\triangleleft_G \cdots (3')$$

従って, $q_2 = (0000 + 0011)^*(0010q_0 + 000\triangleleft_B + 001\triangleleft_R + 0\triangleleft_G) \cdots (3'')$

式(3'') を式(1') に代入して,

$$\begin{aligned} q_0 &\rightarrow 00q_0 + (01 + 0 + 10)(0000 + 0011)^*(0010q_0 + 000\triangleleft_B + 001\triangleleft_R + 0\triangleleft_G) + 0\triangleleft_R + 1\triangleleft_B \\ &= (00 + (01 + 0 + 10)(0000 + 0011)^*0010)q_0 + (01 + 0 + 10)(0000 + 0011)^*(000\triangleleft_B \\ &\quad + 001\triangleleft_R + 0\triangleleft_G) + 0\triangleleft_R + 1\triangleleft_B. \end{aligned}$$

従って,

$$q_0 = (00 + (01 + 0 + 10)(0000 + 0011)^*0010)^*((01 + 0 + 10)(0000 + 0011)^*(000\triangleleft_B + 001\triangleleft_R + 0\triangleleft_G) + 0\triangleleft_R + 1\triangleleft_B).$$

よって,

$$\begin{aligned}
L_R(M_0) &= (00+(01+0+10)(0000+0011)*0010)*((01+0+10)(0000+0011)*001 \triangleleft_R \\
&\quad + 0 \triangleleft_R) \\
&= (00+(01+0+10)(0000+0011)*0010)*((01+0+10)(0000+0011)*001+ \\
&\quad 0).
\end{aligned}$$

3.7.2 モデル検査への応用

モデル検査 [19] においては、検査対象のシステム (system) と望まれる仕様 (specification) を検査器に投入し、システムが仕様を満たすか満たさないかを判定する。仕様を満たさない場合は反例 (エラー軌跡) を出力する。以下に、色付きオートマトンをモデル検査に用いることにより、仕様を満たさない場合において通常よりも詳しい付加情報が与えられる例を示す。

例 18 図 43 に示した 10 進数を受理する ε -NFA が例 13 において非混色な色付き有限オートマトンに変換できることを示したが、そのうち状態 q_3 と状態 q_4 を非混色な受理状態とする ε -NCFA M_m を図 50 に示す。システム M_m にとって望ましい仕様を“値 0 には \pm 符号が付かない”こととし、それを否定する望ましくない仕様を表す DCFA \overline{M}_s を図 51 に示す。 \overline{M}_s に入力された 10 進数は値 0 に小数点が付く場合と付かない場合で色が異なる受理状態に入る。モデル検査では、 M_m と \overline{M}_s の積オートマトンを構成し、受理状態への到達可能性を調べることになる。 M_m と \overline{M}_s の積オートマトン NCFA M_c を図 52 に示す。この積オートマトンには初期状態から受理状態へのパスが存在するので、 $L(M_m) \cap L(\overline{M}_s) \neq \emptyset$ となり、システムは与えられた仕様を満たさないことが分かる。従来のモデル検査では、ここで一つの反例を与えれば十分であるが、色付き受理状態の概念を導入したことにより、更なる付加情報を得ることができている。すなわち、図 52 の 2 つの受理状態 $(q_4, r_1), (q_3, r_2)$ に着目すると、前者は小数点が付かない色付き受理状態のペアであり、後者は小数点が付く色付き受理状態のペアとなっている。このことより、現システムは値 0 に小数点が付く場合も付かない場合もいずれの場合にも \pm 符号が付

く望ましくないシステムであることがわかる。

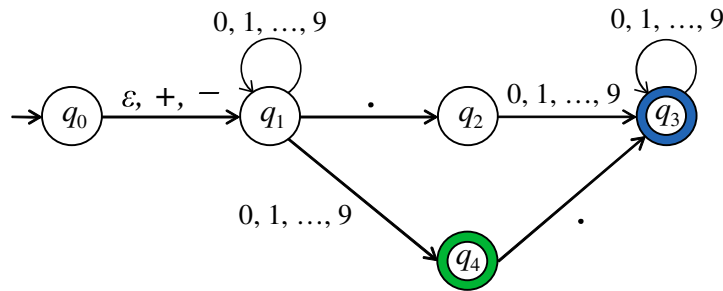


図 50: 非混色に拡張された 10 進数を受け取る ε -NCFA M_m .

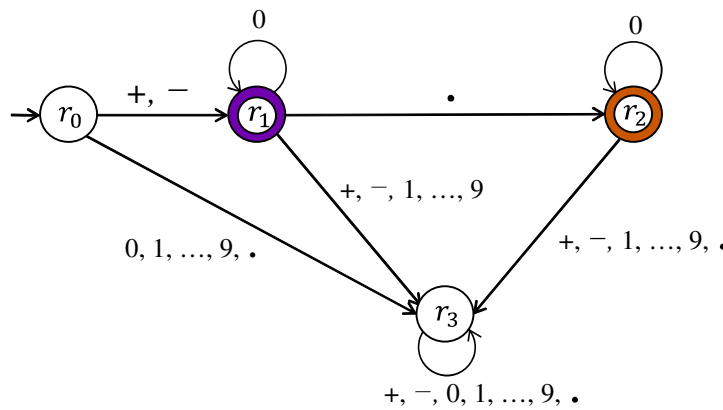


図 51: 値 0 の \pm 符号付き 10 進数を受け取る DCFA \overline{M}_s .

実際のモデル検査では検査対象となるシステムは受理の可否を判断する有限オートマトンではなく遷移系で記述される。また、その遷移系に望むあるいは避けてほしい特性は命題論理を拡張した時相論理式で表されるが、いずれにしても最終的には有限オートマトンの形式に変換されて実行される [39].

例 19 図 53 は Proc2 が一旦待機状態に入ったならば何れは臨界区域に入るという (時相論理式では $\square(W_2 \rightarrow \diamond C_2)$ と表される) 特性 P_2 ならびに Proc1 は無限回にわたり臨界区域に入るという (時相論理式では $\square \diamond C_1$ と表される) 特性 P_1 の充足性を調べるための NCBA $M = (Q, \Sigma, \delta, Q_0, F_1 + F_2)$ である。ここに、

- $Q = \{s_0, s_1, s_2\}$,

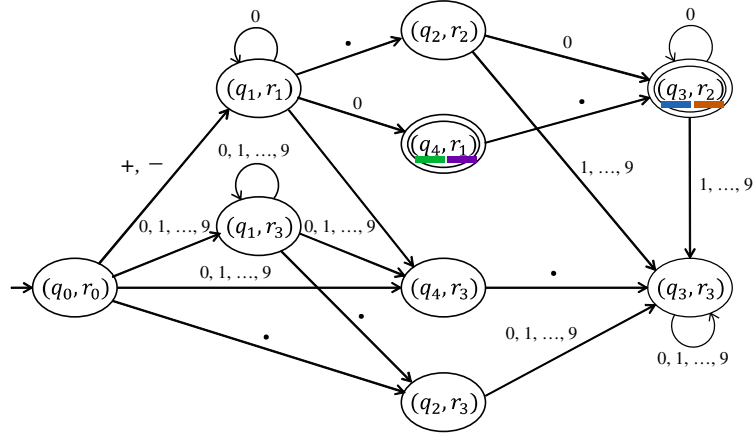


図 52: $L(M_m) \cap L(\overline{M_s})$ を受理する NCFA M_c .

- $\Sigma = 2^{AP} = 2^{\{n_1, w_1, c_1, n_2, w_2, c_2\}}$,

- 例えば,

$$\delta(s_0, W_2 \overline{C_2}) \ni s_2. \text{ ここに, } W_2 \overline{C_1} \triangleq \{X \subseteq AP \mid w_2 \in X, c_2 \notin X\} = \{\{n_1, w_2\}, \{w_1, w_2\}, \{c_1, w_2\}, \dots\},$$

$$\delta(s_0, \overline{C_1}) \ni s_1. \text{ ここに, } \overline{C_1} \triangleq \{X \subseteq AP \mid c_1 \notin X\} = \{\{n_1, n_2\}, \{w_1, n_2\}, \{n_1, w_2\}, \dots\},$$

$$\delta(s_0, TRUE) \ni s_0. \text{ ここに, } TRUE \triangleq \{X \subseteq AP\} = \{\{n_1, n_2\}, \{w_1, n_2\}, \{c_1, n_2\}, \dots\},$$

- $Q_0 = \{s_0\}$,

- $F_1 = \{s_1\}, F_2 = \{s_2\}$.

明らかに, $L_{\omega_2}(M) = TRUE^* W_2 \overline{C_2} \overline{C_2}^\omega = \overline{P_2}$ (すなわち, Proc2 が一旦待機状態に入ったものの永遠に臨界区域に入らない) かつ $L_{\omega_1}(M) = TRUE^* \overline{C_1} \overline{C_1}^\omega = \overline{P_1}$ (すなわち, Proc1 がある時点から永遠に臨界区域に入らない) が成り立つ.

以下に, 検査対象を表す遷移系と検査仕様を表す色付きビュッヒオートマトンの積演算によって得られる反例探索用の色付きビュッヒオートマトンの定義を与える.

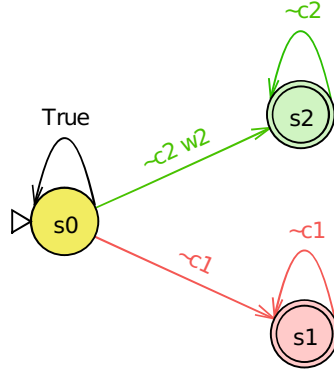


図 53: 特性 P_2 ならびに特性 P_1 の充足性を調べるための NCBA M .

定義 22 遷移系 $TS = (S, Act, \rightarrow, I, AP, L)$ と NCBA $M = (Q, 2^{AP}, \delta, Q_0, \Sigma_{i=1}^k F_i)$ に対して, TS と A の積オートマトン $TS \otimes A$ を以下のような NCBA $M' = (S \times Q \cup \{i\}, 2^{AP}, \delta', \{i\}, \Sigma_{i=1}^k F'_i)$ と定義する. ここに, i は新しい状態であり,

1. $\delta'(\langle s, q_f \rangle, L(t)) = \{\langle t, q_f \rangle \mid s \xrightarrow{\alpha} t \wedge \delta(p, L(t)) \ni q\}$. 各 $s_0 \in I$ に対して,
 $\delta'(i, L(S_0)) = \{\langle s_0, q \rangle \mid \exists q_0 \in Q_0 [\delta(q_0, L(s_0)) \ni q]\}$,
2. $F'_i = \{\langle s, q_f \rangle \mid s \in S, q_f \in F_i\}$

事実 6 TS を AP 上の遷移系, P を AP 上の特性とする.

$$L_\omega(TS \otimes A_{\overline{P}_i}) = \emptyset \iff TS \models P_i.$$

$L_\omega(TS \otimes M_{\overline{P}}) \ni \sigma$ が存在するとき, σ は $TS \models P$ の反例 (counter example) であるという.

例 20 図 54 に $TS_{\text{sm\&pr}} = (S, Act, \rightarrow, I, AP, L)$ と $M = (Q, 2^{AP}, \delta, \{s_0\}, F_1 + F_2)$ の積オートマトン NCBA $M' = TS_{\text{sm\&pr}} \otimes M = (S \times Q \cup \{i\}, 2^{AP}, \delta', \{i\}, F'_1 + F'_2)$ を示す. ここに,

- $S \times Q \cup \{i\} = (\{n_1, w_1, c_1\} \times \{n_2, w_2, c_2\} - \{(c_1, c_2)\}) \times \{s_0, s_1, s_2\} \cup \{i\}$
- δ' については例えば,

- $\delta'(\langle (n_1, n_2), s_0 \rangle, L(t_1, t_2)) = \{ \langle (t_1, t_2), q \rangle \mid (n_1, n_2) \xrightarrow{\alpha} (t_1, t_2) \wedge \delta(s_0, L(t_1, t_2)) \ni q \}$ より,

$$\delta'(\langle (n_1, n_2), s_0 \rangle, W_2\overline{C_2}) = \{ \langle (t_1, t_2), s_2 \rangle \mid (t_1, t_2) \xrightarrow{\alpha} (t_1, t_2) \wedge \delta(s_0, W_2\overline{C_2}) \ni s_2 \} = \{ \langle (n_1, w_2), s_2 \rangle \},$$

$$\delta'(\langle (n_1, n_2), s_0 \rangle, \overline{C_1}) = \{ \langle (t_1, t_2), s_1 \rangle \mid (n_1, n_2) \xrightarrow{\alpha} (t_1, t_2) \wedge \delta(s_0, \overline{C_1}) \ni s_1 \} = \{ \langle (n_1, w_2), s_1 \rangle, \langle (w_1, n_2), s_1 \rangle \},$$

$$\delta'(\langle (n_1, n_2), s_0 \rangle, TRUE) = \{ \langle (t_1, t_2), s_0 \rangle \mid (n_1, n_2) \xrightarrow{\alpha} (t_1, t_2) \wedge \delta(s_0, TRUE) \ni s_0 \} = \{ \langle (n_1, w_2), s_0 \rangle, \langle (w_1, n_2), s_0 \rangle \},$$
- $\delta'(i, L(n_1, n_2)) = \{ \langle (n_1, n_2), q \rangle \mid \delta(s_0, L(n_1, n_2)) \ni q \}$ より,

$$\delta'(i, N_1N_2) = \{ \langle (n_1, n_2), q \rangle \mid \delta(s_0, N_1N_2) \ni q \} = \{ \langle (n_1, n_2, s_0) \rangle, \langle (n_1, n_2, s_1) \rangle \}.$$
 ここに, $N_1N_2 \triangleq \{ X \subseteq AP \mid n_1 \in X, n_2 \in X \} = \{ \{n_1, n_2\}, \dots \}.$

- $F'_1 = S \times \{s_1\}, F'_2 = S \times \{s_2\}.$

明らかに, $L_\omega(M') = L_{\omega_1}(M') + L_{\omega_2}(M')$ が成り立ち,

$$L_{\omega_2}(M') \ni \{n_1, n_2\}(\{n_1, w_2\}\{w_1, w_2\}\{c_1, w_2\})^\omega$$

は $TS_{\text{sm\&pr}} \models P_2$ の反例であり,

$$L_{\omega_1}(M') \ni \{n_1, n_2\}(\{n_1, w_2\}\{n_1, c_2\}\{n_1, n_2\})^\omega$$

は $TS_{\text{sm\&pr}} \models P_1$ の反例である.

■

ここで, 色無しオートマトンによる従来型モデル検査法と色付きオートマトンを用いた新しいモデル検査法の性能比較を行う. 実行すべき検査項目 P_i が k 個存在するものと仮定する.

従来法では, 各検査項目 p_i ごとに検査対象システムを表す遷移系 TS と P_i の否定 \overline{P}_i を表すビュッヒオートマトン \overline{A}_i との積オートマトン $TS \times \overline{A}_i$ の探索を行うの

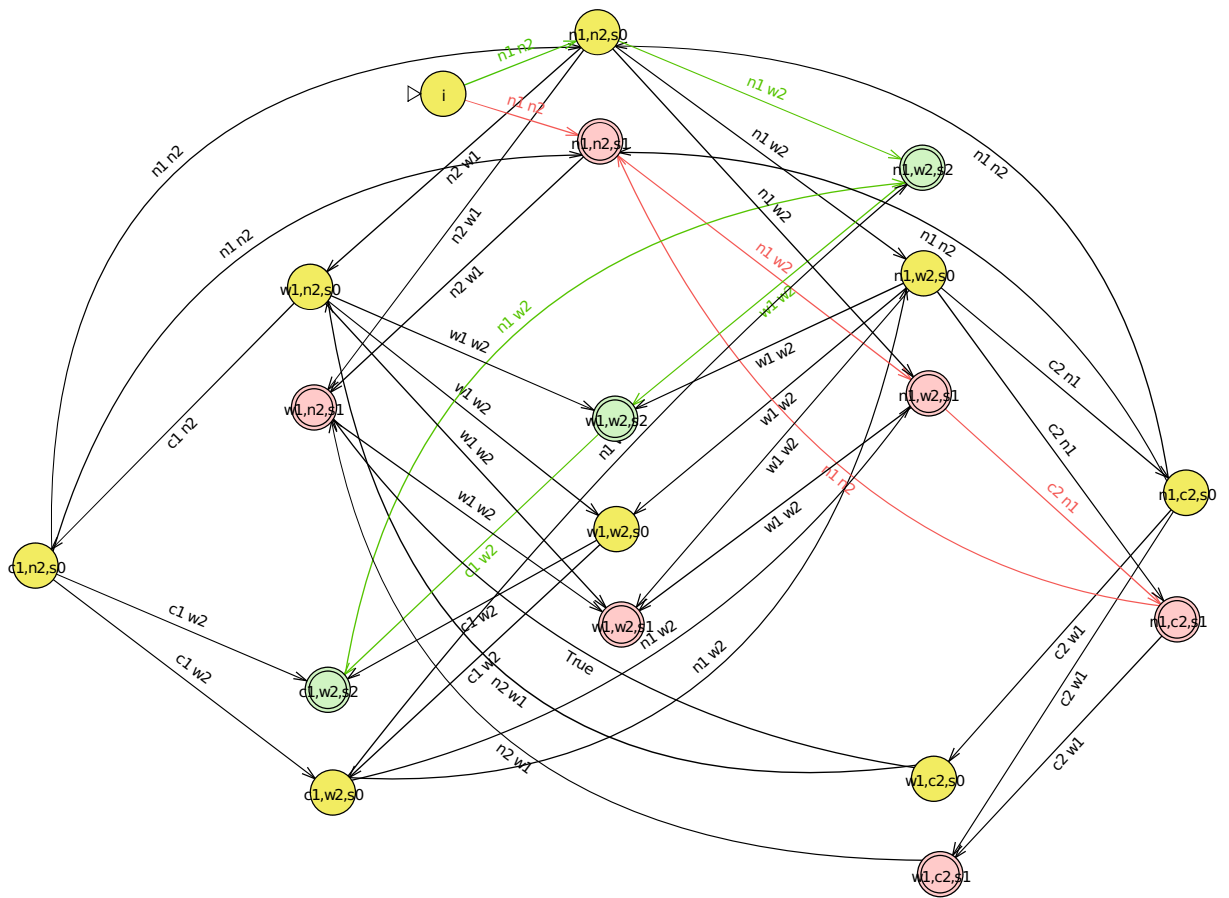


図 54: $TS_{sm\&pr}$ と M の積 NCBA M' .

で, $TS \times \overline{A_i}$ のサイズ和 $O(|TS| \times \sum |\overline{A_i}|)$ に比例する計算量が必要となる. 一方, 一つの色付きオートマトン A に全ての検査項目 $\{A_i\}$ が含まれるのであれば, 遷移系 ST とこのオートマトン A の積オートマトン $TS \times A$ のサイズ $O(|TS| \times |A|)$ に比例する計算量で済む. 従って, $|A| < \sum_{i=1}^k |A_i|$ が成り立つならば, 色付きオートマトンを使う方が有利である.

例えば, 上述の例 18 では, $|A|/(|A_1| + |A_2|) \cong 3/4$ より, 一つの色付きオートマトン A による検査で従来のものを約 $3/4$ に削減できている. 更に最初の例 15 のように, 個々の検査オートマトンが同じサイズ $|A_i| = |A|$ ならば, $|A| < \sum_{i=1}^k |A_i| = k|A|$ となり, 一つの色付きオートマトン A を用いた検査により, 従来型検査に必要な計算量を $1/k$ に削減できることになる.

3.8 まとめ

- 非決定性色付き有限オートマトン (NCFA) に関する諸定義と諸定理: 色付き言語 (ベクトル) および NCFA の定義を与えた. 更に, NCFA の例を示し, NCFA において重要な概念である非混色性を導入した. また, NCFA を決定性 CFA (DCFA) に変換する部分集合構成法を与え, 変換前後で混色状況や言語の一致に支障がないことを定理として示した. (これらの NCFA に関する諸定義と諸定理は [42] で発表した.)
- k 進 n 次元ドブリュイングラフ $DB_{k,n}$ と $D_{k,n}$ の等価性の証明: 末尾から n 番目の文字が $1, 2, \dots$, または $k-1$ のいずれかである文字列の集合 $L_{k,n}$ を受理する $N_{k,n}$ へ NCFA 版の部分集合構成法を適用し, 得られる DCFA $D_{k,n}$ が $DB_{k,n}$ と同型であることを証明した. この際, NCFA 版の部分集合構成法を分析した結果, $D_{k,n}$ の状態遷移が垂直に並べた $k-1$ 個の n 段シフトレジスタに対応していることを発見し, その結果を基にして書き直した $D_{k,n}$ の記

述と $DB_{k,n}$ の定義が一致することを示した。(この証明の詳細は [42] で発表した.)

- NCFA の非混色性に関する決定問題の計算複雑さの解明：NCFA の非混色性に関する三つの決定問題を考察し、それぞれの計算複雑さが下記のようになることを示した.
 - 非混色性検証問題 (UV 問題)： $NLOG$ 完全
 - 非混色分割問題 (UP 問題)：P
 - 非混色拡張問題 (UE 問題)：NP 完全

(これら三つの決定問題の計算複雑さの結果は [42] で発表した.)

- 同時到達可能グラフ $G_{sr}(M)$ の定義：NCFA ならびに ε -NCFA の積オートマトンから得られる無向グラフである同時到達可能グラフ $G_{sr}(M)$ の定義を行った。 $G_{sr}(M)$ の連結成分を確認することにより、NCFA, ε -NCFA の既存の受理状態に色付け可能かどうかを多項式時間で決めることができる。また、それらの既存の非受理状態に色付けできるかという問題が NP 完全であることも、 $G_{sr}(M)$ を用いて証明している。(NCFA, ε -NCFA の $G_{sr}(M)$ の定義は [42, 43] で発表した.)
- 色付き有限オートマトンの応用の可能性を示すため、以下の導入と例示を行った.
 - NCFA の正規表現エンジンへの応用のために、NCFA を機械的に色付き正規表現に変換するアルゴリズムを導入した。そして、NCFA の既存の正規表現エンジンへの取り入れのし易さを示し、実際に NCFA の色付き正規表現への変換を例示した。
 - NCFA のモデル検査への応用のために、無限言語上の色付き非決定性ビュッヒオートマトン (NCBA) を導入した。そして、相互排除問題を解

決する遷移系が二つの特性を満たすかどうかについて，NCBA によってモデル検査を行う例を示した．

(この NCFA を色付き正規表現に変換するアルゴリズムは [55] で発表した.)

4 色付きプッシュダウン・オートマトン

本章では、図 55 で示すようにチョムスキー階層において有限オートマトンの直上に位置するプッシュダウン・オートマトン (PDA) [16, 17, 19, 23, 36, 38, 46] に色付き受理状態の概念を適用 (図 56 参照) させた、色付きプッシュダウン・オートマトン (CPDA) の定義を行う。図 56 で示すように、色付きプッシュダウンオートマトンは色付き有限オートマトンの有限制御部にスタックを付け加えたもので、能力が向上する。さらには、色付き受理状態により、入力文字列を例えば受理 R、受理 G のように識別もできるようになる。

次に、NCEFA において議論を行った三つの決定問題を CPDA において検討する。後述するように、これらの全ての問題が決定不能となってしまうので、CPDA の能力を抑えて議論を行うために、CPDA に無曖昧という制約を与えた場合の計算複雑さについて検討する。

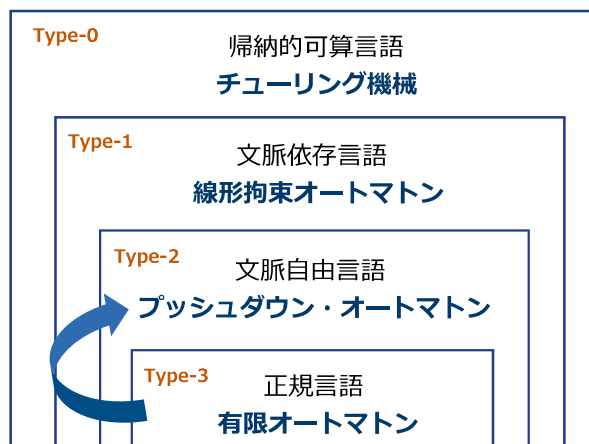


図 55: チョムスキー階層.

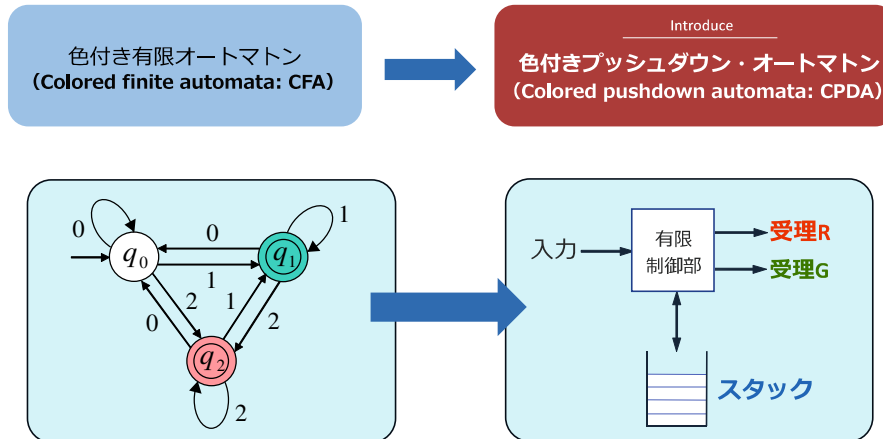


図 56: 色付きの概念をプッシュダウン・オートマトンへ拡張.

4.1 非決定性色付きプッシュダウン・オートマトン (NCPDA)

以下では、色付きプッシュダウン・オートマトン (CPDA) の定義を与える。

定義 23 次のように定まる 6 項組 $M = (Q, \Sigma, \Gamma, \delta, q_0, \sum_{i=1}^k F_i)$, を非決定性色付きプッシュダウン・オートマトン (nondeterministic colored pushdown automaton) といい, NCPDA と略記する. ここに, $\sum_{i=1}^k F_i \subseteq Q$ は色付き受理状態の集合であり, F_i は i 色の受理状態の集合を表す. 他は通常の NPDA と同じである.

また, 計算状況 (q, w, γ) , 計算状況間の遷移関係 \vdash ならびにその推移的閉包 \vdash^* が通常の NPDA と同じように定義される. 計算状況の三つの組は, それぞれ状態 q , 入力の残り w , スタックの内容 γ を意味する. 計算状況の対をつなぐ際には \vdash を用い, 0 ステップ以上の動作を表すには \vdash^* を用いる. $L_i(M) \triangleq \{x \in \Sigma^* \mid (q_0, x, \varepsilon) \vdash^* (q, \varepsilon, \varepsilon), q \in F_i\}$ を M によって i 色で受理される言語と呼ぶ. M が特定の入力文字列を受理するには, (1) 最終的に受理状態に入り, (2) 最終的にスタックが空であるという 2 つの条件が存在することに注意されたい. 条件 (1) のみでの受理の場合, いわゆる“受理状態受理”については, 第 4.3 節の最後で述

べる. $L(M) \triangleq \bigcup_{i=1}^k L_i(M)$ を M によって (統合) 受理される言語と呼ぶ. 特に, $L(M) = \sum_{i=1}^k L_i(M)$ が成り立つとき, $L(M)$ は非混色であると言い, M は $L(M)$ を純色受理すると言う.

同様に, CPDA の場合における同時到達可能グラフが以下のように定義される.

定義 24 NCPDA $M = (Q, \Sigma, \Gamma, \delta, q_0, \sum_{i=1}^k F_i)$ とする. M から得られる無向グラフ $G = (V, E)$

$$\begin{cases} V = Q \\ E = \{(p, q) \in Q \times Q \mid \exists x \in \Sigma^* [(q_0, x, \varepsilon) \vdash^* (p, \varepsilon, \varepsilon), \\ (q_0, x, \varepsilon) \vdash^* (q, \varepsilon, \varepsilon)]\} \end{cases}$$

は, M の同時到達可能グラフ (simultaneously reachable graph) と呼ばれ, $G_{sr}(M)$ と表される.

例 21 回文言語 $L_{\text{pal}} = \{ww^R \mid w \in \{a, b\}^*\}$ を受理するプッシュダウン・オートマトンについて考える. L_{pal} を受理する通常の色無しプッシュダウン・オートマトン NPDA とそのスタックの動きは図 57 のようになる. 次に, L_{pal} を生成する (無曖昧な) 文法 $S \rightarrow aSa \mid bSb \mid \varepsilon$ に対応する NPDA を多色化した NCPDA $M_p = (\{q_0, q_1, \dots, q_7\}, \{a, b\}, \{q_2, q_5\}, \delta, q_0, \{q_3\} + \{q_6\} + \{q_7\})$ の文法遷移図を図 58 に示す. 図において, “/ \textcircled{i} ” と “ \textcircled{i} /” は, スタック記号 q_i のプッシュとポップの動作をそれぞれ表す. ここでのスタックの動きは, 色無しプッシュダウン・オートマトンにおけるスタックの動きと, 変わらないことに注意されたい. しかしながら, スタックの動きは変わらないが, 以下に示すように色付き受理状態を導入することにより, 入力文字列を識別できる.

図 58 で示される受理状態 q_3, q_6, q_7 で受理される言語をそれぞれ $L_R(M_p), L_G(M_p),$

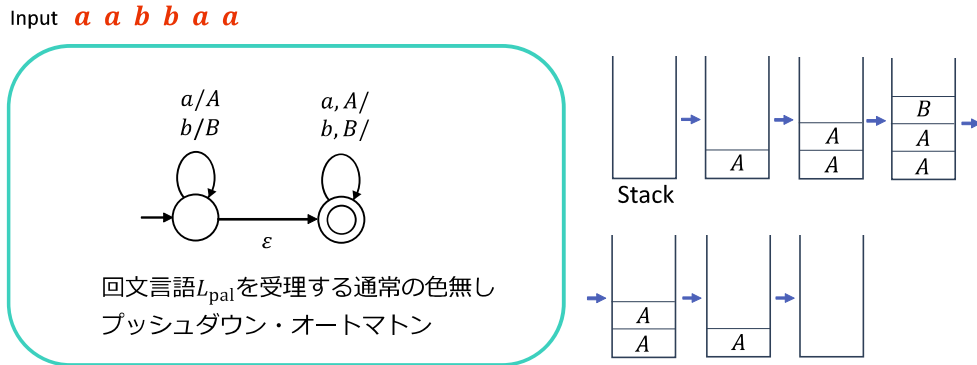


図 57: L_{pal} を受理する通常の色無しプッシュダウン・オートマトンとそのスタックの動き.

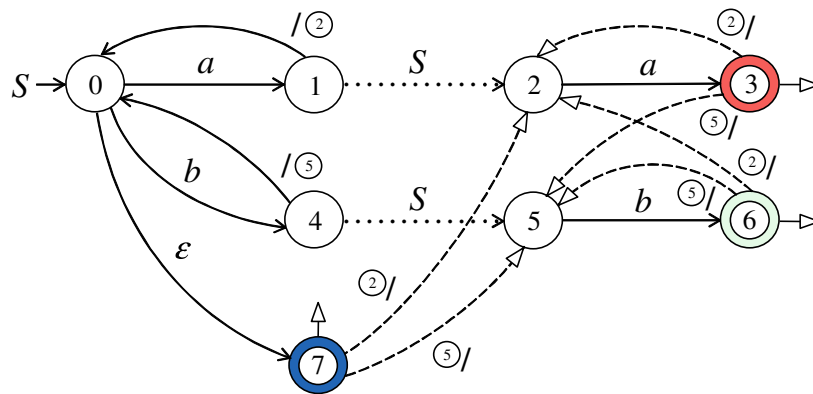


図 58: M_p の文法遷移図.

$L_B(M_p)$ とおくと、以下が成り立つ.

$$L_R(M_p) = aL_{\text{pal}}a,$$

$$L_G(M_p) = bL_{\text{pal}}b,$$

$$L_B(M_p) = \{\varepsilon\},$$

$$L(M_p) = \Sigma_{i=R,G,B} L_i(M_p) = L_{\text{pal}}.$$

よって、 M_p は L_{pal} を純色受理している.

さらに突き詰めて調べると、以前は受理状態でなかった状態 q_0, q_1, q_2, q_4, q_5 において受理される個々の言語は、それぞれ $L_0(M_p) = \{\varepsilon\}$, $L_1(M_p) = \{a\}$, $L_2(M_p) =$

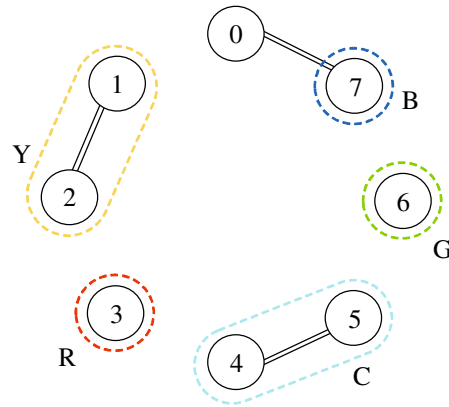


図 59: NCPDA M_p の同時到達可能グラフ $G_{sr}(M_p)$.

$aL_{\text{pal}}, L_4(M_p) = \{b\}$, ならびに $L_5(M_p) = bL_{\text{pal}}$ となることが分かる. これらの同定から, 図 59 に示す M_p の同時到達可能グラフ $G_{sr}(M_p)$ が得られる. もし M_p が図で Y 色に塗られた q_1 または q_2 で停止すれば, 入力文字列の最後に記号 a がないことが分かる. 同様に, 図で C 色になっている q_4 または q_5 で停止した場合は, b が欠落していることを示している.

4.2 NCPDA の非混色性に関する決定問題

CFA と同様に, NCPDA の UV 問題, UP 問題, ならびに UE 問題は次のように定義される. UV 問題の入力は NCPDA であり, UP 問題および UE 問題の入力は NPDA であることに注意されたい.

定義 25 NCPDA の非混色性検証問題 (Unmixedness Verification problem: UV 問題) は以下のように定義される.

$$\begin{cases} \text{入力: NCPDA } M = (Q, \Sigma, \Gamma, \delta, q_0, \sum_{i=1}^k F_i), \\ \text{出力: } \bigcup_{i=1}^k L_i(M) = \sum_{i=1}^k L_i(M)? \end{cases}$$

定義 26 NCPDA の非混色分割問題 (Unmixedness Partitioning problem: UP 問題) は以下のように定義される.

$$\left\{ \begin{array}{l} \text{入力: NPDA } M = (Q, \Sigma, \Gamma, \delta, q_0, F) \text{ と整数 } k \geq 2, \\ \text{出力: } F = \sum_{i=1}^k F_i \text{ であるような非混色の NCPDA } N = (Q, \Sigma, \Gamma, \delta, \\ q_0, \sum_{i=1}^k F_i) \text{ が存在するか?} \end{array} \right.$$

定義 27 NCPDA の非混色拡張問題 (Unmixedness Extension problem: UE 問題) は以下のように定義される.

$$\left\{ \begin{array}{l} \text{入力: An NPDA } M = (Q, \Sigma, \Gamma, \delta, q_0, F_0) \text{ と整数 } k \geq 1, \\ \text{出力: 非混色の NCPDA } N = (Q, \Sigma, \Gamma, \delta, q_0, \sum_{i=0}^k F_i) \\ \text{が存在するか?} \end{array} \right.$$

定理 8 NCPDA の (1) UV 問題, (2) UP 問題 ($k = 2$), (3) UE 問題 ($k = 1$) は全て決定不能である.

(証明) 決定不能であることが知られているチューリング機械の空集合問題を, それぞれの問題に還元する. 任意のチューリング機械 M の正しい受理計算状況系列の集合

$$L\langle M \rangle = \{w_1 \# w_2^R \# w_3 \# w_4^R \# \cdots \# w_m \text{ (or } w_m^R) \# \mid \\ w_1 \in p_0 \Sigma^*, w_m \in \Gamma^* \dot{F} \Gamma^*, w_i \vdash_M w_{i+1} (1 \leq i \leq m-1)\}$$

は 2 つの決定性 PDAM₁, M₂ で受理される言語

$$L(M_1) = \{w_1 \# w_2^R \# \cdots \# w_m \text{ (or } w_m^R) \# \mid \\ w_1 \in p_0 \Sigma^*, w_m \in \Gamma^* \dot{F} \Gamma^*, w_{2i-1} \vdash_M w_{2i} (1 \leq i \leq \lfloor m/2 \rfloor)\},$$

$$L(M_2) = \{w_1 \# w_2^R \# \cdots \# w_m \text{ (or } w_m^R) \# \mid \\ w_1 \in p_0 \Sigma^*, w_m \in \Gamma^* \dot{F} \Gamma^*, w_{2i} \vdash_M w_{2i+1} (1 \leq i \leq \lceil m/2 \rceil - 1)\}$$

の共通集合で表され, $L(M) = \{x \in \Sigma^* \mid p_0 x = w_1, w_1 \# w_2^R \# \cdots \# w_m \text{ (or } w_m^R) \# \in L\langle M \rangle\} = \emptyset \Leftrightarrow L\langle M \rangle = \emptyset \Leftrightarrow L(M_1) \cap L(M_2) = \emptyset$ が成り立つ [17]. ここに, $\Sigma, \Gamma, p_0, F, \#, w^R$ はそれぞれ M の入力アルファベット, テープアルファベット, 初

期状態, 受理状態の集合, Q にも Γ にも属さない計算過程を繋ぐ記号, w の逆順の列をそれぞれ表す. また, m が偶数 [奇数] の場合, $2\lfloor m/2 \rfloor = m$ [$m-1$] ならびに $2(\lfloor m/2 \rfloor - 1) + 1 = m-1$ [m] となることに注意されたい.

NPDA M' は次のように, 決定性 PDAM₁ と M_2 から構成される (図 60 参照):

- (i) M' の初期状態 q_i から M_1, M_2 それぞれの初期状態へ, 新しいスタック記号 Z_0 のプッシュ動作を伴う ε 遷移をさせる.
- (ii) M_1 の受理状態を非受理状態に変更し, それらから M' の新たな受理状態 q_{e1} へ Z_0 のポップ動作を伴う ε 遷移をさせる. 同様に, 下側の M_2 は, 別の受理状態 q_{e2} へ ε 遷移されるように修正される.

M_1 と M_2 内では, スタックに Z_0 が存在するため, M' は入力を受理したと見なすことができないことに注意されたい. 上記の構成から, それぞれの決定不能性を次のように証明する.

- (1) PDA M' を 2 色化した $F_R = \{q_{e1}\}, F_G = \{q_{e2}\}$ なる NCPDA N について, N が非混色である $\Leftrightarrow L_R(N) \cap L_G(N) = \emptyset \Leftrightarrow L(M) = \emptyset$.
- (2) M' の受理状態集合 $F = \{q_{e1}, q_{e2}\}$ を 2 分割して非混色の NCPDA N が得られる $\Leftrightarrow L_R(N) \cap L_G(N) = \emptyset \Leftrightarrow L(M) = \emptyset$.
- (3) $F_0 = \{q_i, q_{e2}\}$ 且つ $q_{e1} \in Q - F_0$ であるような NPDA M'' は, 別の色付き受理状態集合 $F_1 = \{q_{e1}\}$ を加えることによって 2 色の非混色な PDA N' に変えることができる $\Leftrightarrow L_0(N') \cap L_1(N') = (\{\varepsilon\} \cup L_G(N)) \cap L_R(N) = L_G(N) \cap L_R(N) = \emptyset \Leftrightarrow L(M) = \emptyset$.

□

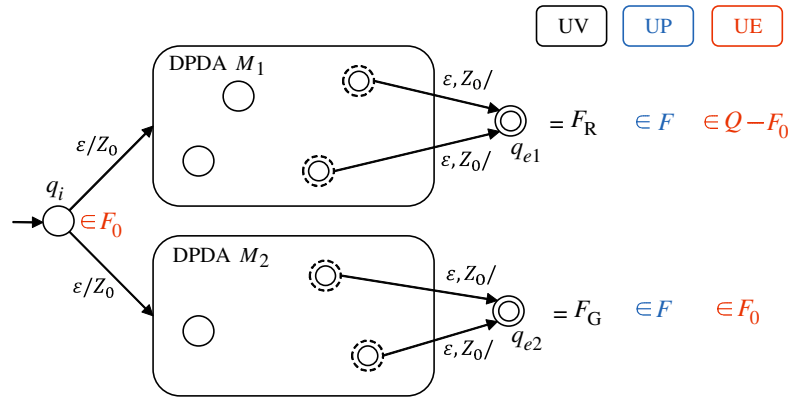


図 60: 定理 8 の証明の概念図.

4.3 無曖昧な NCPDA の非混色性に関する決定問題

図 61 に示すように、CPDA における上述の UV 問題、UP 問題、UE 問題の計算複雑さが、全て決定不能という、NCFA の場合と対照的な結果となった。この節では三つの決定問題の入力である NCPDA, NPDA にそれらが無曖昧であるという制約を与えた場合の計算複雑さを考察する。

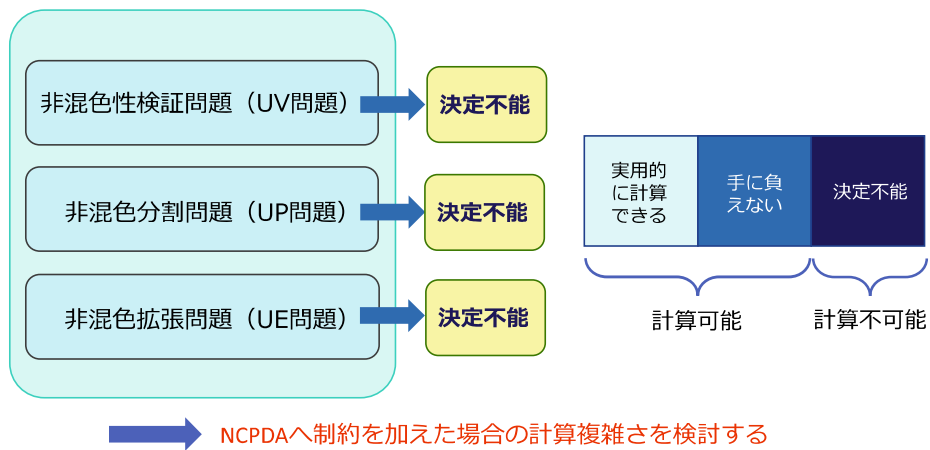


図 61: 三つの決定問題の決定不能性.

定義 28 以下が成り立つとき, NPDA は**無曖昧** (unambiguous) であると言う. $P_1 : (q_0, x, \varepsilon) \vdash^*(q_{f_1}, \varepsilon, \varepsilon)$, $P_2 : (q_0, y, \varepsilon) \vdash^*(q_{f_2}, \varepsilon, \varepsilon)$, $q_{f_1}, q_{f_2} \in F$ なる受理パス $P_1 \neq P_2$ が存在するならば, $x \neq y$.

NPDA における無曖昧の概念図を図 62 に示す.

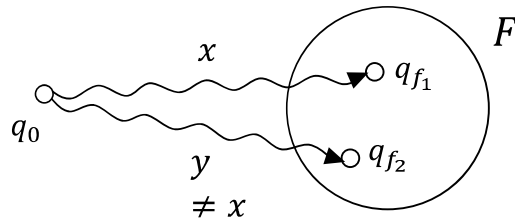


図 62: NPDA における無曖昧の概念図.

事実 7 [46] 文脈自由文法と NPDA の等価性を示すために用いられる変換法 [36, 38, 17, 16] は, それらの無曖昧性を保存する.

NCPDA の曖昧さについて, 次のように 2 つの異なる概念を提案する.

定義 29 (1) 以下が成り立つとき, NCPDA は**強い意味で無曖昧**であると言う: $P_1 : (q_0, x, \varepsilon) \vdash^*(q_{f_1}, \varepsilon, \varepsilon)$, $P_2 : (q_0, y, \varepsilon) \vdash^*(q_{f_2}, \varepsilon, \varepsilon)$, $q_{f_1}, q_{f_2} \in \bigcup_{i=0}^k F_i$ なる受理パス $P_1 \neq P_2$ が存在するならば, $x \neq y$.

(2) 以下が成り立つとき, NCPDA は**弱い意味で無曖昧**であると言う: $P_1 : (q_0, x, \varepsilon) \vdash^*(q_{f_1}, \varepsilon, \varepsilon)$, $P_2 : (q_0, y, \varepsilon) \vdash^*(q_{f_2}, \varepsilon, \varepsilon)$, $\exists i(1 \leq i \leq k)[q_{f_1}, q_{f_2} \in F_i]$ なる受理パス $P_1 \neq P_2$ が存在するならば, $x \neq y$.

NCPDA の強い意味での無曖昧性は, 受理状態間の色の違いを無視して NPDA と見なしたときの通常は無曖昧性に等しい. 一方, 弱い意味での無曖昧性の場合には, 通常 NPDA と見なしたとき, (色の異なる) 2 つ以上の受理パスが存在しても良い. NCPDA における強い意味での無曖昧, 弱い意味での無曖昧の概念図をそれぞれ図 63, 64 に示す.

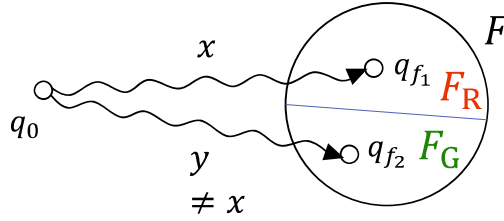


図 63: NCPDA における強い意味での無曖昧の概念図.

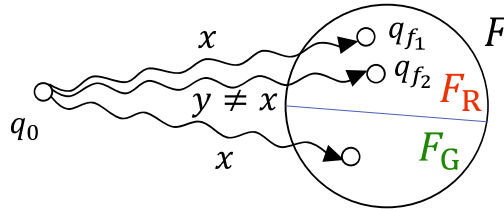


図 64: NCPDA における弱い意味での無曖昧の概念図.

定理 9 (1) 任意の無曖昧な NPDA M に関する UP 問題例 $(M, |F|)$ は常に真である.

(2) 無曖昧な NPDA に関する UE 問題は決定不能である.

(3) 無曖昧な NCPDA の UV 問題は, (3-1) 弱い意味では決定不能であり, (3-2) 強い意味では常に真である.

(証明) (1) 逆に, UP 問題例 $(M, |F|)$ が偽, すなわち $\Sigma_{i=1}^k F_i = F, F_i = \{q_{f_i}\}, F = \{q_{f_1}, \dots, q_{f_k}\}$ なる M から変換した NCPDA N が混色しているものとする. このことは, ある $i_1, i_2 (1 \leq i_1, i_2 \leq k), i_1 \neq i_2$ について, $L_{i_1}(N) \cap L_{i_2}(N) \neq \emptyset$, すなわち, ある入力 x について, M の x 上の 2 つの異なる受理パス $P_1 : (q_0, x, \varepsilon) \vdash^*(q_{f_1}, \varepsilon, \varepsilon)$ と $P_2 : (q_0, x, \varepsilon) \vdash^*(q_{f_2}, \varepsilon, \varepsilon)$ が存在することを意味し, M の無曖昧性に反する.

(2) UE 問題の決定不能性を示すために定理 8 の (3) の証明で用いた NPDA M'' は元々無曖昧である. なぜなら, M'' の非決定的な遷移は q_i からのみ発生するが, M_1 側への遷移は受理状態 q_{e2} へ決して到達しないからである.

(3-1) UV 問題の決定不能性を示すために定理 8 の (1) の証明で用いた NCPDA N について、各 $x \in L_R(N)$ に対する受理パス $(q_i, x, \varepsilon) \vdash^*(q_{e1}, \varepsilon, \varepsilon)$ がそれぞれ丁度 1 つ存在し、また、各 $x \in L_G(N)$ に対する受理パス $(q_i, y, \varepsilon) \vdash^*(q_{e2}, \varepsilon, \varepsilon)$ がそれぞれ丁度 1 つ存在する。従って、 N は弱い意味で無曖昧である。

(3-2) 逆に、ある強い意味で無曖昧な NCPDA N が混色しているものとする。このことは、ある色 $i_1, i_2 (1 \leq i_1, i_2 \leq k), i_1 \neq i_2$ について、 $L_{i_1}(N) \cap L_{i_2}(N) \neq \emptyset$ 、すなわち、ある入力 x について、色付きでない NPDA と見なした N の x 上の 2 つの異なる受理パスが存在することを意味する。□

第 4.1 節で予告したように、ここで受理状態受理の場合について検討する。この受理条件の変更によって影響を受ける証明は、UE の問題の証明 (定理 8 の (3) と定理 9 の (2)) である。今回は、PDA がスタックを空にしないで受理状態に入った場合でも入力文字列を受理したと結論付けなければならないので、定理 8 の証明にある NPDA M'' の構成をそのままにしておくと、好きなだけ多く (最大で M_1 と M_2 の内部状態の個数分) の色付き言語を生成できることになる。

このような状況を防ぐために、 M'' の構造を次のように変更する。

- (i) 初期状態 q_i からの Z_0 のプッシュ動作を伴う 2 つの ε 遷移を、(スタック動作なしの) 純粋な ε 遷移へ変更し、
- (ii) M_1 と M_2 の元の受理状態からの Z_0 のポップ動作を伴う遷移を、(スタック動作なしの) '\$' を読む遷移へ変更する。

ここで、新しく導入された記号 $\$ \notin \Sigma$ によって、状態 q_{e1} および q_{e2} で受理される言語は、他の状態で受理される言語と交わらなくなる。 M'' の元の受理状態の集合 F_0 の要素として、 q_{e2} と q_i に加えて、 M_1 と M_2 の全ての状態を選択する。このような設定により、 q_{e1} は、別の色で色付けされる拡張可能な受理状態集合 F_1 の唯一の候補となる。

このように構築された集合 F_0 は、定理 9 の (2) の証明には使用できない。受理状態としてそのような F_0 を持つ NPDA は曖昧である可能性があるためである。す

なわち、同じ入力文字列上で、初期状態 q_i からそれぞれ M_1 と M_2 の2つの受理状態へ向かう、2つの受理パスが存在する可能性がある。このような曖昧さを防ぐために、 M_2 の状態と同時到達可能な M_1 の状態を F_0 から除外する。これらの同時到達可能な状態は、2つの色の混色を引き起こすので、 F_1 の要素として選択することはできない。

M_2 の状態と同時到達不可能な M_1 の状態に対しては、それらを集合 F_0 の要素のままにしておかなければならない。 M_2 の状態と同時到達可能かどうかを判断できないような状態が M_1 に存在する場合でも、それらが q_{e1} と同じように動作することがわかっているならば、受理状態受理の条件下でも定理9の(2)が成り立つと結論付けられる。

4.4 まとめ

- 非決定性色付きプッシュダウン・オートマトン NCPDA に関する諸定義と例：色付き受理状態の概念を通常のプッシュダウン・オートマトンへ適用した NCPDA の定義を与えた。また、回文言語を受理する NCPDA の例により、NCPDA の有用性を示した。(NCPDA に関する諸定義と例は [43] で発表した。)
- NCPDA の非混色性に関する決定問題の計算複雑さの解明：NCPDA の非混色性に関する三つの決定問題を考察し、それらの計算複雑さがすべて決定不能になることを示した。(これら NCPDA の非混色性に関する三つの決定問題の計算複雑さの結果は [43] で発表した。)
- 無曖昧な NCPDA の非混色性に関する決定問題の計算複雑さの解明：非混色性判定問題への入力である NCPDA ならびに非決定性プッシュダウン・オートマトン (NPDA) を無曖昧なものに制約した場合の非混色性に関する三つ

の決定問題を考察した。UV 問題においては、弱い意味での無曖昧性と強い意味での無曖昧性の概念を定義し検討した。それらの結果が下記のようになることを示した。

- 無曖昧な NCPDA の非混色性検証問題 (UV 問題)
 - * 弱い意味で無曖昧な場合：決定不能
 - * 強い意味で無曖昧な場合：常に真
- 無曖昧な NCPDA の非混色分割問題 (UP 問題)：常に真
- 無曖昧な NCPDA の非混色拡張問題 (UE 問題)：決定不能

(これら無曖昧な NCPDA の非混色性に関する三つの決定問題の計算複雑さの結果は [43] で発表した。)

5 結論

5.1 研究の目的毎の総括

ここで本研究の目的を振り返る.

- **第1段階の目的**： k 進 n 次元ドブリュイングラフ $DB_{k,n}$ と、正規言語の k 個組み $(0 + 1 + \dots + k - 1)^*1(0 + 1 + \dots + k - 1)^{n-1}, \dots, (0 + 1 + \dots + k - 1)^*(k - 1)(0 + 1 + \dots + k - 1)^{n-1}$ を受理する最小状態数の決定性色付き有限オートマトン (DCFA) $D_{k,n}$ の状態遷移図の同型性を証明することにより、ドブリュイングラフに対して正規言語によってある種の特徴づけを行うこと.
- **第2段階の目的**：2進 n 次元ドブリュイングラフと $D_{2,n}$ の同型性を k 進に拡張するための手段として得られた、受理状態を多色化することにより入力文字列を多種類に類別可能にした“色付きオートマトン”を考察すること.

まず第1段階の目的に関して述べる. 2進 n 次元ドブリュイングラフ $DB_{2,n}$ と $D_{2,n}$ の同型性の証明結果を k 進に拡張するための手段として色付き有限オートマトンを導入した. この色付き有限オートマトンを用いて, k 進 n 次元ドブリュイングラフ $DB_{k,n}$ と正規言語の k 個組み $(0 + 1 + \dots + k - 1)^*1(0 + 1 + \dots + k - 1)^{n-1}, \dots, (0 + 1 + \dots + k - 1)^*(k - 1)(0 + 1 + \dots + k - 1)^{n-1}$ を受理する最小状態数の CFA の状態遷移図が同型であることを証明することにより, ドブリュイングラフにある種の特徴づけを行い, 第1段階の目的を達成した.

第2段階の目的は, 色付きオートマトンを考察することであり, 本研究では NCFA の非混色性を実用上の課題と捉え, 非混色性に関する三つの決定問題を提起しそれらの計算複雑さについて議論を行った. また, 色付き受理状態の概念をプッシュ

ダウン・オートマトンに拡張させ同様の決定問題について検討を行った。これらにより、十分に色付きオートマトンを考察し第2段階の目的を達成したとは言えないが、非混色性に関しては、様々な角度から議論を行ったため、十分な考察ができたと考える。

5.2 まとめ

本研究では、2進 n 次元ドブリユイングラフ $DB_{2,n}$ と、正規言語 $(0+1)^*1(0+1)^{n-1}$ を受理する最小状態数決定性有限オートマトン $D_{2,n}$ の状態遷移図の構造的等価性の証明結果を k 進に拡張するために、受理状態が2色以上の色で色付けされ、有限オートマトンに入力文字列の識別能力を与えた色付き有限オートマトン (CFA) を導入し、 $DB_{k,n}$ と、正規言語の k 個組み $(0+1+\dots+k-1)^*1(0+1+\dots+k-1)^{n-1}, \dots, (0+1+\dots+k-1)^*(k-1)(0+1+\dots+k-1)^{n-1}$ を受理する最小状態数の CFA の状態遷移図が同型であることを証明した。現在では、オートマトン理論の研究は成熟しつくしたと見なす専門家もいるような状況の中で、この結果は自明な同型性と論証なしに判断していた古典的な疑問に、明確な回答を与えている。また、構造一致の本質的理由の解明は、グラフ理論とオートマトン理論の関係性の新しい知見に繋がる結果と言える。

次に、非決定性 CFA (NCFA) の非混色性に関する3つの決定問題である非混色性検証問題 (UV 問題)、非混色分割問題 (UP 問題)、ならびに非混色拡張問題 (UE 問題) の計算複雑さについて調査した。その結果を表1に、非決定性色付きプッシュダウン・オートマトン (NCPDA) における上記の3つの問題の結果と共に示す。ここで、“U”と“T”は、それぞれ決定不可能であることならびに自明に決定可能 (恒久的な真) であることを表す。

NCFA の UV 問題と UP 問題が P に属することは、NCFA を決定性に変換せずとも多項式時間内に非混色性を確認できる確証を得たことであり、實際上重要な

結果である。なお、UP問題がPに属することを示す際には、NCFAの積オートマトンから得られる無向グラフである同時到達可能グラフという一般性のある概念を導入した。また、UE問題がNP完全であることは、独立集合問題をUE問題に還元することにより証明を行った。UE問題のNP完全性に関しては、その実行時間が実用的な範囲に納まるアルゴリズムの設計が困難であるため、受理状態の色の増大数を抑える等の妥協策が必要であると言える。

プッシュダウン・オートマトンに関して、一切制約のないNCPDAに関する上記の問題が決定不能であることは、そのようなオートマトンが混色かどうかを確認し、あるいは非混色なオートマトンを作成することが一般的に不可能であることを意味する。一方、NCPDAに制約を与えた場合、ここでは無曖昧という制約を与えた場合、表1の3行目に見られるように様々な状況が発生する。実際、例13のNCPDA M_p が非混色であることは偶然ではなく、その無曖昧さの直接的な結果である。UV問題においては、弱い意味で無曖昧性ならびに強い意味で無曖昧性というNCPDAに関する独自の曖昧性を定義し、強い意味で無曖昧な場合のUV問題と無曖昧な場合のUP問題は常に真となり、それら以外の弱い意味で無曖昧の場合のUV問題、無曖昧な場合のUE問題は決定不能となることを示した。

表 1: NCFA と NCPDA の非混色性判定問題.

入力	UV	UP	UE
NCFA	$NLOG$ 完全	P	NP 完全
NCPDA(一般)	U	U	U
NCPDA(無曖昧)	弱: U 強: T	T	U

5.3 今後の課題

最後に、色付きオートマトンに関する研究の今後の可能性について述べる。まず最初に、本論文で扱うべき範囲内であったものの、時間不足により取り上げることが出来なかった課題を幾つか挙げる。

- 決定性色付き有限オートマトンの状態数を最小化するアルゴリズム：本論文の第3章において非決定性から変換して得られた決定性色付き有限オートマトン $D_{k,n}$ の状態数はそれ以上減らすことができなかった。しかしながら、一般的には元のオートマトンと受理能力が等価でありながら状態数がより少ないオートマトンが存在する。このようなときに用いるべき Hopcroft アルゴリズム [16] のような状態数最小化アルゴリズムとその計算量を明らかにすること。
- “色付き” Myhill-Nerode の定理：色なしの場合、ある同値類の有限性によって正規言語を特徴づけるとともに、与えられた言語の非正規性を示すためにも用いられる同定理が、色付き有限オートマトンで受理される言語族に対してどのように拡張されるのか？
- “色付き” 言語（ベクトル）の閉包性：色無し言語に対するブール演算（和集合、共通集合、補集合）や接続演算などの適用結果は当然色無し言語になるが、同様な演算を色付き言語ベクトルに対してどのように定義すべきか？
- “色付き” 正規表現の色無し正規表現への応用：第3章で色付き有限オートマトンに対応する色付き正規表現なる表現形式を与えたが、この表現形式は色無しの場合にも適用可能である。しかも従来の正規表現よりもコンパクトでかつ直観的な言語表現になっている。

- “色付き”正規文法：通常の色なし有限オートマトンの場合にはそれらと等価な表現能力を持つ，正規文法が存在した．色付き有限オートマトンの場合に同様な等価性をもたらすような色付きの正規文法はどのように定義されるべきか？
- “色付き”単項2階論理式：色なしの場合，各正規言語は比較演算子 $<$ を含む単項2階論理式で特徴づけられることが知られているが，色付き有限オートマトンで受理される色付き言語に対応する（であろう）論理式はどのように色付けされるのか？
- 無曖昧な“色付き”有限オートマトンならびに自己検証型“色付き”有限オートマトン：一見すると無曖昧性と非混色性は類似した概念であるが，プッシュダウン・オートマトンに関する結果で見られたように，これらはある意味互いに直交した概念と言える．そこで，既存の無曖昧な有限オートマトンを色付きのものに拡張することが考えられる．また，関連研究として述べたように，自己検証有限オートマトンは2色有限オートマトンの一種と考えることができるので，それを3色以上のものに拡張することが考えられる．

次に，本研究の進展に伴い，新たに浮かび上がってきた課題について述べる．

本論文で得られた色付きオートマトンに関する諸結果には肯定的なものや否定的なものも存在するが，それらを総合的に判断すると，今後の探求すべき対象として，次のような条件を満たすものが望ましいことがわかる．

- (i) 受理状態集合を有するオートマトンであること．
- (ii) 決定性の受理能力が（状態数は増加するものの）非決定性（あるいは決定性の一般化）と変わらないこと．
- (iii) それらの受理能力がプッシュダウン・オートマトンを超えないこと．

上記の条件を満たす既存のオートマトンを幾つか挙げる．

- 2方向 (two-way) 有限オートマトン [16, 17, 18] : 決定性有限オートマトンは2方向非決定性のものと等価である.
- 交代性 (alternating) オートマトン [18, 59] : 決定性有限オートマトンは交代性のものと等価である.
- 入力駆動 (input-driven) プッシュダウン・オートマトン [47, 48] : 決定性入力駆動プッシュダウン・オートマトンは非決定性のものと等価である.
- 木 (tree) オートマトン [18, 57] : 決定性上昇型木オートマトンは非決定性 (上昇または下降型) のものと等価である.
- ω 言語上のオートマトン [18, 58] : 決定性 Muller オートマトンは第 3.6 節の非決定性ビュッヒオートマトンと等価である.

以上の他に, 上述の条件 (i) は満たすものの他の条件については不明であるようなオートマトンとして, (生成系でない) 確率 (stochastic) 有限オートマトン, 量子 (quantum) 有限オートマトン, 記号的 (symbolic) 有限オートマトンなどを挙げる事ができる. 以上のような色付きオートマトンの適用および応用領域を図 65 に示す.

このように, 本論文で提案し分析した色付きオートマトンとその非混色性は, これまで研究され応用されてきた様々なオートマトンの変種に対して幅広く適用可能な概念である.

本論文では, オートマトンの新しい二つの計算モデルを提案し, それらの非混色性の計算複雑さから得られる新しい知見を明らかにすることにより, オートマトン理論, 計算複雑さの理論ならびにそれらの応用分野の発展に貢献する結果を示した.

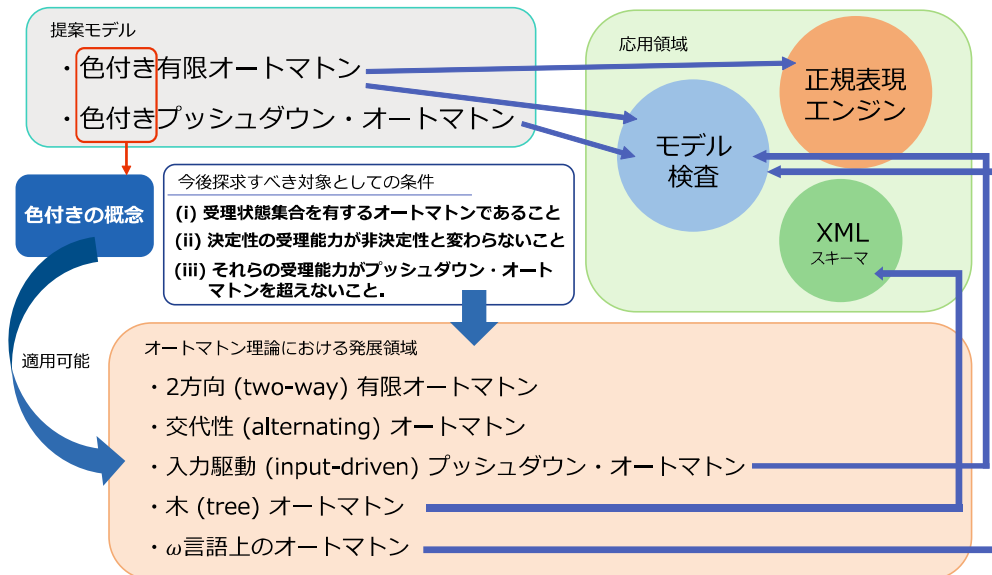


図 65: 色付きオートマトンの適用および応用領域.

6 参考文献

- [1] A. M. Turing, On Computable Numbers, with an Application to the Entscheidungsproblem, Proceedings of the London Mathematical Society. 2 42, pp. 230–265, 1936.
- [2] Turing’s proof, https://en.wikipedia.org/wiki/Turing%27s_proof (retrieved July, 2022).
- [3] G. Hoffmann de Visme, Binary Sequences, Hodder & Stoughton, 1971.
- [4] H. Miyakawa, Y. Iwatare, and H. Imai, Coding Theory, Shoukoudou Co., 1973 (in Japanese).
- [5] D. Du, F. Cao, and F. Hsu, De Bruijn digraphs, kautz digraphs, and their generalizations, in Combinatorial Network Theory, pp. 65-96, Kluwer Academic, the Netherlands, 1996.
- [6] T. Gagie, G. Manzini, and J. Siren, Wheeler graphs: A framework for BWT-based data structure, Volume 698, pp. 67-78, Theoretical Computer Science, 2017.
- [7] 当麻喜弘, 順序回路論, 昭晃堂, 1976.
- [8] De Bruijn graph, https://en.wikipedia.org/wiki/De-Bruijn_graphs (retrieved July, 2022).
- [9] C. S. Burriss, F. C. Motta, and P. D. Shipman, An Unoriented Variation on de Bruijn Sequences. Graphs and Combinatorics, 33, pp. 845-858, 2017.
- [10] A. Gibbons, Algorithmic Graph Theory, Cambridge University Press, 1985.

-
- [11] M. N. S. Swamy and K. Thulasiraman, *Graphs, Networks, and Algorithms*, John Wiley & Sons, 1981.
- [12] Z. Xu, Multi-Shift de Bruijn Sequence, *Computer Science - Discrete Mathematics*, 68R15, G.2.1, F.2.2, 2010.
- [13] P. B. Dragon, O. I. Hernandez, J. Sawada, and A. Williams, Constructing de Bruijn Sequences with Co-Lexicographic Order: The k -ary Grandmama Sequence, *European Journal of Combinatorics Volume 72*, pp. 1-11, 2018.
- [14] E. Moreno, De Bruijn sequences and De Bruijn graphs for a general language, *Information Processing Letters* 96, pp. 214-219, 2005.
- [15] P. Compeau, P. Pevzner, and G. Tesler, How to apply de Bruijn graphs to genome assembly, *Nature Biotechnology*, Volume 29 Number 11, 2011.
- [16] J. E. Hopcroft, R. Motowani, and J. D. Ullman, *Introduction to Automaton Theory, Languages, and Computation*, Addison-Wesley Longman, 2001.
- [17] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley Publishing, 1979.
- [18] E. Moriya, *Formal Language and Automaton*, Saiensu-sha Co., Ltd. Publishers, 2001 (in Japanese).
- [19] J. V. Leeuwen, *Handbook of Theoretical Computer Science Volume B*, The MIT Press/Elsevier, 1990.
- [20] G. Castiglione, A. Restivo, and M. Sciortino, Nondeterministic Moore automata and Brzozowski's minimization algorithm, *Theoretical Computer Science*, 450, pp. 81-91, 2012.

-
- [21] J. R. Buchi, *Finite Automata, Their Algebras and Grammars*, Springer-Verlag New York, 1989.
- [22] M. R. Garey and D. S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, 1979.
- [23] G. Rozenberg and A. Salomaa, *Handbook of Formal Languages Volume 1 Word, Language, Grammar*, Springer-Verlag, Berlin Heidelberg, 1997.
- [24] L. Kallmeyer, *Parsing Beyond Context-Free Grammars*, Springer-Verlag Berlin Heidelberg, 2010.
- [25] W. Kuich and H. Maurer, The Structure Generating Function and Entropy of Tuple Languages, *Information and Control* 19, pp. 195-203, 1971.
- [26] P. Linz, *An Introduction to Formal Languages and Automata*, Sixth Edition, Jones & Bartlett Learning, 2017.
- [27] G. Jirásková and G. Pighizzini, Optimal simulation of self-verifying automata by deterministic automata, *Information and Computation*, 209, pp. 528-535, 2011.
- [28] O. Kupferman and Y. Lustig, Lattice Automata, Verification, Model Checking, and Abstract Interpretation, *Proceedings of VMCAI 2007*, pp. 199-213, 2007.
- [29] R. L. Graham, M. Grötschel, and L. Lovász, *Handbook of Combinatorics Volume 1*, The MIT Press Cambridge, Massachusetts, 1995.
- [30] S. Eilenberg, *Automata, Languages, and Machines*, Academic Press, 1976.

-
- [31] M. Holzer and M. Kutrib, Descriptive and computational complexity of finite automata — A survey, *Information and Computation*, 209, pp. 456-470, 2011.
- [32] I. H. Sudborough, On Tape-Bounded Complexity Classes and Multihead Finite Automata, *Journal of Computer and System Sciences*, 10, pp. 62-76, 1975.
- [33] L. W. Beineke and R. J. Wilson, *Selected Topics in Graph Theory 2*, Academic Press, 1983.
- [34] F. Harary, *Graph Theory*, Addison-Wesley Publishing Company, 1972.
- [35] A. L. Rosenberg, *The Pillars of Computation Theory*, Springer Science + Business Media, 2010.
- [36] M. Sipser, *Introduction to the Theory of Computation*, Kyoritsu-shuppan, 2008 (in Japanese).
- [37] C. Herzog, Pushdown Automata with Bounded Nondeterminism and Bounded Ambiguity, *Theoretical Computer Science* 181, pp. 141-157, 1997.
- [38] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 3rd Edition, Addison-Wesley Longman, 2007.
- [39] C. Baier and J. Katoen, *Principles of model checking*, The MIT Press, 2008.
- [40] Nim Programming Language, <https://nim-lang.org/> (retrieved July, 2022).
- [41] R. Cox, Regular Expression Matching in the Wild, <https://swtch.com/rsc/regexp/regexp3.html> (retrieved July, 2022).
-

-
- [42] Y. Takahashi and A. Ito, Finite Automata with Colored Accepting States and Their Unmixedness Problems, Special section of IEICE Transactions on Information and Systems, Vol. E 105-D, No. 3, pp. 491–502, 2022.
- [43] Y. Takahashi and A. Ito, On the Unmixedness Problems of Colored Pushdown Automata, Proceedings of NICS'21 8th NAFOSTED Conference on Information and Computer Science, pp. 55–59, 2021.
- [44] 高橋芳明, 伊藤暁, ドブリュイングラフと状態数最小化有限オートマトンの等価性について, IEICE Technical Report, Vol. 117, No. 28, pp. 107–114, 2017.
- [45] E. Indzhev and S. Kiefer, On Complementing Unambiguous Automata and Graphs With Many Cliques and Cocliques, [arXiv:2105.07470 \[cs.FL\]](https://arxiv.org/abs/2105.07470), <https://arxiv.org/abs/2105.07470>, May 2021.
- [46] C. Herzog, Pushdown Automata with Bounded Nondeterminism and Bounded Ambiguity, Theoretical Computer Science, Volume 181, pp.141–157, 1997.
- [47] R. Alur and P. Madhusudan, Visibly pushdown languages, Proceedings of the thirty-sixth annual ACM Symposium on Theory of Computing, pp. 202–211, 2004.
- [48] A. Okhotin and K. Salomaa, Complexity of Input-Driven Pushdown Automata, ACM SIGACT News, Vol. 45, No. 2, 2014.
- [49] P. Compeau, P. Pevzner and G. Tesler, How to apply de Bruijn graphs to genome assembly, Nature Biotechnology, Volume 29 number 11, pp. 987-991, 2011.
- [50] Koorde, <https://en.wikipedia.org/w/index.php?title=Koorde&oldid=1095548974> (retrieved Aug, 2022).
-

-
- [51] J. A. Brzozowski, Derivatives of regular expressions, J. ACM 11(4), pp. 481-494, 1964.
- [52] 新屋他, 正規表現技術入門, 技術評論社, 2015.
- [53] A. V. エイホ他, コンパイラ-原理・技法・ツール I, サイエンス社, 1990.
- [54] D. Picheta, Nim in Action, Manning Publications, 2017.
- [55] 高橋芳明, 伊藤暁, 色付き有限オートマトンから色付き正規表現への変換アルゴリズム, 2022年度(第73回)電気・情報関連学会中国支部連合大会, 2022.
- [56] M. Holzer and S. Jakobi, A Note on the Computational Complexity of Some Problems for Self-Verifying Finite Automata, IFIG Research Report 1702, April 2017.
- [57] C. Ferdinand, H. Seidl, and R. Wilhelm, Tree Automata for Code Selection, Acta Informatica 31, pp. 741-760. 1994.
- [58] J. Esparza, Automata Theory An Alogorithmic Approach Lecture Notes, <https://www7.in.tum.de/~esparza/automatanotes.html> (retrieved July, 2022).
- [59] A. Chandra, D. Kozen, and L. Stockmeyer, Alternation, Journal of the ACM 28 (1), pp. 114-133, 1981.
- [60] Y. Chen, J. Hu, X. Ling, and T. Huang, A novel 3D NoC architecture based on De Bruijn graph, Computers & Electrical Engineering, Volume 38, Issue 3, pp. 801-810, 2012.

A 付録

A.1 倍化法によるドブリュイングラフの構成

de Bruijn によって提案された $DB_{2,n}$ の増加的構成法を倍化法と言う。この方法を任意のグラフに対して一般化したものは line graph construction と呼ばれている [8].

Algorithm 1 倍化法 [3]

- 1: 各辺 (u, v) の中央に点 w を追加し, 2つの辺 $(u, w), (w, v)$ に分割する.
 - 2: 追加された新点 u, v 同士の間, 旧点 w を通過する長さ 2 の有向道 $(u, w), (w, v)$ が存在するならば, (u, v) なる有向辺を追加する.
 - 3: すべての旧点と, それらに不随する辺を削除する.
-

例 22 ドブリュイングラフ $DB_{2,1}$ から倍化法を用いて, $DB_{2,2}$ を求める.

まず, ドブリュイングラフ $DB_{2,1}$ の各辺を新しい状態とする (図 66). 次に状

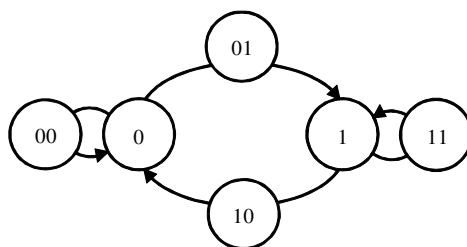


図 66: 倍化法 (例 22 のステップ 1).

態 10 に着目する．図 67 に示すように，状態 10 からは 0 という 1 つの状態を経て，状態 00 に遷移する長さ 2 の有向道が存在する．よって，状態 10 から状態 00 に遷移する辺を構成する．また，状態 10 は図 68 に示すように，状態 01 にも条件を満

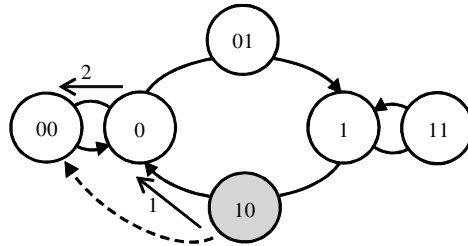


図 67: 倍化法 (例 22 のステップ 2) .

たす長さ 2 の有向道が存在するので，状態 10 から状態 01 に遷移する辺を構成する．この作業を，図 66 で構成した全ての新しい状態について行う．そして，旧点

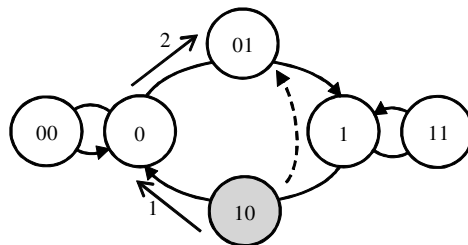


図 68: 倍化法 (例 22 のステップ 2 の続き) .

0, 1 とそれらに不随する辺を削除すると，図 69 を構成することができる． $DB_{2,2}$ から倍化法により，段階的に求めた $DB_{2,3}, DB_{2,4}$ を図 70, 71 に示す．ここで，DFA D_1, D_2, D_3, D_4 とドブリュイングラフ $DB_{2,1}, DB_{2,2}, DB_{2,3}, DB_{2,4}$ を比較してみると，ラベルを無視した構造同士は互いに同一であることがわかる．

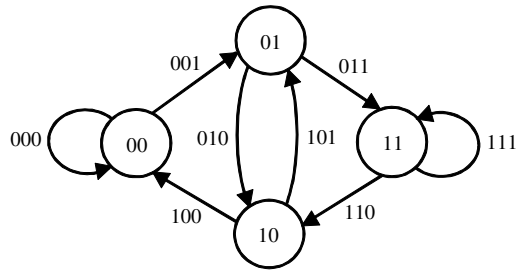


図 69: ドブリュイングラフ $DB_{2,2}$.

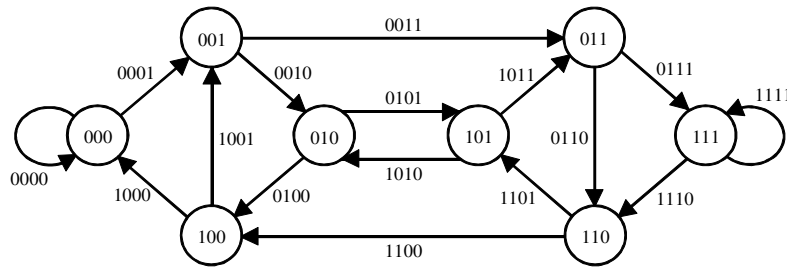


図 70: ドブリュイングラフ $DB_{2,3}$.

A.2 倍化法の状態遷移表による解釈

ドブリュイングラフのもう一つの構成法である倍化法において、ドブリュイングラフは、 $DB_{2,n-1}$ のグラフから $DB_{2,n}$ を構成している。この考え方に基づいて、 D_{n-1} から D_n を構成する方法を検討する。この構成法を増加的部分集合構成法と呼ぶ。

D_{n-1} を遷移表で表すと、図 72 のようになる。ここで \rightarrow がついた状態は初期状態、 $*$ がついた受理状態を意味する。以下に増加的部分集合構成法のアルゴリズムを示す。

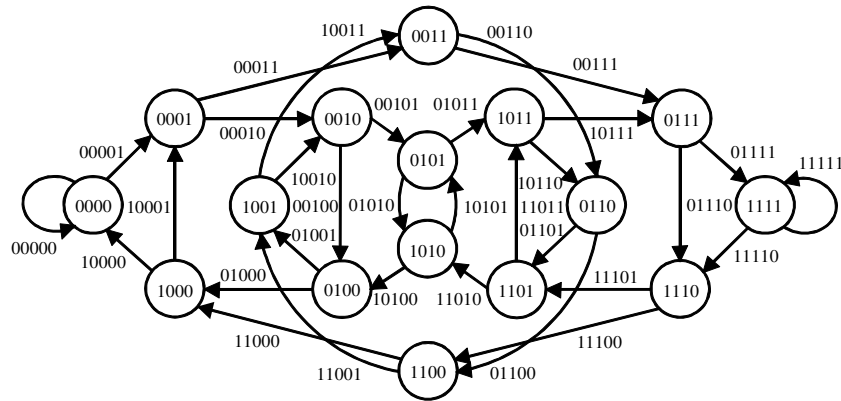


図 71: ドブリュイングラフ $DB_{2,4}$.

Algorithm 2 増加的部分集合構成法

- 1: D_{n-1} の遷移表を基にして、①に A の部分と同じ数だけ状態を増やし、そのすべてを受理状態にする。その際の状態番号は、上から順番に A の最下部の状態番号より 1 ずつ番号を増やしていき、B を構成する。
 - 2: A の元々受理状態だった状態の受理マークを取り消す。
 - 3: ②の q_0 の状態から下方に移動しながら、ラスタ順に③から、②の状態をコピーしていく。
 - 4: C の部分の状態を全て消し、ステップ 3 で③にコピーした状態全てを、C の部分にコピーする。
-

例 23 Algorithm2 の増加的部分集合構成法を用いて、DFA D_1 から D_2 を求める。変換の様子を図 73 に示す。

A.3 状態数最小化 DCFA の一意性

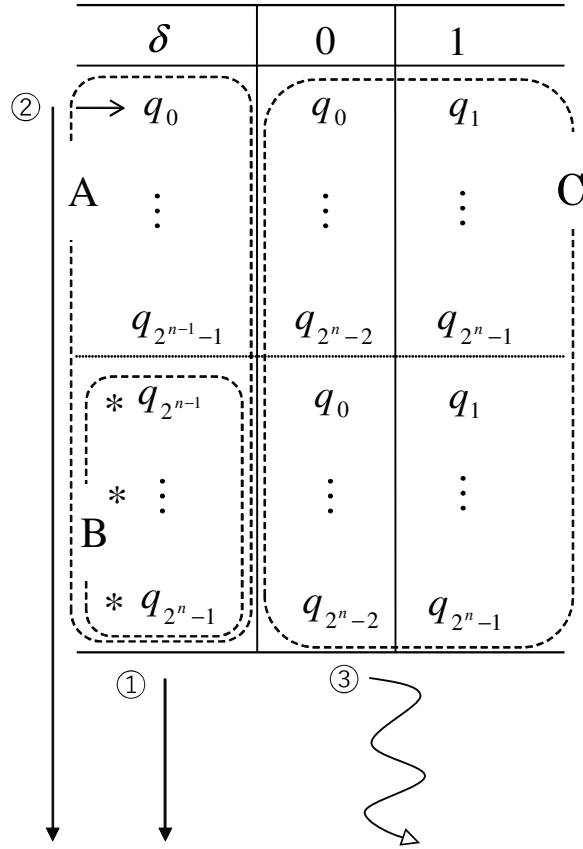


図 72: D_n の増加的部分集合構成法.

この項では、状態数が最小化された決定性色付き有限オートマトンの一意性を示す。一意性を示すことにより、 k 進 n 次元ドブリュイングラフ $DB_{k,n}$ が、正規言語 $(0 + 1 + \dots + k - 1)^* 1 (0 + 1 + \dots + k - 1)^{n-1}, \dots, (0 + 1 + \dots + k - 1)^* (k - 1) (0 + 1 + \dots + k - 1)^{n-1}$ で特徴づけられたと言える。

定義 30 DCFA $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^k F_i)$ とする。任意の $i = 1, \dots, k, q \in Q$, に対して、次のように定義する。

$$L_i(M, q) \triangleq \{x \in \Sigma^* \mid \hat{\delta}(q, x) \in F_i\},$$

すなわち、その初期状態が最初に与えられた q_0 でなく q であるような M によって i 色で受理される言語である。以下では、 $L_i(M, q)$ を $L_i(q)$ と略記する。

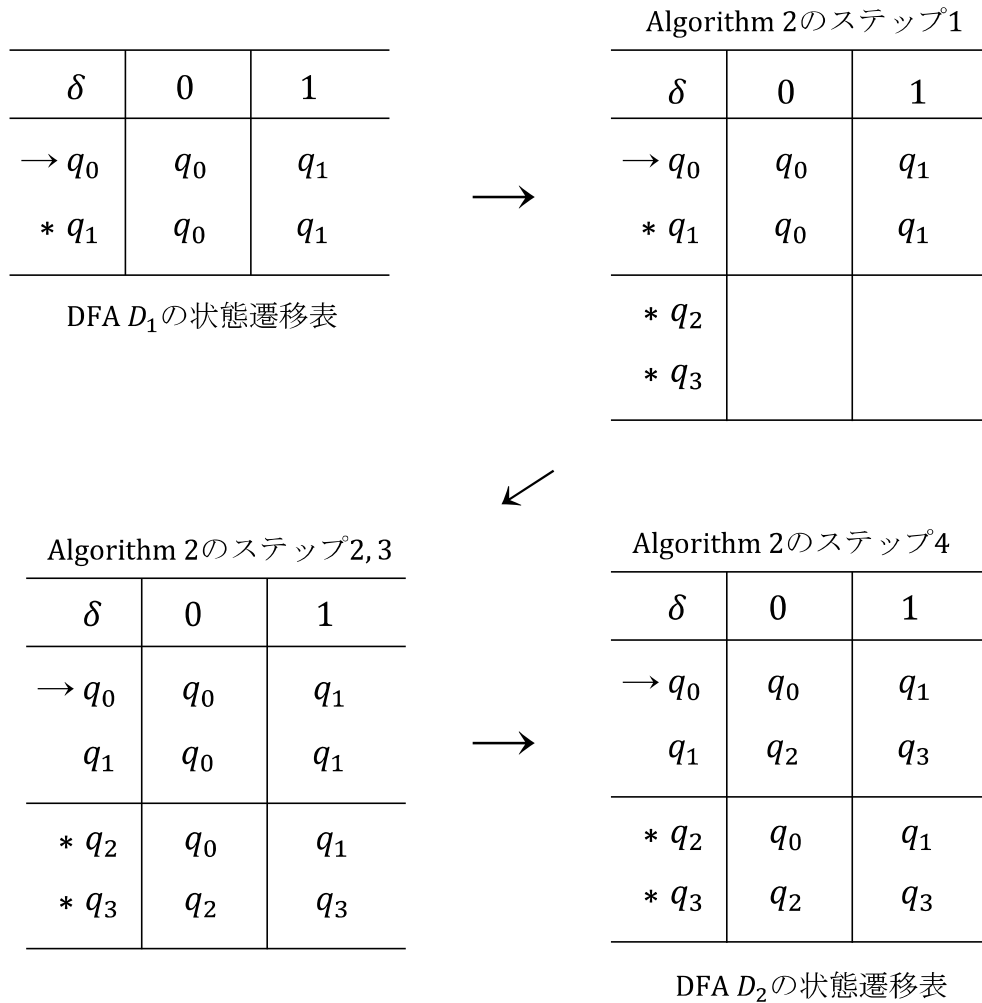


図 73: 増加的部分集合構成法による DFA D_1 から D_2 への変換過程.

定義 31 DCFA $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^k F_i)$ とする. 任意の $p, q \in Q$, に対して,

$$p \equiv q \iff \text{すべての } i = 1, \dots, k \text{ について, } L_i(p) = L_i(q)$$

と定義し, p と q は同値である (または区別不能) という.

事実 8 DCFA $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^k F_i)$ に対して,

$$\begin{aligned} p \equiv q &\iff (L_1(p), L_2(p), \dots, L_k(p)) = (L_1(q), L_2(q), \dots, L_k(q)) \\ &\iff (L_0(p), L_1(p), \dots, L_k(p)) = (L_0(q), L_1(q), \dots, L_k(q)). \end{aligned}$$

ここに, $L_0(q) \triangleq \Sigma^* - \sum_{i=1}^k L_i(M)$. 故に, $\sum_{i=0}^k L_i(q) = \Sigma^*$ より,

$$\begin{aligned} p \neq q &\iff \text{ある } i = 0, 1, \dots, k \text{ について, } L_i(p) \neq L_i(q) \\ &\iff \text{ある } i_1 \neq i_2 \text{ について, } L_{i_1}(p) \neq L_{i_1}(q) \text{ 且つ } L_{i_2}(p) \neq L_{i_2}(q). \end{aligned}$$

□

以下の議論では, NFA を表す線形方程式 [23, 35] の自然な拡張を使用する.

定義 32 DCFA $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^k F_i)$ とする.

任意の $q \in Q$ について, 状態 q の方程式を定義する:

$$q \rightarrow a_1 q_1 + a_2 q_2 + \dots + a_K q_K [+ \varepsilon_i],$$

ここに, $\delta(q, a_j) = q_j, j = 1, \dots, K, K = |\Sigma|$ であり右端の項 $\varepsilon_i, i = 1, \dots, k$ は $q \in F_i$ のときそのときのみ加えられる.

事実 9 DCFA $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^k F_i)$ とする. $q \in Q$ の方程式

$$q \rightarrow a_1 q_1 + a_2 q_2 + \dots + a_K q_K + \varepsilon_i$$

は以下の言語方程式と等価である. 各 $j = 1, \dots, k, j \neq i$ に対して,

$$L_j(q) = a_1 L_j(q_1) + a_2 L_j(q_2) + \dots + a_K L_j(q_K)$$

かつ

$$L_i(q) = a_1 L_i(q_1) + a_2 L_i(q_2) + \dots + a_K L_i(q_K) + \varepsilon.$$

定理 10 (状態数最小化 DCFA の一意性) 純色で色付けされた言語 $\sum L_i$ が与えられたとき, そのような $\sum L_i$ を純色で受理する任意の状態数最小 DCFA の遷移図は, 名前の付け替えを除いて同型である. すなわち, そのような 2 つの DCFA の状態の各組間に全単射が存在する.

(証明) DCFA $M = (Q, \Sigma, \delta, q_0, \sum_{i=1}^k F_i)$ と DCFA $M' = (Q', \Sigma, \delta', p_0, \sum_{i=1}^k F'_i)$ の両方が $\sum_{i=1}^k L_i$ を受理すると仮定する. $L(M) = L(M')$ から, 各 $i = 1, \dots, k$ に対して $L_i(q_0) = L_i(p_0)$, すなわち $q_0 \equiv p_0$ である. q_0 と p_0 の方程式を

$$\begin{cases} q_0 \rightarrow a_1 q_1 + a_2 q_2 + \dots + a_K q_K [+ \varepsilon] \\ p_0 \rightarrow a_1 p_1 + a_2 p_2 + \dots + a_K p_K [+ \varepsilon], \end{cases}$$

とする. これは以下と同値である.

$$\begin{cases} L_i(q_0) = a_1 L_i(q_1) + \dots + a_K L_i(q_K) [+ \varepsilon], i = 1, \dots, k \\ L_i(p_0) = a_1 L_i(p_1) + \dots + a_K L_i(p_K) [+ \varepsilon], i = 1, \dots, k \end{cases}$$

$$\Leftrightarrow \text{各 } i = 1, \dots, k, j = 1, \dots, K \text{ について, ある } L_i(q_j) = L_i(p_j)$$

$$\Leftrightarrow \text{各 } j = 1, \dots, K \text{ について, } q_j \equiv p_j.$$

同様にして, q_j と p_j の方程式から, M と M' の次のある状態間の同値性が得られる. この手順を続けると, $q_0 \in Q$ から到達可能な任意の $q \in Q$ は, $p_0 \in Q'$ から到達可能なある $p \in Q'$ と同値であると最終的に結論付けることができる.

一方, $q_1 \equiv p, q_2 \equiv p$ なる $q_1, q_2 \in Q, p \in Q', q_1 \neq q_2$ が存在するならば, $q_1 \equiv q_2$ が得られる. これは, DCFA M が状態数最小ではないことを意味する. 従って, そのような状態であってはならず, Q と Q' の間の対応は 1 対 1 でなければならない. □

謝辞

本論文は筆者が山口大学大学院創成科学研究科システム・デザイン工学系専攻在学中の研究成果をまとめたものである。指導教官で主査でもある同専攻の伊藤暁先生には、大学学部生時代から20年間に渡りご指導いただきました。また、社会人の私に研究の機会を与えていただき、懇切丁寧な御指導で、理論的研究を進める厳しさと一つ概念を突き詰めて熟考する大切さを御教授賜りました。心から感謝の意を表します。

本論文審査において副査を務めていただきました同専攻の多田村克己先生、王躍先生、河村圭先生、福士将先生、ならびに中正和久先生には、理論的研究の応用の可能性を主張する重要性や、工学的かつ読者を惹きつける論文の記述方法等を懇切丁寧かつ的確に御指導いただき、本論文をより学術的な内容に仕上げることができました。上記の先生方に心からお礼申し上げます。

最後に、これまで私の研究活動に関する行動に理解を示してくれて、研究の行く末をあたたく見守ってくれた家族に感謝いたします。

本論文に関する研究業績

論文誌

1. Y. Takahashi and A. Ito, “Finite Automata with Colored Accepting States and Their Unmixedness Problems,” Special section of IEICE Transactions on Information and Systems, Vol. E 105-D, No. 3, pp. 491–502, 2022.

国際会議

1. Y. Takahashi and A. Ito, “On Equivalence of de Bruijn Graphs and State-minimized Finite Automata,” The 20th Japan Conference on Discrete and Computational Geometry, Graphs, and Games, pp. 143-144, 2017.
2. Y. Takahashi and A. Ito, “Colored Finite Automata and de Bruijn Graphs,” The 22nd Japan Conference on Discrete and Computational Geometry, Graphs, and Games, pp. 117-118, 2019.
3. Y. Takahashi and A. Ito, “On the Unmixedness Problems of Colored Push-down Automata,” Proceedings of 2021 8th NAFOSTED Conference on Information and Computer Science (NICS), pp. 55–59, 2021.

国内学会

1. 高橋芳明, 伊藤暁, “ドブリュイニングラフと状態数最小化有限オートマトンの等価性について,” 電子情報通信学会技術報告, Volume 117, Number 28, pp. 107-113 (Web版では pp. 77-83), 2017.
2. 高橋芳明, 伊藤暁, “色付き有限オートマトンとドブリュイニングラフ,” 令和元年度(第70回)電気・情報関連学会中国支部連合大会論文集, R19-18-01-02, 2019.

3. 高橋芳明, 伊藤暁, “色付きプッシュダウン・オートマトンの非混色性問題について,” 2021年度(第72回)電気・情報関連学会中国支部連合大会論文集, R21-18-07, 2021.
4. 高橋芳明, 伊藤暁, “色付き有限オートマトンから色付き正規表現への変換アルゴリズム,” 2022年度(第73回)電気・情報関連学会中国支部連合大会論文集, R22-18-03, 2022.

受賞

1. 電子情報通信学会中国支部 奨励賞, “色付き有限オートマトンとドブリュイングラフ,” 2020年2月.