

論文 / 著書情報
Article / Book Information

題目(和文)	定量的性質を記述可能な拡張時間論理とそれを用いた仕様の検証手法に関する研究
Title(English)	
著者(和文)	富田 堯
Author(English)	Takashi Tomita
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第9328号, 授与年月日:2013年9月25日, 学位の種別:課程博士, 審査員:米崎 直樹,徳田 雄洋,権藤 克彦,渡部 卓雄,西崎 真也
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第9328号, Conferred date:2013/9/25, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

定量的性質を記述可能な拡張時間論理
とそれを用いた仕様の検証手法に関する研究

富田 堯

東京工業大学
大学院情報理工学研究科
計算工学専攻

平成 25 年度 博士論文

目次

第 I 部 序論	7
第 1 章 はじめに	9
1.1 背景と目的	9
1.2 本研究の内容	11
1.3 本論文の構成	13
第 2 章 定量的性質	15
2.1 定性的性質 vs. 定量的性質	16
2.2 確率と頻度	19
2.3 平均利得	21
第 3 章 準備	23
3.1 モデル	23
3.1.1 状態遷移系	24
3.1.2 離散時間 Markov 連鎖 (DTMC)	28
3.1.3 連続時間 Markov 連鎖 (CTMC)	31
3.1.4 利得付き状態遷移系	36
3.2 時間論理	40
3.2.1 線形時相論理 LTL	40
3.2.2 確率時間論理	43
3.3 オートマトン	51
3.3.1 ω -オートマトン	51
3.3.2 ω -正規オートマトン	53
3.3.3 多閾値平均利得オートマトン	55
3.3.4 言語クラス	58
3.4 リアクティブシステム	60
3.5 ゲーム	65

第 II 部 確率頻度時間論理とその検証手法 71

第 4 章 確率頻度時間論理 PFTL 73

4.1	構文	75
4.2	離散時間意味論	76
4.3	連続時間意味論	78
4.4	仕様記述例	80

第 5 章 PFTL モデル検査 83

5.1	CTL 型フラグメントモデル検査	84
5.1.1	DTMC に対するモデル検査	85
5.1.2	CTMC に対するモデル検査	88
5.2	LTL 型フラグメントモデル検査	93
5.2.1	DTMC に対するモデル検査	94
5.2.2	CTMC に対する統計的モデル検査	97
5.3	適用例	106
5.3.1	被食者-捕食者モデル	106
5.3.2	実験 1：逐次確率比検定による解析	108
5.3.3	実験 2：区間推定による解析	110

第 III 部 平均利得時間論理とその検証手法 113

第 6 章 平均利得線形時相論理 LTL^{mp} 115

6.1	構文	116
6.2	意味論	118
6.3	仕様記述例	123
6.4	LTL^{mp} vs. ω -Kleene 閉包言語	125

第 7 章 ω -正規オートマトンの平均利得拡張 127

7.1	多閾値平均利得 (MTMP) ω -正規オートマトン	128
7.2	閉包性	131
7.3	空判定	134
7.3.1	Büchi 受理実行存在の必要充分条件	135
7.3.2	平均利得受理実行存在の必要充分条件	136
7.3.3	線形制約を用いた平均利得受理実行存在の必要充分条件 の表現	139

7.3.4	空判定手続き	141
7.4	表現力	143
7.4.1	決定性 vs. 非決定性 MTMP ω -正規オートマトン	143
7.4.2	LTL ^{mp} vs. 決定性 MTMP ω -正規オートマトン	144
第 8 章	LTL^{mp} のモデル検査と充足可能性判定	149
8.1	LTL ^{mp} から MTMP ω -正規オートマトンへの変換	150
8.1.1	連言的 MPNF 式を満たすモデルのふるまいを受理する MPBA の構成法	150
8.1.2	LTL ^{mp} 文から等価な MTMPBA への変換	154
8.2	モデル検査	155
8.3	確率的モデル検査	156
8.4	充足可能性判定	158
8.5	最適化	158
第 9 章	LTL^{mp} 仕様の平均ケースの実現可能性と最適合成	165
9.1	LTL ^{mp} フラグメント LTL ^{mp-}	166
9.2	多全称閾値平均利得 Rabin ゲーム	169
9.3	ゲーム上の最適対抗戦略問題への帰着	171
9.4	MUTMPRG 最適対抗戦略の構成法	174
9.4.1	MUTMPRG 最適対抗戦略合成の概略	175
9.4.2	MUTMPRG 最適対抗戦略合成の部分手続き	177
9.4.3	MUTMPRG 最適対抗戦略合成手続き	195
9.5	計算量	199
第 IV 部	結論	203
第 10 章	関連研究	205
10.1	定量的言語	205
10.2	実時間論理	206
10.3	確率時間論理	207
10.4	平均利得時間論理	208
10.5	定量的ゲーム	209
第 11 章	おわりに	211

謝辞	215
参考文献	216
付録	230
付録 A 正規言語	233
付録 B 確率と統計	235
B.1 完全加法族と確率空間	235
B.2 Bernoulli 試行	237
B.3 Markov 性	237
B.4 統計的手法	239
B.4.1 区間推定	239
B.4.2 逐次確率比検定	240
付録 C 線形計画法	243

第I部

序論

第1章 はじめに

1.1 背景と目的

ハードウェアやソフトウェアの開発と検証には、しばしば**形式手法** (formal method) が用いられている。形式手法とは、システムやその**仕様**¹ (specification) を数学的に記述、検証する手法である。形式手法を用いて検証や開発を行う場合、コストが非常に高くなるものの、不具合がないことを数学的に保証することができる。そのため、高い安全性が求められる、不具合（バグ）が大きな損害を与えうるシステム（プログラム）の開発には非常に有効な手法である。そのようなシステムの主な例としては、組み込みシステムなどの各種制御システムなどが挙げられる。

システムの形式的検証には、しばしば**モデル検査**[37] (model checking) が用いられている。モデル検査は、与えられたシステムが仕様を満たすことを網羅的かつ自動的に検証する手法であり、多くのモデル検査器 [77, 67, 83, 70, 88] が開発されている。ここで、曖昧さがないように仕様が記述されていなければならないため、仕様記述言語として**時間論理** (temporal logic) がしばしば用いられている。また、大規模で複雑な仕様には欠陥（矛盾や不備）が含まれていることもあり、仕様自体の検証が必要とされることも多い。そのため、仕様の**無矛盾性**² (consistency) を検証する研究も行われてきた。しかしながら、各種制御システムのように、外部環境と継続的に相互作用するシステム、**リアクティブシステム**³ (reactive system, RS) の仕様は、無矛盾であるだけでは不十分である。なぜならば、RS は外部環境からの未来の入力を直接制御することはできないため、どのような入力に対しても、その時点の出力を過去情報のみを用いて適切に選択して仕様を満たさなければならないからである。そのように、仕様を満たすように適切に出力を選択し続けることが可能であるとき、すなわち、その仕様を実現する RS が存在するとき、その仕様は**実現可**

¹システムが満たすべき性質のこと。

²**充足可能性** (satisfiability) ともいう。

³しばしばプログラムと同一視される。

能 (realizable) あるいは**プログラム可能** (programmable) であるという。そして、仕様からそれを実現する RS を生成することを**合成** (synthesis) といい、時間論理などで記述された仕様から RS を自動合成する手法も研究されている [23, 68, 1, 72]。この自動合成による開発過程では、仕様が要件定義者の意図を正しく反映しているのであれば、不具合が混入する余地がない。

形式的な仕様記述や状態遷移系の検証では、系のふるまいの定性的な時間的性質が注目されてきた。例えば、

- 「常にある性質が成り立つ」,
- 「デッドロック状態へ到達する可能性がない」,
- 「与えられた要求に対して、いつか必ず適切な手順に従って応答を返す」,
- 「ある性質が無限にしばしば成り立つ」

などの、主に「系が備えるべき機能」に関する性質である。こうした性質の多くは、 ω -正規言語 (ω -regular language), すなわち, Büchi オートマトンなどの ω -**正規オートマトン**⁴ (ω -regular automata) が認識できる言語として表現可能である。そのため、 ω -正規オートマトンや、それらより表現力の弱い言語を直観的に定義できる**線形時相論理**[69] (Linear Temporal Logic, LTL) などが広く研究されてきた。そして、NuSMV[67, 36]⁵や SPIN[77, 52] などの LTL モデル検査ができるツールが開発された。また、LTL 実現可能性判定/合成問題は 2EXPTIME-完全 [68, 1, 72] であることが示され、Lily[61, 54], Acacia+[2, 17, 18] や Unbeast[82, 44] などの LTL 実現可能性判定器も開発されている。

一方で、より現実的な系を解析するために、定量的な時間的性質を記述、検証するための研究も行われてきた。例えば、

- 「要求に対して、5 秒以内に応答する」などの実時間制限に関する性質,
- 「要求に対して、一定時間以内に適切に応答する確率が 80% 以上である」などの確率的性質,
- 「故障状態である頻度が 3% 以下である」などの頻度的性質,

⁴単に ω -オートマトンと呼ぶこともある。

⁵仕様記述言語として**計算木論理**[38] (Computation Tree Logic, CTL) なども用いることもできる。

● 「長期的な平均利得の上界が10以下である」のような利得に関する性質などの、主に「系が備えるべき性能」に関する性質である。このような定量的性質を記述するための仕様記述言語として、**実時間論理**[5, 56, 94, 3, 9, 4, 20] (real-time logic) や**確率時間論理**[51, 11, 13, 12, 10] (probabilistic temporal logic) などが開発されてきた。また、UPPAAL[83, 15]などの実時間モデル検査器、PRISM[70, 58]やYmer[88, 89]などの確率的モデル検査器も開発されている。特に確率時間論理では実時間性や利得の期待値等の拡張も提案されている[13, 10]。ただし、これらの拡張時間論理で記述された仕様の検証は不能あるいは困難であることも多い。例えば、TPTL[5]やMTL[56]などの実時間論理は、モデル検査と充足可能性判定が決定不能である。また、最も単純な確率時間論理の一つであるPCTL[51]の充足可能性判定の決定可能性は未解決である。しかしながら、定量的性質に注目した拡張時間論理仕様には、**定量解析**(quantitative analysis)や**最適化**(optimization)といったより高度な検証手法を適用できるという大きな利点がある。例えば、確率的モデル検査器PRISMは、「ある性質を満たすふるまいの生起確率がある閾値以上であるか否か」を判定できるだけでなく、「ある性質を満たすふるまいの生起確率」そのものを計算することもできる。また近年になって、**平均利得**(mean-payoff)に関する目的関数に対して、最適にふるまうプログラムや戦略を合成する手法がいくつも提案されている[33, 16, 32, 86, 18]。さらに、LTLの平均利得拡張時間論理もいくつか開発されている[19, 17]。

本研究では、こうした定量的な性質を記述可能な拡張時間論理、及び、それによって記述されたまた、仕様の効果的、効率的な検証手法を開発することを目的としている。

1.2 本研究の内容

本研究では、定量的な性質を記述できる以下の二つの拡張時間論理を開発し、また、それらによって記述された仕様の検証手法を開発した。

- **確率頻度時間論理PFTL** (Probabilistic Frequency Temporal Logic)
- **平均利得線形時相論理LTL^{mp}** (LTL with Mean-Payoff constraints)

PFTL[80]は確率時間論理に頻度の概念を導入した時間論理である。この時間論理では、各状態における分岐の可能性に関する定量的性質(確率的性質)

と、分岐の結果として現れる時系列上の定量的性質（頻度的性質）を記述することができる。また、PFTLの意味論は離散時間だけでなく連続時間の上でも与えており、離散時間的性質だけでなく実時間的な性質の記述にも用いることができる。これにより確率的モデルのランダムなふるまいの性質をより直観的かつ詳細に記述できるようにした。また、離散時間/連続時間 Markov 連鎖 (Discrete-time/Continuous-time Markov chain, DTMC/CTMC) モデルに対する PFTL モデル検査手法を開発した [80, 96]。なお、本研究では、このモデル検査手法を以下の二つの PFTL フラグメントのモデル検査手法に分けて与えた。

- CTL 型フラグメント PFCTL
 - CTL と同様に、時間演算子の直前にパス限量子を置かなければならない PFTL フラグメント
- LTL 型フラグメント PFLTL
 - LTL⁶と同様に、パス限量子を部分式の中で用いることができない PFTL フラグメント

既存の CTL 型の確率時間論理に対しては数値計算ベースのモデル検査手法 [51, 11, 12, 59] が与えられており、DTMC/CTMC に対する PFCTL モデル検査には同様の手法を与えた [80]。また、DTMC に対する PFLTL モデル検査では、確率的 LTL モデル検査 [40, 41, 24] に帰着する手法を開発した。そして、厳密な検証が困難であった CTMC に対する PFLTL モデル検査については、**統計的手法**を用いた近似的モデル検査手法を与えた [80, 96]。この統計的アプローチ [92] は、通常のモデル検査では困難な確率微分方程式モデルなどの連続状態確率的モデルに対する検証にも用いることが可能である。また、既存の確率時間論理ではそれら連続モデルのふるまいの性質を直観的に記述することが困難であったが、PFLTL では新たに導入した頻度演算子を用いることで比較的容易に記述できる。

本研究で開発したもう一つの時間論理 LTL^{mp} [81, 97] は、LTL に平均利得の概念を導入した時間論理である。この時間論理では、モデルの実行上の遷移や状態、あるいは、その時間的性質の成立に対して発生する利得の平均的な性質について記述することができる。既存の平均利得拡張時間論理 LTL^{lim} [19]

⁶LTL はパス限量子を持たない時間論理であるが、モデル検査などの仕様としての LTL 式 ψ は「すべてのパスに関して ψ 」というパス限量が暗黙的に付加されているものとして解釈される。

では、モデルの実行上の平均利得に関する性質しか言及できなかったため、平均利得制約を含んだ仕様の充足可能性判定に用いることはできなかった。一方、 LTL^{mp} は、この $\lim$ を拡張した時間論理であり、時間的性質の成立に対して発生する利得を考慮できる。そのため、イベント生起頻度や平均コストなどの平均利得制約を含んだ仕様の充足可能性についても検証可能である。そして本研究では、 LTL^{mp} で記述された仕様に関するいくつかの検証手法を与えた [81, 98]。ここで、線形時間論理仕様の検証では、仕様と等価なオートマトンを構成し、そのオートマトンの検証問題に帰着して解くオートマトンベースの手法がしばしば用いられている。本研究でも、 LTL^{mp} 仕様と等価な拡張オートマトン、**多閾値平均利得** (multi-threshold mean-payoff, MTMP) ω -**正規オートマトン**の性質を検証することで各種検証問題を解く手法を与えた [81, 98]。MTMP ω -正規オートマトンは、 ω -正規オートマトンに重み関数と平均利得に関する受理条件 [6] を加えたオートマトンであり、その空判定問題は**線形制約充足問題** (linear constraint satisfaction problem, LCSP) に帰着できる [81]。そして、 LTL^{mp} のモデル検査と充足可能性判定は、この MTMP ω -正規オートマトンの空判定問題に帰着できることを示した [81]。また、 LTL^{mp} の充足可能性判定がこの LCSP に帰着して決定できることを利用し、 LTL^{mp} 仕様に従う平均利得目的関数を**線形計画法** (linear programming, LP) を用いて最適化する手法を開発した [81]。さらに、有限メモリをもった既知の環境下における、 LTL^{mp} のフラグメントで記述されたリアクティブシステム仕様の実現可能性判定/最適合成手法を開発した [98]。この合成手法は、システムの具体的な使用状況が分かっている場合に有効な手法であり、想定されている環境のもとで最適な性能を示すシステムを合成することができる。

1.3 本論文の構成

第 I 部の残りでは、まず、2 章で本研究で扱う定量的性質について非形式的に説明する。次に、3 章で準備として、モデルである状態遷移系、仕様を記述するための時間論理、 ω -オートマトンとそれに関連した概念や、確率や利得に関する拡張の定義を与える。

第 II 部は第 4 章と第 5 章からなり、確率頻度時間論理 PFTL とその検証手法について述べる。第 4 章では、確率時間論理に頻度の概念を導入した時間論理 PFTL の構文及び意味論を与える。そして第 5 章で離散時間/連続時間モデル検査手法について論じる。また、例として、既存の確率的モデル検査では検証困難な確率的振動系を PFTL モデル検査で検証できることを示す。

第III部は第6章から第9章よりなり、平均利得線形時間論理 LTL^{mp} とその検証手法について述べる。第6章では、 LTL に平均利得式を導入した時間論理 LTL^{mp} の構文及び意味論を与える。第7章では、 LTL^{mp} 文で定義される言語を認識できる拡張 ω -正規オートマトン、 $MTMP$ ω -正規オートマトンの定義を与える。また、 $MTMP$ ω -正規オートマトンの空判定アルゴリズムを与え、さらに、その表現力についても考察する。第8章では、 LTL^{mp} 仕様のモデル検査と充足可能性判定問題が $MTMP$ ω -正規オートマトンの空問題に帰着して決定できることを示す。また、線形計画法を用いて LTL^{mp} 仕様を充足可能な範囲で最適化する手法を与える。そして、第9章では、確率的環境のもとで LTL^{mp} フラグメント仕様を実現する、平均ケースにおいて最適なプログラムを合成する手法を与える。

第IV部は第10章と第11章からなる結論である。第10章では関連研究について述べる。最後に、第11章でまとめる。

また、付録で正規言語、確率と統計、線形計画法に関する基本的定義を補完する。

第2章 定量的性質

システムが満たすべき性質を仕様という。ソフトウェア/ハードウェアの仕様は、システムの利用者や設計者などの利害関係者の**要求** (requirement) を反映して規定される。また、生命システムなどの検証では、仕様は解析対象システムのふるまいの特性として規定される。システムが仕様を満たす（不具合がない）ことを厳密に検証するためには、まず、曖昧性がないように仕様が記述されてなくてはならない。自然言語で記述された仕様は、その意味解釈に曖昧性をもつため、システムがその仕様を満たすか否かの厳密な検証は一般に不能である。また、仕様には機能や性能、コストなどシステムの様々な属性に関する性質が含まれるが、すべての性質を曖昧性なく記述し、また、厳密に検証することは困難である。例えば、「ユーザ・インターフェースの使いやすさ」や「メンテナンスのしやすさ」などのシステム利用者/運用者の主観に影響される属性は、形式的に一般的な定義を与えることは困難である。そのため、形式的検証の対象となる仕様は、その意味解釈に曖昧性をもたない形式的な言語で記述されていなくてはならない。形式的な仕様記述言語としては、時間的性質を記述できる様相論理である**時間論理** (temporal logic) がしばしば用いられている。時間的性質は、一つの時間系列に関する性質である**線形時間的性質** (linear-time property) と状態のもつ分岐の可能性に関する性質である**分岐時間的性質** (branching-time property) に分けることができ、それらの性質を記述できる時間論理をそれぞれ**線形時間論理** (linear-time logic) と**分岐時間論理** (branching-time logic) という。線形時間的性質の例としては、以下のような性質がある。

- 「いつかある状態 s に到達する」
- 「常にある性質 ψ を満たす」
- 「ある性質 ψ が成り立った後に、A, B, C というイベントが順に生起する」

一方で、分岐時間的性質は、ある線形時間的性質 ψ を満たすふるまいが生起する可能性に関する以下のような性質である。

- 「生起する可能性のあるいくつかのふるまいが性質 ψ を満たす」
- 「生起する可能性のあるすべてのふるまいが性質 ψ を満たす」

このような分類とは別に、時間的性質は、**定性的性質** (qualitative property) と **定量的性質** (quantitative property) に分けることができる。本章では、定性的性質と定量的性質の違い、及び、本研究で扱う定量的性質について非形式的に説明する。

本章の構成

まず、2.1 節で定性的性質と定量的性質の違いについて述べる。2.2 節では、4 章で与える定量的拡張時間論理で導入されている確率と頻度の概念について説明する。そして、2.3 節では、6 章で与える定量的拡張時間論理で導入されている平均利得 (mean-payoff) の概念について説明する。

2.1 定性的性質 vs. 定量的性質

ソフトウェア/ハードウェア仕様は、利害関係者の要求を反映して規定される。要求は、大きく分けて**機能要求** (functional requirement) と**非機能要求** (non-functional requirement) に分類できる。

- 機能要求：「システムが備えるべき機能」に関する要求
- 非機能要求：「システムが備えるべき性能」に関する性能要求などの要求

「システムが備えるべき機能」に関する仕様の多くは、以下のような定量的な言及を含まない性質として与えることができる。

- 「常にある性質が成り立つ」
- 「デッドロック状態へ到達する可能性がない」
- 「与えられた要求に対して、いつか必ず適切な手順に従って応答を返す」
- 「ある性質が無限にしばしば成り立つ」

このような、量的な言及を含まない性質を**定性的性質**という。システムの検証ではシステムの機能的側面に注目することが多く、定性的仕様を形式的に記述/検証する手法が研究されてきた。定性的仕様記述言語としては LTL[69] や CTL[38] などの古典的な時間論理がしばしば用いられており、**モデル検査**[37] (model checking) や**充足可能性判定** (satisfiability checking), **実現可能性判定**[68, 1, 72] (realizability checking) などの検証手法が開発されている。

- モデル検査
 - 入力：モデル M と仕様 φ
 - 出力：モデル M (のすべてのふるまい) が仕様 φ を満たすか否か (yes/no)
- 充足可能性判定
 - 入力：仕様 φ
 - 出力：仕様 φ を満たすモデルが存在するか否か (yes/no)
- 実現可能性判定
 - 入力：リアクティブシステム (RS) 仕様 φ
 - 出力：仕様 φ を実現する RS が存在するか否か (yes/no)

これらの検証は「量」に言及しないため、**定性的検証** (qualitative verification) と呼ばれる。また、検証結果は付加的な情報も合わせて出力することも多い。例えば、モデル検査では、モデル (のすべてのふるまい) が仕様を満たさない場合には仕様を満たさないふるまい (反例) を生成して出力する。また、充足/実現可能性判定では、仕様が充足/実現可能な場合には仕様を充足/実現するモデル/RS を生成/合成して出力する。

一方で、「システムが備えるべき性能」に関する仕様の多くは、以下のような定量的な言及を含む性質である。

- 「要求に対して、5 秒以内に応答する」 (実時間束縛に関する性質)
- 「要求に対して、一定時間以内に適切に応答する確率が 80% 以上である」 (確率的性質)
- 「故障状態である頻度が 3% 以下である」 (頻度的性質)

- 「長期的な平均利得の上界が10以下である」(利得¹に関する性質)

このような定量的な言及を含む性質を**定量的性質** (quantitative property) という。これらの定量的仕様記述言語としては、実時間論理や確率時間論理などの定量的拡張時間論理がしばしば用いられている。多くの定量的拡張時間論理で記述できる性質は、“ある性質を満たすまでの時間”や“ある性質を満たすふりまいの生起確率”などの“注目している量”に関する閾値条件であり、真偽評価が可能な性質である。この意味ではそれらの定量的拡張時間論理は定性的であるとも言えるが、定量的拡張時間論理のモデル検査など検証は「量」に言及するため、**定量的検証** (quantitative verification) と呼ばれる。また、その性質が満たされるか否かだけでなく、“注目している量”について定量的に評価可能な場合もある。そのため、定量的性質を含んだ仕様は、「満たすか否か」のような検証だけでなく、**定量解析**や**最適化**などに用いることができる。

- 性能解析

- 入力：モデル M と変数 θ
 - * θ は “ある性質 ψ を満たすふりまいの生起確率” などの “注目している量” を表す
- 出力：モデル M についての変数 θ の値

- 仕様最適化

- 入力：仕様 φ と目的関数 θ
- 出力：仕様 φ に従う目的関数 θ の最適値 (とその最適値を与えるモデル)

- 最適合成

- 入力：RS 仕様 φ と目的関数 θ
- 出力：仕様 φ に従い、目的関数 θ を最適化する RS とその最適値

これらの手法では、厳密値と近似値の差を定量的に評価するため、近似が有効であることも多い。例えば、確率的モデル検査では統計的手法を用いた近似的モデル検査手法 [92] が開発されている。また、定量的合成手法においても厳密に最適な RS ではなく、 ε -最適な RS を合成する手法が多い。これは、一般に、

¹一般的な呼び方としては「重み (weight)」が使われるが、ゲーム理論では主に「利得 (payoff)」, Markov 連鎖では主に「報酬 (reward)」や「コスト (cost)」と呼ぶ。

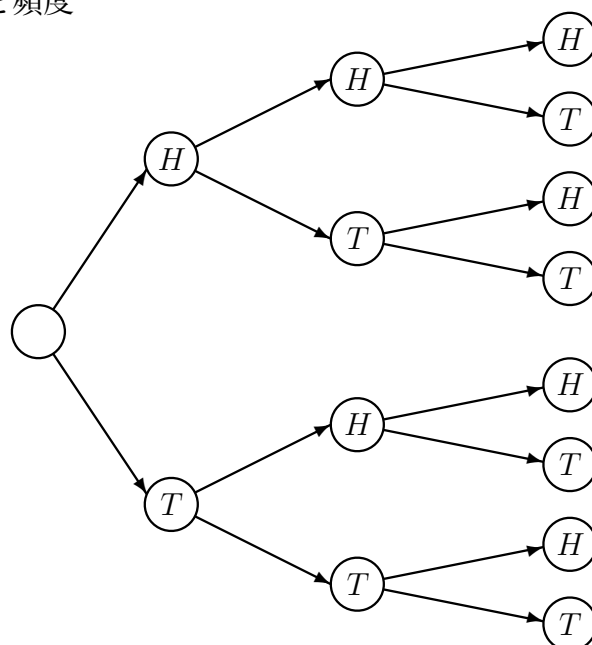


図 2.1: コイン投げ.

厳密に最適な RS は無限メモリ（非有界なサイズの過去情報）を要求しうるが、そのような RS は実用的でないためである。

2.2 確率と頻度

定性的な分岐時間的性質では、「 ψ を満たすふるまいが生起する可能性がある（あるパスで ψ を満たす）」ことや「 ψ を満たすふるまいがかならず生起する（すべてのパスで ψ を満たす）」ことしか表現できない。しかしながら、実際のシステムでは、生起する可能性のあるふるまいの中でも、「生起しやすいふるまい」と「生起しにくいふるまい」が存在していることも多い。そのようなシステムは、各時間で確率的に遷移（あるいはイベント）が生起する確率的システムとしてモデル化される。形式的検証の対象となるモデルは、ほとんどの場合、半永久的に動作し続けるシステムであり非可算無限個の結果（ふるまい）が存在するが、確率的モデルの上では分岐の可能性を定量的に評価することができる。例えば、「表が出る確率が p ($0 \leq p \leq 1$) であるコイン」を投げて「表」か「裏」のどちらが出たか観測することを無限回繰り返す試行を考える。（図 2.1）この場合、「最初の 3 回すべてが表」であるふるまいの生起確率は p^3 と評価できる。

確率的性質とはこのような定量的分岐時間的性質であり、PCTL[51] や CSL[11, 12] などの確率時間論理で記述することができる。

一方で、頻度は「分岐の結果（ふるまい）上の時間の割合」に関する定量的な線形時間的性質である。定性的な線形時間的性質では、「いつか（ある時間で） ψ が成り立つ」や「常に（すべての時間で） ψ が成り立つ」、「無限にしばしば ψ が成り立つ」などしか表現できない。定量的な線形時間的性質としては、「“時間区間 I 中の” すべての時間で ψ が成り立つ」や「“時間区間 I 中の” ある時間で ψ が成り立つ」などの実時間的な時間束縛に関する性質がしばしば扱われるが、その場合でも、注目している時間区間中の定性的な線形時間的性質に言及しているにすぎない。しかしながら、ふるまい上（あるいはその一部の時間区間中）では「多くの時間で成り立つ性質」や「少しの時間でしか成り立たない性質」が存在しうる。例えば、自然数列 $0, 1, \dots$ を考えると、偶数や素数は可算無限個存在し数列上には無限にしばしば出現する。ここで、 n 以下の偶数の個数を n_{even} としたとき、「偶数の出現頻度」、すなわち、 n_{even}/n は n を無限大に近づけた場合には $1/2$ に収束する。一方、 n 以下の素数の個数を n_{prime} としたとき、「素数の出現頻度」 n_{prime}/n は n を無限大に近づけた場合には 0 に収束する。つまり、ほとんどすべての自然数は素数でない。このように、頻度の概念を用いることで、ふるまい上の定量的性質を詳細に捉えることができる。

頻度は確率と深く関連した概念であり、頻度によって確率を定義することもできる²。前述の「表が出る確率が p ($0 \leq p \leq 1$) であるコイン」を無限回投げ続ける試行では、 n 回目までに表が出た回数を n_{head} としたとき、ほとんどすべての試行結果で「表の出現頻度」 n_{head}/n は、 p に収束するはずである。このように、それぞれの時間で同一の確率で生起するイベントを無限回の繰り返す試行によって、イベントの生起確率を定義する立場を頻度主義とよぶ。ここで、それぞれの時間のイベント生起確率が同一でない場合には、それぞれの時間で事象確率は頻度を用いて捉えることはできないことに注意が必要である。しかし、この場合でも頻度を「確率の平均値」として見なすことができる。このように頻度は確率的ふるまいの特徴を規定する概念として適しているといえる。

第II部では、本節で述べた頻度の概念を導入した拡張確率時間論理と、その検証手法について述べる。

²そのような立場を頻度主義と呼ぶ。

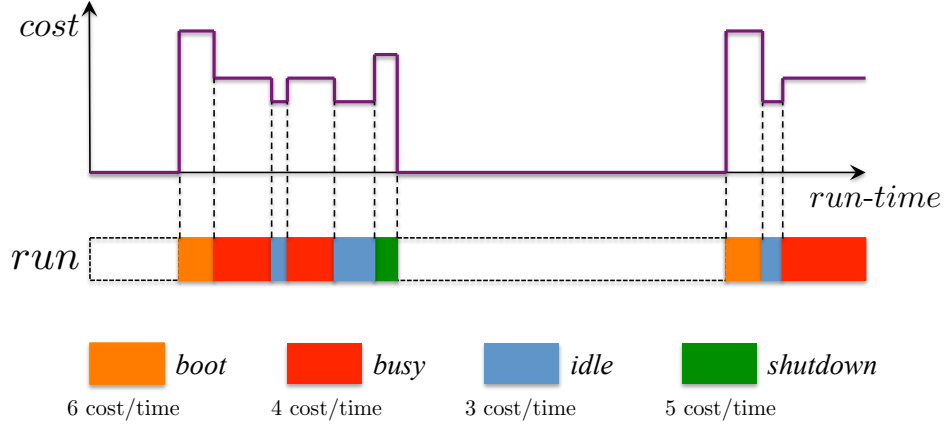


図 2.2: 処理モジュールの実行コスト

2.3 平均利得

システムをモデル化するには、しばしば「イベント e の生起」と「イベントに対応する命題 p_e の成立」を同一視する。そして実用的なシステムでは、動作（イベントの生起）に何らかの資源やコストが必要なことも多い。そのため、利得（コスト）に関する定量的な検証もしばしば求められている。例えば、起動時 (*boot*)、動作時 (*busy*)、アイドル時 (*idle*)、シャットダウン時 (*shutdown*) にそれぞれ単位時間当たり 6, 4, 3, 5 のコストが必要な処理モジュールの実行コストは図 2.2 のように与えられる。利得に関する性質には様々なものがあるが、形式的検証では累積利得 (accumulated payoff) や平均利得 (mean-payoff)、そしてその期待値に注目することが多い。

- 累積利得：注目している時間区間中に発生する利得の総和。
- 平均利得：注目している時間区間中に発生する利得の平均。

時間区間が非有界であるときには、累積利得や平均利得に関する性質は、多くの場合、その上極限/下極限に関する性質のことを指す。また、その期待値については、報酬構造拡張した確率的モデル検査 [13, 10] ではしばしば解析の対象となっている。

平均利得の概念は、長期的な実行上の平均コストやイベント生起頻度に関する性質を捉えるために用いることができる。例えば、「イベント $e_1, \dots, e_n$ に注目した場合の平均コスト」は「 $p_{e_1}, \dots, p_{e_n}$ の成立に対してそれぞれ $c_1, \dots, c_n$

の利得が発生する場合の平均利得」であり、また、イベント e の生起頻度は、「 p_e の成立に対して 1 の利得が発生する場合の平均利得」と等しい。平均利得は元々はゲーム理論で用いられていた概念であるが [45]、近年になってから定量的合成 [33, 16, 32, 86, 18] において注目されている。ゲーム理論は複数の主体の意思決定に関する理論であり、定量的合成における RS と環境という二つの意思決定主体の相互作用は発展型二人交代手番ゲームとしてモデル化できる。発展型二人交代手番ゲームは、チェスや将棋のような二つの意思決定主体がそれぞれの勝利条件を満たすことを目的として交互に手を選ぶことで進行するゲームである。この勝利条件は様々な種類があり、チェスや将棋では「相手の王をとること」が勝利条件となる³。そして、実現可能性判定/合成問題では、「仕様を満たす/満たさないこと」が RS/環境の勝利条件となるゲームを構成しそのゲーム上の勝利戦略問題に帰着して解くことができる [25, 29, 26, 34]。

第 III 部では、LTL に平均利得の概念を導入した拡張時間論理と、その検証手法について述べる。

³将棋やチェスでは引き分けもあり得る

第3章 準備

本章では準備として、システムのふるまいを抽象化したモデルとしてしばしば用いられている状態遷移系とそれに確率や利得の概念を導入した拡張状態遷移系、時間的性質を記述できる様相論理である時間論理と確率時間論理、無限長の語に対して受理/不受理を判定するオートマトンである ω -オートマトンと、それらに関連した概念の定義を紹介する。また、外部環境と継続的に相互作用するリアクティブシステムと、リアクティブシステムと環境の相互作用をモデル化するゲームの定義とそれに関連した概念についても述べる。

なお、最も基本的な有限長語のクラスである正規言語の定義は付録A、確率と統計に関連した基本的な概念や定義は付録B、また、最適化手法である線形計画法については付録Cで簡単に説明する。

本章の構成

まず、3.1節で本研究で解析の対象となる最も基本的なモデルである状態遷移系と関連した概念の定義を与える。また、確率の概念が導入された状態遷移系であるMarkov連鎖と、利得の概念が導入された系についてもここで紹介する。次に、3.2節で最も基本的な時間論理である線形時相論理LTLと離散時間確率時間論理であるPCTL、連続時間確率時間論理であるCSLの構文と意味論を簡単に紹介する。そして、3.3節で、 ω -正規言語を認識するオートマトンである ω -正規オートマトンと、それらとは異なり多閾値平均利得条件を受理条件としてもつオートマトンを紹介する。また、それらの表現力に関して簡単に述べる。3.4節では、リアクティブシステムとゲームに関する概念と定義を紹介する。

3.1 モデル

解析の対象となる現実のシステムの注目している本質を抽出して抽象化したものをモデルという。そして、モデルの性質を検証することで、システムの

構造やふるまいの理解と予測が可能となる。本論文では解析の対象としていくつかのモデルを扱うが、本節では最も基本的なモデルの一つである **ラベル付き状態遷移系** (labeled transition systems, LTS) と関連した概念について簡単に紹介する。また、その確率的拡張である **Markov 連鎖** (Markov chain, MC)、及び、利得の概念が導入された状態遷移系/MC についても紹介する。

3.1.1 状態遷移系

状態遷移系は、システムのふるまいを記述するモデルとしてしばしば用いられている。状態遷移系は離散的な状態の集合と、それらの間の遷移関係からなる。状態は、システムの現在の状況、例えば変数の現在の値の集合などを示す。また多くの場合、各状態にはその状態で満たしている命題の集合がラベル付けされている。そのような状態遷移系をラベル付き状態遷移系（以下、単に状態遷移系と呼ぶ）という。このラベル付けされる命題は、モデル化と解析の際に注目する最も基本的な性質である。遷移関係 (transition relation) は、システムの現在の状態が次の時間でどの状態に変化できるかを規定する。そして、状態遷移系は与えられた初期状態から遷移関係に従って現在の状態から遷移できる状態のうちの一つへ次々に遷移していく。その状態の列が状態遷移系のふるまいであり、そのふるまいの時間的性質は各状態のラベルの列で特徴付けられる。

形式的には、以下のように定義される。

定義 3.1 (状態遷移系) 状態遷移系 T は五つ組 $\langle S, \rightarrow, S_{init}, AP, \Lambda \rangle$ である。ここで、

- S は可算の状態集合,
- $\rightarrow \subseteq S^2$ は遷移関係,
 - ただし, $\forall s. \exists s'. \langle s, s' \rangle \in \rightarrow$,
- $S_{init} \subseteq S$ は初期状態の集合,
- AP は原子命題の有限集合,
- $\Lambda : S \rightarrow 2^{AP}$ は各状態で成り立つ原子命題を割り当てるラベリング関数 (labeling function) である。 ■

遷移関係 $\rightarrow$ に含まれる対 $\langle s, s' \rangle$ は、現在の状態 s から状態 s' へ遷移できることを表している。また、定義 3.1 では、どんな状態も一つ以上の遷移先をもっていることを前提としている。本論文ではこのような停止しない状態遷移系のみを考える。

状態遷移系のふるまいは、現在の状態から遷移可能な状態を非決定的に選択していくことで得られる状態列として定義される。これはしばしば実行 (execution, run) と呼ばれるが、本論文では**パス** (path) と呼ぶ。

定義 3.2 (パス) $\mathcal{T} = \langle S, \rightarrow, S_{init}, AP, l \rangle$ を状態遷移系とする。このとき、以下を満たす無限状態列 $\rho = s_0 s_1 \dots \in S^\omega$ を $\mathcal{T}$ のパスと呼ぶ。

- $s_0 \in S_{init}$,
- 任意の $i \in \mathbb{N}$ について、 $\langle s_i, s_{i+1} \rangle \in \rightarrow$ 。

また、 $\mathcal{T}$ のすべてのパスの集合を $\text{Paths}^{\mathcal{T}}$ と書く。 ■

ここで、状態の無限列であるパスに限らず、ある集合の要素の無限列 $seq = e_0 e_1 \dots$ に対して、 $seq[i]$ で i 番目の要素 e_i 、 $seq[i \dots]$ で i 番目以降の無限列 $e_i e_{i+1} \dots$ を表すものとする。また、 $\mathcal{T}$ の初期状態集合 S_{init} を単一状態 $\{s\}$ で置き換えた状態遷移系のすべてのパスの集合を $\text{Paths}_s^{\mathcal{T}}$ と書く。

状態遷移系のパスの原子命題集合 AP によって特徴付けられる時間的性質は、各状態にラベル付けされた原子命題部分集合の列によって与えられる。一般に、有限のアルファベット Σ の要素の無限列を、 Σ 上の**無限長語** (infinite word, ω -word) と呼ぶ。

定義 3.3 (文字, 語, 言語) Σ をアルファベットとする。このとき、 $a \in \Sigma$ を文字、文字の無限列 $\sigma = a_0 a_1 \dots \in \Sigma^\omega$ を無限長語、無限長語の集合 $L \in 2^{\Sigma^\omega}$ を無限長言語と呼ぶ。 ■

本論文では、以後、無限長語と無限長言語をそれぞれ単に語と言語と呼ぶ。

そして、 ρ 上のラベルの列を ρ 上の語という。

定義 3.4 (パス上の語) $\mathcal{T} = \langle S, \rightarrow, S_0, AP, \Lambda \rangle$ を状態遷移系とする。このとき、 $\mathcal{T}$ のパス $\rho = s_0 s_1 \dots \in \text{Paths}^{\mathcal{T}}$ に対して、語 $\Lambda(s_0) \Lambda(s_1) \dots \in (2^{AP})^\omega$ を ρ 上の語といい、 $\text{word}_\Lambda(\rho)$ と書く。また、 $\mathcal{T}$ のすべてのパス上の語の集合 $\{\text{word}_\Lambda(\rho) \in (2^{AP})^\omega \mid \rho \in \text{Paths}^{\mathcal{T}}\}$ を $\text{Words}^{\mathcal{T}}$ と書く。 ■

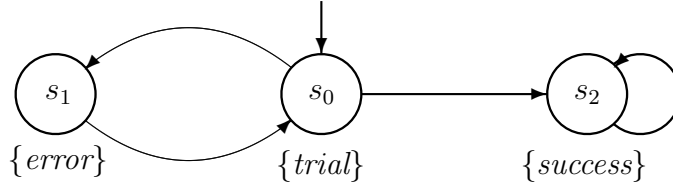


図 3.1: 例 3.5 の試行錯誤系

また、パスの集合と同様に、 $\mathcal{T}$ の初期状態集合 S_{init} を単一集合 $\{s\}$ で置き換えた状態遷移系のパス上の語の集合を $\text{Words}_s^{\mathcal{T}}$ と書く。

例 3.5 次のような状態遷移系

$$\mathcal{T} = \langle \{s_0, s_1, s_2\}, \rightarrow, \{s_0\}, \{trial, error, success\}, \Lambda \rangle$$

を考える。(図 3.1)。ただし,

$$\rightarrow = \{ \langle s_0, s_1 \rangle, \langle s_0, s_2 \rangle, \langle s_1, s_0 \rangle, \langle s_2, s_2 \rangle \}, \quad (3.1)$$

$$\Lambda(s) = \begin{cases} trial & \text{if } s = s_0, \\ error & \text{if } s = s_1, \\ success & \text{otherwise.} \end{cases} \quad (3.2)$$

この系は直観的には、いつか成功 *success* し続けるか、延々と試行 *trial* と失敗 *error* を繰り返す系である。 ■

到達可能性

状態遷移系のふるまいの時間的性質は、しばしば**到達可能性** (reachability) を基に解析できる。ここで、状態遷移系の遷移関係に限らず、状態間の遷移関係上の到達可能関係は以下のように定義される。

定義 3.6 (到達可能関係) S を状態の集合、 $\rightarrow \subseteq S^2$ を S 上の遷移関係とする。このとき、状態間の到達可能関係 $\rightarrow^* \subseteq S^2$ は、 $\rightarrow$ の推移閉包

$$\rightarrow^+ = \{ \langle s_1, s_n \rangle \in S^2 \mid \exists s_1, s_2, \dots, s_n. \forall i \in [1, n). (\langle s_i, s_{i+1} \rangle \in \rightarrow) \}$$

と恒等関係 $\{\langle s, s \rangle \mid s \in S\}$ の和集合である。また、 $\langle s, s' \rangle \in \rightarrow^*$ のとき、 s' は s から **到達可能** (reachable) であるという。 ■

以後は、遷移関係 $\rightarrow$ について、 $\langle s, s' \rangle \in \rightarrow$ を $s \rightarrow s'$ と表記する。また、遷移関係の推移閉包 $\rightarrow^+$ や到達可能関係 $\rightarrow^*$ についても同様の表記法を用いる。

そして、状態集合 S 上の遷移関係 $\rightarrow$ によるふるまいの特性は、状態の集合 S を互いに到達可能な状態の部分集合 $S' \subseteq S$ に分割したときの構造的特徴によってしばしば解析できる。そのような互いに到達可能な状態の部分集合を **強連結成分** (strongly-connected component, SCC) と呼ぶ。

定義 3.7 (強連結成分, 極大強連結成分, 底強連結成分) S を状態の集合、 $\rightarrow \subseteq S^2$ を S 上の遷移関係とする。このとき、以下を満たす部分状態集合 $S' \subseteq S$ と部分遷移関係 $\rightarrow' \subseteq S' \times S' \cap \rightarrow$ の対 $\langle S', \rightarrow' \rangle$ を強連結成分、強連結成分の中で極大なものを **極大強連結成分** (maximal SCC, MSCC) と呼び¹、 $\mathcal{T}$ 上の MSCC の集合を $\text{MSCC}^{\mathcal{T}}$ と書く。

- 任意の $s, s' \in S'$ について、 $s(\rightarrow')^* s'$ 。

また、以下を満たす強連結成分 $\langle S', \rightarrow' \rangle$ を、**底強連結成分** (bottom SCC, BSCC) と呼び、 $\mathcal{T}$ 上の BSCC の集合を $\text{BSCC}^{\mathcal{T}}$ と書く。

- 任意の $s \in S$, $s' \in S'$ について、 $s' \rightarrow' s$ ならば $s \in S'$ 。

なお、SCC $\langle S', \rightarrow' \rangle$ に関して、遷移関係に言及する必要がある場合、その状態部分集合 S' のことをしばしば (M/B)SCC と呼ぶ。

ここで、複数の MSCC が交わることはない。もし交わるのであればそれらの和集合も SCC であり、もとの MSCC の極大性に反する。また、有限状態の状態遷移系上では、少なくとも一つは BSCC が存在し、どんな状態も少なくとも一つの BSCC へ到達可能である。

また、与えられたグラフ (状態集合とその間の遷移関係) 上の MSCC の集合 $\text{MSCC}^{\mathcal{T}}$ を計算することを SCC 分解といい、多項式時間で計算できることが知られている [79]。

例 3.8 例 3.5 の状態遷移系 (図 3.1) を考える。このとき、 $s_2 \not\rightarrow s_0$, $s_2 \not\rightarrow s_1$, $s_1 \rightarrow s_0$, $s_2 \rightarrow s_0$ である。ここで、 $\langle \{s_0\}, \emptyset \rangle$ のよう単一状態と空遷移関係からなる対は SCC である。このような SCC を **自明な SCC** と呼ぶ。また、MSCC としては、 $\langle \{s_0, s_1\}, \{\langle s_0, s_1 \rangle, \langle s_1, s_0 \rangle\} \rangle$ と $\langle \{s_2\}, \{\langle s_2, s_2 \rangle\} \rangle$ が得られる。 ■

¹極大強連結成分は単に強連結成分と呼ばれることもあるが、本論文では極大でないものも強連結成分と呼ぶ。

3.1.2 離散時間 Markov 連鎖 (DTMC)

状態遷移系では各時間で遷移可能な状態へ非決定的に遷移するが、ある性質を持ったふるまいの起こりやすさを定量的に評価することはできない。しかし、現実的なシステムには、しばしば起こりやすいふるまいと起こりにくいふるまいが存在する。そのようなシステムのモデル化にはしばしば Markov 連鎖 (MC) が用いられる。MC は、系の未来の状態の確率分布が現在の状態のみに依存する²離散状態確率過程である。そして、遷移が生起する時間間隔が離散的な確率過程を**離散時間 Markov 連鎖** (discrete-time MC, DTMC) という。

形式的には以下のように定義される。

定義 3.9 (離散時間 Markov 連鎖) DTMC $\mathcal{D}$ は五つ組 $\langle S, \mathbf{P}, P_{init}, AP, \Lambda \rangle$ である。ここで、

- S は可算の状態集合,
- $\mathbf{P} : S^2 \rightarrow [0, 1]$ は状態をインデックスとした遷移確率行列 (transition probability matrix),
 - ただし、任意の状態 $s \in S$ について、 $\sum_{s' \in S} \mathbf{P}(s, s') = 1$,
- $P_{init} : S \rightarrow [0, 1]$ は初期状態分布 (initial distribution),
 - ただし、 $\sum_{s \in S} P_{init}(s) = 1$,
- AP は原子命題の有限集合,
- $\Lambda : S \rightarrow 2^{AP}$ はラベリング関数である。 ■

DTMC $\mathcal{D} = \langle S, \mathbf{P}, P_{init}, AP, \Lambda \rangle$ は状態遷移系

$$\mathcal{T}^{\mathcal{D}} = \langle S, \{\langle s, s' \rangle \mid \mathbf{P}(s, s') > 0\}, \{s \mid P_{init}(s) > 0\}, AP, \Lambda \rangle$$

の遷移と初期分布に確率の概念を導入したモデルである。DTMC のパスの集合 $\text{Paths}^{\mathcal{D}}$ とパス上の語については、この状態遷移系のパスの集合 $\text{Paths}^{\mathcal{T}^{\mathcal{D}}}$ とパス上の語として定義される。

遷移が非決定的に選ばれる状態遷移系とは異なり、DTMC では遷移確率行列 $\mathbf{P}$ に従って確率的に遷移が選択される。ここで、 $\mathbf{P}(s, s')$ は状態 s から状態 s' へ 1 ステップで遷移する確率を表す。そのため、DTMC のパスの生起確率

²この性質を Markov 性という (付録 B.3 節)。

は $\mathbf{P}$ で特徴付けられる。ただし、 $\mathcal{D}$ のパスの集合 $\text{Paths}^{\mathcal{D}}$ は一般に非可算集合であるため、その集合族 $2^{\text{Paths}^{\mathcal{D}}}$ 上では単純には確率を定義できない。無限列の集合の族の上でその生起確率を定義する場合、一般的には**筒集合** (cylinder set) から生成される完全加法族³を通して確率を定義する (付録 B.1)。

定義 3.10 (DTMC 上のパスの筒集合) $\mathcal{D} = \langle S, \mathbf{P}, P_{init}, AP, \Lambda \rangle$ を DTMC とする。このとき、プレフィックスが $s_0 \dots s_n$ であるようなパスの集合 $\text{Paths}^{\mathcal{D}} \cap ((s_0 \dots s_n)S^\omega)$ を $s_0 \dots s_n$ の筒集合といい、 $\text{Cyl}^{\mathcal{D}}(s_0 \dots s_n)$ と書く。 ■

このとき、 $\text{Cyl}^{\mathcal{D}}(s_0 \dots s_n)$ に含まれるパスの生起確率は容易に定義することができる。そして、この筒集合を含む最小の完全加法族からなる可側空間上の確率測度を以下のように定義できる。

定義 3.11 (DTMC のパス集合上の確率測度) $\mathcal{D} = \langle S, \mathbf{P}, P_{init}, AP, \Lambda \rangle$ を DTMC, $\mathcal{F}^{\mathcal{D}}$ を $\text{Paths}^{\mathcal{D}}$ 上のすべての筒集合を含む最小の完全加法族とする。このとき、可側空間 $\langle \text{Paths}^{\mathcal{D}}, \mathcal{F}^{\mathcal{D}} \rangle$ 上の確率測度 $\text{Pr}^{\mathcal{D}} : \mathcal{F}^{\mathcal{D}} \rightarrow [0, 1]$ は以下のように定義できる。

$$\text{Pr}^{\mathcal{D}}(\text{Cyl}^{\mathcal{D}}(\epsilon)) = 1, \quad (3.3)$$

$$\text{Pr}^{\mathcal{D}}(\text{Cyl}^{\mathcal{D}}(s_0 \dots s_n)) = P_{init}(s_0) \cdot \prod_{i=0}^{n-1} P(s_i, s_{i+1}) \quad (3.4)$$

$\mathcal{F}^{\mathcal{D}}$ が完全加法族であるため、 $\text{Paths}^{\mathcal{D}}$ 上のそれぞれの筒集合に対して確率測度が定義されれば充分である。

ここで、DTMC $\mathcal{D}$ の初期状態分布 P_{init} を $P'_{init}(s) = 1$ である状態分布 P'_{init} で置き換えた DTMC $\langle S, \mathbf{P}, P'_{init}, AP, l \rangle$ のパス集合上の確率測度を $\text{Pr}_s^{\mathcal{D}}$ と書く。

例 3.12 例 3.5 の状態遷移系 (図 3.1) の遷移関係と初期状態に確率

$$\mathbf{P}(s, s') = \begin{cases} 1 & \text{if } s = s' = s_2 \text{ or } s = s_1 \wedge s' = s_0, \\ 1/2 & \text{if } s = s_0 \text{ and } s' = s_1 \vee s' = s_2, \\ 0 & \text{otherwise,} \end{cases} \quad (3.5)$$

$$P_{init}(s) = \begin{cases} 1 & \text{if } s = s_0, \\ 0 & \text{otherwise.} \end{cases} \quad (3.6)$$

³ σ -代数ともいう。

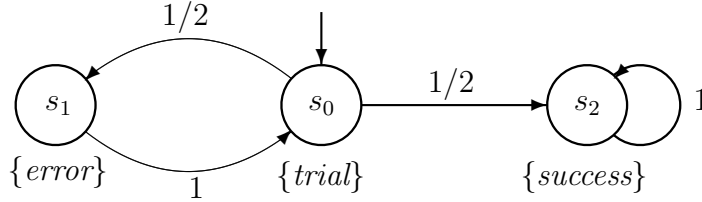


図 3.2: 例 3.12 の確率的試行錯誤系

を導入した DTMC

$$\mathcal{D} = \langle \{s_0, s_1, s_2\}, \mathbf{P}, P_{init}, \{trial, error, success\}, \Lambda \rangle$$

を考える (図 3.2). この DTMC は試行の成功率が $1/2$ であるような仕様錯誤系であり, 最初の試行で成功してそれ以後状態 s_2 にとどまる確率が $1/2$, 次に試行で成功してそれ以後状態 s_2 にとどまる確率が $1/4, \dots$, というふるまいを示す. ■

到達確率

状態遷移系と同様に, DTMC $\mathcal{D} = \langle S, \mathbf{P}, P_{init}, AP, \Lambda \rangle$ のふるまいの性質も到達可能性をもとに解析できる. 定性的な到達可能性については, $\mathcal{D}$ から確率的概念を取り除いた状態遷移系 $\mathcal{T}^{\mathcal{D}}$ の到達可能性を基に解析できる. このとき, DTMC 上の到達可能性や (M/B)SCC は, $\mathcal{T}^{\mathcal{D}}$ 上の到達可能性や (M/B)SCC として定義される. 以後は, DTMC の定性的遷移関係 $\{\langle s, s' \rangle \mid \mathbf{P}(s, s') > 0\}$ を単に $\rightarrow$ と書くこととする.

さらに DTMC では確率的に遷移が選択されるため, 到達可能性を定量的に検証できる. DTMC $\mathcal{D}$ 上では, 各状態 $s \in S$ から状態集合 $S' \subseteq S$ への到達確率 $\text{Reach}_s^{\mathcal{D}}(S')$ は以下のような線形制約式の解として与えられる.

$$\text{Reach}_s^{\mathcal{D}}(S') = \begin{cases} 1 & \text{if } s \in S' \\ 0 & \text{if } s \not\vdash^* S' \\ \sum_{s' \in S} \mathbf{P}(s, s') \cdot \text{Reach}_{s'}^{\mathcal{D}}(S') & \text{otherwise.} \end{cases} \quad (3.7)$$

ここで、各状態 $s \in S$ に対する $\text{Reach}_s^{\mathcal{D}}(S')$ の $|S|$ 個の要素の列を、ベクトルと見なし $\overrightarrow{\text{Reach}}^{\mathcal{D}}(S')$ と書くこととする。なお、この場合に限らず、状態を下付き文字としてもつ関数 f_s は、 $\overrightarrow{f}$ のように表記したときには状態をインデックスとしたベクトルと見なすこととする。

定常状態分布

DTMC のふるまいの長期的な性質は BSCC の特徴を基にしばしば解析できる。なぜならば、有限状態の DTMC のほとんどすべてのパスは BSCC のうちの一つへ到達し、パス上の状態出現頻度比は到達した BSCC に依存した**定常状態分布** (steady-state distribution, stationary distribution) に収束するためである。DTMC $\mathcal{D} = \langle S, \mathbf{P}, P_{\text{init}}, AP, \Lambda \rangle$ 上の BSCC $\langle S', \rightarrow' \rangle \subseteq \langle S, \rightarrow \rangle$ に対して、 $\langle S', \rightarrow' \rangle$ に到達したほとんどすべてのパス上では、各状態 $s \in S'$ の出現頻度 $\text{Freq}_s^{\mathcal{D}}(S')$ が以下のような線形制約式の解に収束する。これは $\langle S', \rightarrow' \rangle$ 上の定常状態分布と呼ばれる。

$$\sum_{s \in S'} \text{Freq}_s^{\mathcal{D}}(S') = 1, \quad (3.8)$$

$$\text{Freq}_s^{\mathcal{D}}(S') = \sum_{s' \in S'} \text{Freq}_{s'}^{\mathcal{D}}(S') \cdot \mathbf{P}(s', s) \quad \text{for each } s \in S'. \quad (3.9)$$

3.1.3 連続時間 Markov 連鎖 (CTMC)

窓口処理などの現実的なシステムは、サービス要求発生間隔やタスク処理時間が連続的かつランダムなものとして捉えられることが多い。そのため、そうしたシステムはしばしば Markov 性をもつ連続時間確率過程、**連続時間 Markov 連鎖** (continuous-time MC, CTMC) としてモデル化される。CTMC のふるまいは DTMC とは異なり、それぞれの状態滞在時間 t は離散的なものではなく、状態ごとに設定された指数分布 Exp_{λ} (exponential distribution, 付録 B.3 節, 定義 B.7) に従う。

$$\text{Exp}_{\lambda}(t) = \lambda \cdot \exp(-\lambda \cdot t). \quad (3.10)$$

ただし、 $\lambda > 0$ は設定されたパラメータである。

形式的には、以下のように定義できる。

定義 3.13 (連続時間 Markov 連鎖) CTMC $\mathcal{C}$ は五つ組 $\langle S, \mathbf{R}, P_{init}, AP, \Lambda \rangle$ である。ここで、

- S は可算の状態集合,
- $\mathbf{R} : S^2 \rightarrow \mathbb{R}$ は以下を満たす状態でインデックスされた遷移速度行列⁴ (transition rate matrix),
 - 任意の状態 $s \in S$ と $s' \neq s$ について, $\mathbf{R}(s, s') \geq 0$,
 - 任意の状態 $s \in S$ について, $\mathbf{R}(s, s) = -\sum_{s' \in S \setminus \{s\}} \mathbf{R}(s, s')$.
- $P_{init} : S \rightarrow [0, 1]$ は DTMC と同様の初期状態分布,
- AP は原子命題の集合,
- $l : S \rightarrow 2^{AP}$ はラベリング関数である. ■

DTMC とは異なり, $R(s, s')$ ($s \neq s'$) は状態 s から状態 s' への遷移速度, すなわち, s' へ遷移するまで s に滞在している時間 t が従う指数分布 $\mathcal{Exp}_\lambda$ のパラメータ λ を表す。そのため, 状態 s から別の状態へ遷移するまでの時間は, パラメータ $-\mathbf{R}(s, s)$ の指数分布に従う。この $-\mathbf{R}(s, s)$ を退出速度 (exit rate) という。ここで, $\mathbf{R}(s, s) = 0$ であるときには, s は他のどの状態へも遷移しない状態であり, 吸収状態 (absorbing state) という。状態 s が吸収状態でないとき, 次に状態 s' へ遷移する確率はそれぞれの状態への遷移速度の比, すなわち, $-\mathbf{R}(s, s')/\mathbf{R}(s, s)$ で与えられる。

CTMC のふるまいでは, それぞれの遷移が生起するタイミングが指数分布に従ったランダムな実数値をとる。そのため, その時間的性質は単なる状態列によってではなく, 各状態の滞在時間列によっても特徴付けられる。形式的には状態とその状態滞在時間の対の列を**時間パス** (timed path) といい, 以下のように定義される。

定義 3.14 (時間パス) $\mathcal{C} = \langle S, \mathbf{R}, P_{init}, AP, \Lambda \rangle$ を CTMC とする。このとき, 以下を満たす状態-時間対の無限列 $\langle s_0, t_0 \rangle \langle s_1, t_1 \rangle \dots \in (S \times \mathbb{R}_{>0})^\omega$ または有限列 $\langle s_0, t_0 \rangle \langle s_1, t_1 \rangle \dots \langle s_n, \infty \rangle \in (S \times \mathbb{R}_{>0})^*(S \times \infty)$ を $\mathcal{C}$ 上の時間パスという。

- $s_0 \in S_{init}$.
- 無限列であるとき, 任意の $i \in \mathbb{N}$ について $\mathbf{R}(s_i, s_{i+1}) > 0$.

⁴無限小生成行列 (infinitesimal generator matrix) ともいう。

- 有限列であるとき, 任意の $i \in \mathbb{N}_{<n}$ について $\mathbf{R}(s_i, s_{i+1}) > 0$.

また, $\mathcal{C}$ 上の時間パスの集合を $\text{Paths}^{\mathcal{C}}$ と書く. ■

滞在時間が無限であるときには, それ以後常にその状態に滞在し続けることを表す. また, 時間パス $\rho = \langle s_0, t_0 \rangle \langle s_1, t_1 \rangle \dots$, あるいは, $\langle s_0, t_0 \rangle \dots \langle s_n, \infty \rangle$ に対して, i 番目の状態 s_i を $\text{state}(\rho, i)$, i 番目の状態の滞在時間 t_i を $\text{time}(\rho, i)$ で表す. さらに, 時刻 t での状態, すなわち, $i = \min\{i' \mid \sum_{j=0}^{i'} t_j > t\}$ 番目の状態 $\text{state}(\rho, i)$ を $\rho @ [t]$, 時刻 t 以後のサフィックス時間パス, すなわち, 前述の i 番目の状態に $t - \sum_{j=0}^{i-1} t_j$ 時間滞在した後の時間パス $\langle s_i, \sum_{j=0}^i t_j - t \rangle \langle s_{i+1}, t_{i+1} \rangle \dots$ を $\rho @ [t \dots]$ と書く.

CTMC では, 遷移だけでなく状態滞在時間も確率的に決定され, CTMC のパスの生起確率は遷移速度行列 $\mathbf{R}$ で特徴付けられる. そのため, 滞在時間も考慮した筒集合から生成される完全加法族を通してパス集合上の生起確率が定義される.

定義 3.15 (CTMC 上の時間パスの筒集合) $\mathcal{C} = \langle S, \mathbf{R}, P_{\text{init}}, AP, \Lambda \rangle$ を CTMC とする. このとき, 状態列のプレフィックスが $s_0, \dots, s_n \in S^*$, かつ, それぞれの状態滞在時間が非負実数 $\mathbb{R}_{\geq 0}$ 上の区間 $I_0, \dots, I_{n-1}$ に含まれるような時間パスの集合を $s_0, I_0, \dots, s_{n-1}, I_{n-1}, s_n$ の筒集合といい, $\text{Cyl}^{\mathcal{C}}(s_0, I_0, \dots, s_{n-1}, I_{n-1}, s_n)$ と書く. つまり,

$$\begin{aligned} \text{Cyl}^{\mathcal{C}}(s_0, I_0, \dots, s_{n-1}, I_{n-1}, s_n) = \{ & \rho \in \text{Paths}^{\mathcal{C}} \mid \forall i \in \mathbb{N}_{\leq n}. (\text{state}(\rho, i) = s_i \\ & \text{and } \forall i \in \mathbb{N}_{<n} \text{time}(\rho, i) \in I_i) \} \end{aligned} \quad (3.11)$$
■

このとき, DTMC の場合と同様に, 筒集合 $\text{Cyl}^{\mathcal{C}}(s_0, I_0, \dots, s_{n-1}, I_{n-1}, s_n)$ に含まれるパスの生起確率は容易に定義することができる. そして, この筒集合を含む最小の完全加法族からなる可側空間上の確率測度を以下のように定義できる.

定義 3.16 (CTMC のパス集合上の確率測度) $\mathcal{C} = \langle S, \mathbf{R}, P_{\text{init}}, AP, \Lambda \rangle$ を CTMC, $\mathcal{F}^{\mathcal{C}}$ を $\text{Paths}^{\mathcal{C}}$ 上のすべての筒集合を含む最小の完全加法族とする. このとき, 可側空間 $\langle \text{Paths}^{\mathcal{C}}, \mathcal{F}^{\mathcal{C}} \rangle$ 上の確率測度 $\text{Pr}^{\mathcal{C}} : \mathcal{F}^{\mathcal{C}} \rightarrow [0, 1]$ は以下のように定義できる.

$$\begin{aligned} \text{Pr}^{\mathcal{C}}(\text{Cyl}^{\mathcal{C}}(\epsilon)) &= 1, \\ \text{Pr}^{\mathcal{C}}(\text{Cyl}^{\mathcal{C}}(s_0, I_0, \dots, s_{n-1}, I_{n-1}, s_n)) &= \end{aligned} \quad (3.12)$$

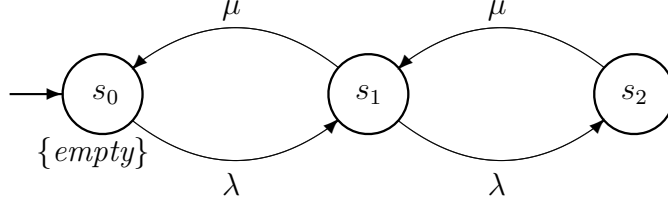


図 3.3: 待ち行列 CTMC の例

$$P_{init}(s_0) \cdot \prod_{i=0}^{n-1} \frac{\mathbf{R}(s_i, s_{i+1})}{-\mathbf{R}(s_i, s_i)} \cdot (\exp(\mathbf{R}(s_i, s_i) \cdot \inf I_i) - \exp(\mathbf{R}(s_i, s_i) \cdot \sup I_i)) \quad (3.13)$$

DTMC の場合と同様に，CTMC $\mathcal{C}$ の初期状態分布 P_{init} を $P'_{init}(s) = 1$ である関数 P'_{init} でそれぞれ置き換えた CTMC の時間パスの集合を $\text{Paths}_s^{\mathcal{C}}$ ，確率測度を $\text{Pr}_s^{\mathcal{C}}$ と書く．

例 3.17 CTMC でモデル化されるシステムの例としては，待ち行列などがある．ここでは， $\mathcal{C} = \langle \{s_0, s_1, s_2\}, \mathbf{R}, P_{init}, \{\text{empty}\}, \Lambda \rangle$ を考える（図 3.3）．ただし，

$$\mathbf{R} = \begin{pmatrix} -\lambda & \lambda & 0 \\ \mu & -(\mu + \lambda) & \lambda \\ 0 & \mu & -\lambda \end{pmatrix} \quad (3.14)$$

$$\Lambda(s) = \begin{cases} \{\text{empty}\} & \text{if } s = s_0 \\ \emptyset & \text{otherwise.} \end{cases} \quad (3.15)$$

$$P_{init}(s) = \begin{cases} 1 & \text{if } s = s_0 \\ 0 & \text{otherwise.} \end{cases} \quad (3.16)$$

この例は， λ に従う指数分布で要求が発生し， μ に従う指数分布に従う時間で処理するキューサイズが 2 である待ち行列システムと解釈できる．

注意 3.18 時間パス ρ が無限長であってもその総経過時間 $\sum_{i=0}^{\infty} \text{time}(\rho, i)$ が有限であることもありうる．その場合，そのふるまいでは有限時間区間に無限回の遷移が起きることを表しており，そのようなふるまいは**爆発的** (explosive) であるという．

爆発的なパスは現実的ではないため、通常は解析の対象とはならない。本論文でも、CTMC は非爆発的、すなわち、ほとんどすべての時間パスが非爆発的（式 (3.17)）であると仮定する。

$$\Pr^{\mathcal{C}} \left(\left\{ \rho \in \text{Paths}^{\mathcal{C}} \mid \sum_{i=0}^{\infty} \text{time}(\rho, i) < \infty \right\} \right) = 0. \quad (3.17)$$

■

一様化

CTMC は一般に状態によって異なる退出速度 $-\mathbf{R}(s, s)$ を持つため確率測度の計算が煩雑になる。そのため、状態集合が有限であるときには、一般には**一様化**（uniformisation）と呼ばれるテクニックが用いられる。

定義 3.19 (一様化 DTMC) $\mathcal{C} = \langle S, \mathbf{R}, P_{\text{init}}, AP, \Lambda \rangle$ を CTMC とする。このとき、 $\lambda \geq \sup\{-\mathbf{R}(s, s) \mid s \in S\}$ であるパラメータ λ に対して、遷移確率行列

$$\mathbf{P} = \mathbf{I} + \frac{1}{\lambda} \cdot \mathbf{R} \quad (3.18)$$

をもつ DTMC $\langle S, \mathbf{P}, P_{\text{init}}, AP, \Lambda \rangle$ を λ で一様化された DTMC といい、 $\text{univ}_{\lambda}(\mathcal{C})$ と書く。ただし、 $\mathbf{I}$ は単位行列とする。 ■

直観的には、一様化 DTMC はそれぞれの状態 s の遷移のしにくさを自己ループ確率 $\mathbf{P}(s, s) = 1 + \mathbf{R}(s, s)/\lambda$ で反映させた遷移確率行列をもつ DTMC である。この一様化 DTMC のパス上の状態の滞在時間が、一様化パラメータ λ の指数分布に従って遷移が生起すると見なすことで CTMC と等価な振舞いを観測できる。ここで等価な振る舞いとは、一様化 DTMC ベースの時間パスに含まれる同じ状態が連続して出現した時間パス、例えば、

$$\langle s_0, t_0 \rangle \dots \langle s_0, t_i \rangle \langle s_1, t_{i+1} \rangle \dots \langle s_1, t_j \rangle \langle s_2, t_{j+1} \rangle \dots$$

を、連続した同じ状態の滞在時間を統合した時間パス、

$$\langle s_0, \sum_{n=0}^i t_n \rangle \langle s_1, \sum_{n=i+1}^j t_n \rangle \dots$$

と見なしたときに等価という意味である。

この一様化 DTMC $\text{univ}_\lambda(\mathcal{C}) = \langle S, \mathbf{P}, P_{init}, AP, \Lambda \rangle$ を用いることで、 t 時間後の過渡確率（現在状態 s にいたときに t 時間後に状態 s' にいる確率） $\Pi_t^{\mathcal{C}}(s, s')$ を効率的に計算できる。

$$\Pi_t^{\mathcal{C}}(s, s') = \sum_{k=0}^{\infty} \mathfrak{Pois}_{\lambda \cdot t}(k) \cdot \mathbf{P}^k \quad (3.19)$$

$$\mathfrak{Pois}_{\lambda \cdot t}(k) = \frac{(\lambda \cdot t)^k \cdot e^{-\lambda \cdot t}}{k!} \quad (3.20)$$

ここで、 $\mathfrak{Pois}_{\lambda \cdot t}(k)$ は、単位時間あたり λ 回ランダムに生起するイベントが t 時間以内にちょうど k 回発生する確率であり、その分布を **Poisson 分布** (Poisson distribution, 付録 B.3 節の定義 B.8) と呼ぶ。

到達確率と定常状態分布

CTMC $\mathcal{C}$ 上の到達可能性や SCC はその一様化 DTMC $\text{univ}_\lambda(\mathcal{C})$ 上の到達可能性や SCC として定義されるまた、 $\mathcal{C}$ 上の（時間非有界）到達確率や定常状態分布は、 $\text{univ}_\lambda(\mathcal{C})$ 上の（時間非有界）到達確率や定常状態分布として計算できる。

3.1.4 利得付き状態遷移系

本節では、**利得付き状態遷移系**（以下、単に利得系と呼ぶ）とそれに関連した概念の定義を与える。利得系は、システムの動作におけるコストなどをモデル化できる重み付き状態遷移系である。そしてそのふるまいの性質はパス上の語のみではなく、状態や遷移に関連付けられた利得（コスト）の列として特徴付けられる。

形式的には、以下のように定義される。

定義 3.20 (利得系) d -次元利得系 $\mathcal{P}$ は六つ組 $\langle S, \rightarrow, S_{init}, AP, \Lambda, w \rangle$ あるいは対 $\langle \langle S, \rightarrow, S_{init}, AP, \Lambda \rangle, w \rangle$ である。ここで、

- $\langle S, \rightarrow, S_{init}, AP, \Lambda \rangle$ は状態遷移系、
- $w : S \rightarrow \mathbb{R}^d$ はそれぞれの状態に d -次元実ベクトルを関連付ける重み関数である。 ■

利得系のパスやパス上の語については、状態遷移系と同様に定義される。

新たに導入された利得に関する時間的性質は、重み関数によって与えられるパス上の重み $\mathbf{x} \in \mathbb{R}^d$ の列によって特徴付けられる。そのような実ベクトル列を**トレース** (trace) と呼ぶ。

定義 3.21 (トレース) d -次元実ベクトルの無限列 $\mathbf{x}_0 \mathbf{x}_1 \dots \in \mathbb{R}^d$ を (d -次元) **トレース** と呼ぶ。 ■

そして、パス ρ 上の利得の列を ρ 上の**利得トレース** (payoff trace) , また、その時点までの平均利得の列を ρ 上の**平均利得トレース** (mean-payoff trace) と呼ぶ。

定義 3.22 (パス上の利得トレースと平均利得トレース) $\mathcal{P}$ を利得系 $\langle S, \rightarrow, S_{init}, AP, \Lambda, w \rangle$ とする。このとき、 $\mathcal{P}$ のパス $\rho = s_0 s_1 \dots \in \text{Paths}^{\mathcal{P}}$ に対して、以下のように定義されるトレース $\text{payoff}_w(\rho) \in (\mathbb{R}^d)^\omega$ を ρ 上の利得トレースという。

$$\text{payoff}_w(\rho)[i] = w(s_i) \quad \text{for each } i \in \mathbb{N}. \quad (3.21)$$

また、以下のように定義されるトレース $\text{mp}_w(\rho) \in (\mathbb{R}^d)^\omega$ を ρ 上の平均利得トレースという。

$$\text{mp}_w(\rho)[i] = \frac{1}{i+1} \sum_{j=0}^i w(s_j) \quad \text{for each } i \in \mathbb{N}. \quad (3.22)$$

■

また、ベクトル $\mathbf{x} = \langle x_1, \dots, x_d \rangle$ の i 次元射影 x_i を $\pi_i(\mathbf{x})$ で表す。そして、重み関数 w の i 番目の次元の重み関数を $w[i]$ と書く。すなわち、 $w[i](s) = \pi_i(w(s))$ とする。

集積点と上極限/下極限

パス ρ の平均利得に関する定量的な特徴は、様々な観点からとらえることができる。[6] では、その平均利得トレース $\text{mp}_w(\rho)$ の**集積点**⁵ (accumulation point) の集合 $\text{Acc}(\text{mp}_w(\rho))$ を通して捉えている。直観的に言えば、集積点とは、その任意の近傍にトレースの要素が無限個含まれる点である。数学的には以下のように定義される。

⁵極限点 (limit point) ともいう。

定義 3.23 (集積点) 以下を満たす点 $\mathbf{x} \in \mathbb{R}^d$ をトレース $\tau = \mathbf{x}_0\mathbf{x}_1 \dots \in (\mathbb{R}^d)^\omega$ の集積点という.

$$\forall \epsilon \in \mathbb{R}_{>0}, i \in \mathbb{N}. \exists j > i. (|\mathbf{x} - \mathbf{x}_j| < \epsilon). \quad (3.23)$$

また, τ のすべての集積点の集合を $\mathfrak{Acc}(\tau)$ と書く. ■

重み関数 w が与える重みが有界であれば, パスの平均利得トレースの集積点集合は, 非空, 有界かつ連結的である [6]. なぜならば, どんなパス ρ であっても, 平均利得トレースのどの次元 k についても, i 番目と $i+1$ 番目の要素の差 $d_k[i]$ は $i \rightarrow \infty$ で明らかに 0 に収束するためである.

$$\begin{aligned} d_k[i] &= \text{mp}_{w[k]}(\rho)[i+1] - \text{mp}_{w[k]}(\rho)[i] \\ &= \frac{w[k](\rho[i+1]) - \text{mp}_{w[k]}(\rho)[i]}{i+2} \\ &\leq \frac{2 \cdot \max\{|w[k](s)| \mid s \in S\}}{i+2} \end{aligned} \quad (3.24)$$

集積点の概念は極限に関する概念を一般化したものであり, トレース τ に関する集積点集合の i 次元成分射影区間の最大値 $\max \pi_i(\mathfrak{Acc}(\tau))$ (最小値 $\min \pi_i(\mathfrak{Acc}(\tau))$) は τ の i 次元成分射影列 $(\pi_i(\tau[0]))(\pi_i(\tau[1])) \dots \in \mathbb{R}$ の上極限 $\limsup_{n \rightarrow \infty} (\pi_i(\tau[n]))$ (下極限 $\liminf_{n \rightarrow \infty} (\pi_i(\tau[n]))$) と一致する. そのため, 平均利得トレースの i 次元成分の上極限/下極限に関して以下の等式が成り立つ.

$$\limsup_{n \rightarrow \infty} \text{mp}_{w[i]}(\rho)[n] = \max \pi_i(\mathfrak{Acc}(\text{mp}_w(\rho))), \quad (3.25)$$

$$\liminf_{n \rightarrow \infty} \text{mp}_{w[i]}(\rho)[n] = \min \pi_i(\mathfrak{Acc}(\text{mp}_w(\rho))). \quad (3.26)$$

例 3.24 利得系 $\langle \{s_0, s_1, s_2\}, \rightarrow, \{s_0\}, \{off, on, idle, busy\}, \Lambda, w \rangle$ を考える (図3.4). ただし,

$$\begin{aligned} \rightarrow &= \{ \langle s_0, s_0 \rangle, \langle s_1, s_1 \rangle, \langle s_2, s_2 \rangle, \\ &\quad \langle s_0, s_1 \rangle, \langle s_1, s_2 \rangle, \langle s_1, s_2 \rangle, \langle s_2, s_1 \rangle \}, \end{aligned} \quad (3.27)$$

$$\Lambda(s) = \begin{cases} \{off\} & \text{if } s = s_0, \\ \{on, idle\} & \text{if } s = s_1, \\ \{on, busy\} & \text{otherwise,} \end{cases} \quad (3.28)$$

$$w(s) = \begin{cases} 0 & \text{if } s = s_0, \\ 1 & \text{if } s = s_1, \\ 6 & \text{otherwise.} \end{cases} \quad (3.29)$$

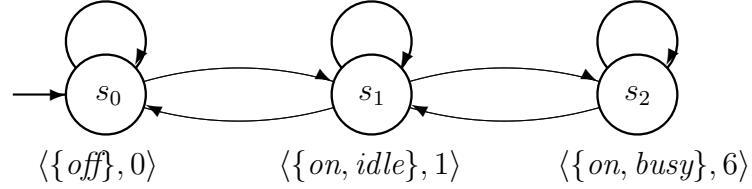


図 3.4: 例 3.24 の利得系.

この利得系では, s_0 は *off* 状態でありコストは 0, s_1 は *on* かつ *idle* 状態でありコストは 1, s_2 は *on* かつ *busy* 状態でありコストは 6, というシステムを解釈できる. パス $(s_0 s_1 s_2 s_1)^\omega$ の利得トレースは

$$\text{payoff}_w((s_0 s_1 s_2 s_1)^\omega) = (0 \ 1 \ 6 \ 1)^\omega$$

である. また, 平均利得トレースは

$$\text{mp}_w(\rho) = 0 \ \frac{1}{2} \ \frac{7}{3} \ 2 \ \frac{8}{5} \ \frac{3}{2} \ \frac{15}{7} \ 2 \ \dots$$

であり, 2 に収束する. この例に限らず, 有限長のトレース $\tau_1, \tau_2 \in (\mathbb{R}^d)^+$ に対して, $\tau_1 \tau_2^\omega$ の形のトレースの平均利得トレースは一点に収束する. ■

利得付き DTMC

確率的モデル検査でもしばしば利得付き拡張確率的モデルを解析することがあり, 確率的モデル検査器 PRISM[70, 58] では利得に関する期待値の検証等を行うことができる.

本論文では, 以下のような利得付き確率的モデルのみを扱う.

定義 3.25 (利得付き DTMC) d -次元利得付き DTMC $\mathcal{D}$ は六つ組 $\langle S, \mathbf{P}, P_{init}, AP, \Lambda, w \rangle$ あるいは対 $\langle \langle S, \mathbf{P}, P_{init}, AP, \Lambda \rangle, w \rangle$ である. ここで,

- $\langle S, \mathbf{P}, P_{init}, AP, \Lambda \rangle$ は DTMC,
- $w : S \rightarrow \mathbb{R}^d$ はそれぞれの状態に d -次元実ベクトルを関連付ける重み関数である. ■

利得系のパスやパス上の語については状態遷移系, パスの生起確率については DTMC, (平均) 利得トレースについては利得系と同様に定義される.

3.2 時間論理

システムのふるまいが仕様を満たしているか否かや仕様に欠陥がないことを数学的に検証するためには、曖昧性がないように仕様を規定されていなければならない。そのため、形式的に仕様を記述するための言語として、時間的性質を記述できる様相論理である時間論理がしばしば用いられている。時間論理では、命題論理で用いる通常の演算子に加えて、時間的性質を記述できる**時間演算子** (temporal operator) やふるまい生起の可能性を記述できる**パス限量子** (path quantifier) をもつ。

本節では、最も広く用いられている時間論理の一つである**線形時相論理**[69] (linear temporal logic, LTL) を紹介する。また最も広く用いられているもう一つの時間論理である**計算木論理**[38] (computation tree logic, CTL), CTLに確率の概念を導入した**確率計算木論理**[51] (probabilistic CTL, PCTL), さらに実時間性も導入した**連続確率論理**[11, 12] (continuous stochastic logic, CSL) を簡単に紹介する。

3.2.1 線形時相論理 LTL

線形時相論理 LTL[69] は、一つの時間系列に関する性質を記述できる線形時間論理の一種である。通常の命題論理結合子に加え、時間演算子として「next 演算子 X (あるいは $\bigcirc$)」と「(強い) until 演算子 U 」をもち、ふるまいの時間的な性質を記述できる。

定義 3.26 (LTL の構文) AP を原子命題の集合とする。このとき、LTL の式 φ は以下のように帰納的に定義される。

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid X\varphi \mid \varphi U\varphi.$$

ただし、 $p \in AP$ とする。 ■

直観的には、 $X\varphi$ は「次 (neXt) のステップで φ が成り立つ」、 $\varphi_1 U \varphi_2$ は「いつか φ_2 が成り立ち、それまでの間 (Until) φ_1 が成り立っている」ことを表す。

時間論理では、命題論理で用いる論理記号や結合子と通常の時間演算子 $\Diamond$ (あるいは F) と $\Box$ (あるいは G) や「弱い (Weak) until 演算子 W 」「release

演算子 R」も略記としてしばしば用いられている.

$$\top \equiv \varphi \vee \neg\varphi, \quad (3.30)$$

$$\perp \equiv \neg\top, \quad (3.31)$$

$$\varphi_1 \wedge \varphi_2 \equiv \neg(\neg\varphi_1 \vee \neg\varphi_2), \quad (3.32)$$

$$\varphi_1 \rightarrow \varphi_2 \equiv \neg\varphi_1 \vee \varphi_2, \quad (3.33)$$

$$\varphi_1 \leftrightarrow \varphi_2 \equiv (\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_1), \quad (3.34)$$

$$\Diamond\varphi \equiv \top U\varphi, \quad (3.35)$$

$$\Box\varphi \equiv \neg\Box\neg\varphi, \quad (3.36)$$

$$\varphi_1 W\varphi_2 \equiv (\varphi_1 U\varphi_2) \vee \Box\varphi_1, \quad (3.37)$$

$$\varphi_1 R\varphi_2 \equiv \neg(\neg\varphi_1 U\neg\varphi_2). \quad (3.38)$$

$\Diamond\varphi$ は「いつか φ が成り立つ」ことを表す. $\Box$ は $\Diamond$ の双対であり, $\Box\varphi$ は「常に φ が成り立つ」ことを表す. また $\varphi_1 W\varphi_2$ は, 「 φ_2 を満たすまで φ_1 を満たし続けているが, 必ずしもいつか φ_2 を満たす必要はない (そのときには常に φ_1 を満たす)」という性質を表し, そして R は U の双対であり, $\varphi_1 R\varphi_2$ は「 φ_1 が φ_2 を解放する (Release)」ことを表し, $\varphi_2 W(\varphi_1 \wedge \varphi_2)$ と等価である.

例 3.27 ここで, LTL 式で記述できる時間的性質をいくつか紹介する.

- $\Diamond\Box\varphi$: 「常に φ が成り立つこと ($= \Box\varphi$)」がいつか成り立つ, すなわち, ある時点以後常に φ が成り立つ.
- $\Box\Diamond\varphi$: 「いつか φ が成り立つこと ($= \Diamond\varphi$)」が常に成り立つ.
 - 「無限にしばしば (infinitely often) φ が成り立つ」と表現される.
- $\Box(\varphi \rightarrow \Diamond\psi)$: 常に, φ が成り立つときにはいつか ψ が成り立つ.
 - 要求 φ に対する応答 ψ の反応性に関する性質としてしばしば用いられている.
- $\Box(\varphi \rightarrow \psi_1 U\psi_2)$: 常に, φ が成り立つときにはいつか ψ_2 が成り立ち, それまでの間 ψ_1 が成り立ち続けている.
 - 上の性質の似ているが, 要求 φ に対して応答 ψ_2 するまでの間, ψ_1 が成り立ち続けなければならないことを表す. ■

LTL の意味論は以下のように与えられる.

定義 3.28 (LTL の意味論) AP を原子命題の集合とする. このとき, 無限長語 $\sigma = a_0a_1 \dots \in (2^{AP})^\omega$ と AP 上の LTL 式 φ に対して, 充足関係 $\models$ は以下のように帰納的に定義される.

$$\begin{aligned} \sigma, i \models p &\Leftrightarrow p \in a_i \text{ for } p \in AP, \\ \sigma, i \models \neg\varphi &\Leftrightarrow \sigma, i \not\models \varphi, \\ \sigma, i \models \varphi_1 \vee \varphi_2 &\Leftrightarrow \sigma, i \models \varphi_1 \text{ or } \sigma, i \models \varphi_2, \\ \sigma, i \models X\varphi &\Leftrightarrow \sigma, i+1 \models \varphi, \\ \sigma, i \models \varphi_1 U \varphi_2 &\Leftrightarrow \exists j \geq i. (\sigma, j \models \varphi_2 \text{ and } \forall k \in [i, j). \sigma, k \models \varphi_1). \end{aligned}$$

ただし, $i, j, k \in \mathbb{N}$ である. ■

以後は, $\sigma, 0 \models \varphi$ を単に $\sigma \models \varphi$ と書く.

充足可能性

LTL 式 φ に対して, $\sigma \models \varphi$ であるような語 σ が存在するとき, φ は**充足可能**(無矛盾) であるという. LTL 式 φ を満たす語の集合 $\{\sigma \in (2^{AP})^\omega \mid \sigma \models \varphi\}$, つまり, φ が規定する言語を $L(\varphi)$, LTL で記述できる言語のクラス $\{L(\varphi) \mid \varphi \in LTL\}$ を $\mathcal{L}(LTL)$ と書く.

ここで, LTL 式の充足可能性判定問題に関して以下が成り立つ.

定理 3.29 ([76]) LTL 充足可能性問題は PSPACE-完全である. ■

モデル検査

LTL モデル検査では, 状態遷移系モデル $\mathcal{T} = \langle S, \rightarrow, S_{init}, AP, \Lambda \rangle$ のすべてのふるまい上の語が LTL 仕様 φ を満たすか否かが網羅的に検証される. ここで, $\mathcal{T}$ のすべてのふるまいが φ を満たすことを $\mathcal{T} \models \varphi$ と書く.

$$\mathcal{T} \models \varphi \stackrel{\text{def}}{=} \forall \rho \in \text{Paths}^{\mathcal{T}}. (\text{word}_\Lambda(\rho) \models \varphi). \quad (3.39)$$

このとき, LTL モデル検査に関して以下が成り立つ.

定理 3.30 ([60]) 状態遷移系に対する LTL モデル検査問題は PSPACE-完全である. ■

また、DTMC に対する LTL モデル検査（確率的 LTL モデル検査）手法も提案されている。この場合、DTMC モデル $\mathcal{D} = \langle S, \mathbf{P}, P_{init}, AP, \Lambda \rangle$ のほとんどすべてのふるまい上の語が LTL 仕様 φ を満たすか否かが検証される。ここで、 $\mathcal{D}$ 上での φ の充足確率 $\Pr^{\mathcal{D}}(\{\rho \in \text{Paths}^{\mathcal{D}} \mid \text{word}_{\Lambda}(\rho) \models \varphi\})$ が 1 である、すなわち、 $\mathcal{D}$ のほとんどすべてのふるまいが φ を満たすことを $\mathcal{D} \models \varphi$ と書く。

$$\mathcal{D} \models \varphi \stackrel{\text{def}}{=} \Pr^{\mathcal{D}}(\{\rho \in \text{Paths}^{\mathcal{D}} \mid \text{word}_{\Lambda}(\rho) \models \varphi\}) = 1. \quad (3.40)$$

確率的 LTL モデル検査に関しては以下が成り立つ。

定理 3.31 ([40, 41]) DTMC に対する確率的 LTL モデル検査（LTL 充足確率問題）は、EXPTIEM で決定（計算）可能である。

3.2.2 確率時間論理

LTL が一つの時間系列の性質について複数の時間演算子を用いて記述できたのに対して、CTL では時間演算子の用法に制限があるものの「全称パス限量子 A」と「存在パス限量子 E」を用いることで状態のもつ分岐の可能性を定性的に記述できた。CTL に確率的概念を導入した PCTL や CSL では、「確率パス限量子 P」によって、分岐の可能性に関する定量的な性質を記述できる。また、時間的性質の起こりやすさを定量的に表現できるため、 $\varphi_1 U \varphi_2$ に関して φ_2 が成り立つタイミングに関して記述できるようになっている。

本項では、まず CTL について紹介し、その後、その確率的拡張である PCTL と CSL を紹介する。

計算木論理 CTL

計算木論理 CTL[38] は、分岐の可能性の性質を記述できる分岐時間論理の一種である。通常の命題論理結合子と LTL の時間演算子に加えて、存在パス限量子 E と全称パス限量子 A をもつ。これによって、LTL のような一つの時間系列に関する性質を記述できる線形時間論理とは異なり、状態のもつ分岐の可能性の性質を記述できる。ただし、CTL では時間演算子の直前にパス限量子を置かなければならないという制限がある。

形式的には、CTL の構文は以下のように定義される。

定義 3.32 (CTL の構文) AP を原子命題の集合とする。このとき、CTL の式 φ は以下のように帰納的に定義される。

$$\begin{aligned}\varphi &::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid E[\psi] \mid A[\psi], \\ \psi &::= X\varphi \mid \varphi U \varphi.\end{aligned}$$

ただし、 $p \in AP$ とする。 ■

ここで、 φ の型の式を**状態式** (state formula)、 ψ の型の式を**パス式** (path formula) という。状態式は状態がもつ可能性に関する性質を、パス式はパス上に実際に現れる時間的性質を表現している。

直観的には、 $E[\psi]$ は「その状態を始点として、パス式 ψ を満たすふるまいが存在する」、 $A[\psi]$ は「その状態を始点としたすべてのふるまいがパス式 ψ を満たす」ことを表す。

このとき、LTL と同様に以下のような略記などがしばしば用いられている。

$$E[\Diamond\varphi] \equiv E[\text{TRUE} \vee \varphi], \quad (3.41)$$

$$A[\Diamond\varphi] \equiv A[\text{TRUE} \vee \varphi], \quad (3.42)$$

$$E[\Box\varphi] \equiv \neg A[\Diamond\neg\varphi], \quad (3.43)$$

$$A[\Box\varphi] \equiv \neg E[\Diamond\neg\varphi]. \quad (3.44)$$

次に CTL の意味論を与える。LTL のような線形時間論理では、語（あるいはラベル付けされたパス）の性質として意味が与えられていた。一方で CTL などの分岐時間論理の意味は、状態遷移系上の状態の性質として意味が与えられる。

定義 3.33 (CTL の意味論) 状態遷移系 $\mathcal{T} = \langle S, \rightarrow, S_{init}, AP, \Lambda \rangle$ 、状態 $s \in S$ 、各状態を始点としたパス $\rho \in \bigcup_{s \in S} \text{Paths}_s^{\mathcal{T}}$ 、 AP 上の CTL 状態式 φ 、CTL パス式 ψ に対して、充足関係 $\models$ は以下のように帰納的に定義される。

$$\begin{aligned}\mathcal{T}, s &\models p \Leftrightarrow p \in \Lambda(s), \\ \mathcal{T}, s &\models \neg\varphi \Leftrightarrow \mathcal{T}, s \not\models \varphi, \\ \mathcal{T}, s &\models \varphi_1 \vee \varphi_2 \Leftrightarrow \mathcal{T}, s \models \varphi_1 \text{ or } \mathcal{T}, s \models \varphi_2, \\ \mathcal{T}, s &\models E[\psi] \Leftrightarrow \exists \rho \in \text{Paths}_s^{\mathcal{T}}. (\mathcal{T}, \rho \models \psi) \\ \mathcal{T}, s &\models A[\psi] \Leftrightarrow \forall \rho \in \text{Paths}_s^{\mathcal{T}}. (\mathcal{T}, \rho \models \psi) \\ \mathcal{T}, \rho &\models X\varphi \Leftrightarrow \mathcal{T}, \rho[1] \models \varphi \\ \mathcal{T}, \rho &\models \varphi_1 U \varphi_2 \Leftrightarrow \exists i \in \mathbb{N}. (\mathcal{T}, \rho[i] \models \varphi_2 \text{ and } \forall j < i. (\mathcal{T}, \rho[j] \models \varphi_1)), \\ \mathcal{T} &\models \varphi \Leftrightarrow \forall s \in S. (\mathcal{T}, s \models \varphi).\end{aligned}$$

例 3.34 ここで, LTL 式で記述できる時間的性質をいくつか紹介する.

- $E[\Diamond \varphi]$: いつか φ が成り立つパスがある.
- $A[\Box \varphi]$: すべてのパスで常に φ が成り立つ.
- $A[\Box E[\Diamond \varphi]]$: すべてのパス上に “いつか φ が成り立つパスがある” 状態が常に出現している, すなわち, すべてのパス上のすべての状態が φ が成り立つ状態へ到達する可能性をもち続けている.

注意 3.35 (CTL*) 時間演算子の直前にパス限量子を必ずしも置かなくてもよい時間論理を CTL* という. 状態遷移系モデルと LTL 式の充足関係式 (3.39) より, LTL 式 ψ は CTL* 上では $A[\psi]$ と表現される. また, CTL と LTL は CTL* の真部分集合であり, 一方では表現できるが他方では表現できない性質が存在する.

確率計算木論理 PCTL

確率計算木論理 PCTL[51] は, CTL に確率の概念を導入した離散時間確率時間論理である. 確率が導入されたことにより, 「あるパスで ψ を満たす ($E[\psi]$)」や「すべてのパスで ψ を満たす ($A[\psi]$)」のような定性的な性質ではなく, 「 ψ を満たすパスの “量” に関する制約」を表現することが可能となっている. PCTL では確率パス限量子 P を用いてそのような確率に関する定量的性質を記述することができる. また, ふるまいの確率に関する定量的な評価において, 単なる「いつか φ を満たすパスの生起確率」ではなく, 「時刻 t までに φ を満たすパスの生起確率」のような時間 (ステップ) 制限付きの性質がしばしば注目される. そのため, PCTL では until 演算子に時間区間制約を付加することができるように拡張されている.

形式的には以下のように定義される.

定義 3.36 (PCTL の構文) AP を原子命題の集合とする. このとき, AP 上の PCTL の式 φ は以下のように帰納的に定義される.

$$\begin{aligned}\varphi &::= p \mid \neg \varphi \mid \varphi \vee \varphi \mid P_{\sim c}[\psi], \\ \psi &::= X\varphi \mid \varphi U^I \varphi.\end{aligned}$$

ただし, $p \in AP$, $c \in [0, 1]$, $\sim \in \{<, >, \leq, \geq\}$, $I \subseteq \mathbb{N}$ は自然数上の区間とする.

直観的には、 $P_{\sim c}[\psi]$ は「その状態が線形時間的性質 ψ を満たすパスの生起確率が閾値条件 $\sim c$ を満たす」ことを表す。また、 $\varphi_1 U^I \varphi_2$ は「時間区間 I に含まれるある時間で φ_2 が成り立ち、それまでの間は φ_1 が成り立っている」ことを表す。

ここで、パス式 $\varphi_1 U^{[0, \infty)} \varphi_2$ については区間 $[0, \infty)$ をしばしば省略する。また、PCTL の構文に限らず、時間区間に関しては以下のような略記を用いる。

$$\sim c \equiv \{t \mid t \sim c\}, \quad (3.45)$$

$$= c \equiv [c, c], \quad (3.46)$$

$$I + c \equiv \{t + c \mid t \in I\} \cap [0, \infty), \quad (3.47)$$

$$[c \pm c'] \equiv [c - c', c + c']. \quad (3.48)$$

パス式に関しては、CTL 同様の以下のような略記も用いる。

$$P_{\sim c}[\Diamond^I \varphi] \equiv P_{\sim c}[\top U^I \varphi] \quad (3.49)$$

$$P_{\sim c}[\Box^I \varphi] \equiv P_{\sim 1-c}[\Diamond^I \neg \varphi] \quad (3.50)$$

本論文ではさらに、時間区間制約を考慮する until 式に関する略記として Δ 演算子を用いる。

$$\Delta^I \psi \equiv \neg \psi U^I \psi. \quad (3.51)$$

直観的には、 $\Delta^I \psi$ は「現時点からみて、時間区間 I 中で初めて ψ が成り立つ (立ち上がる)」ことを表している。

次に PCTL の意味論を与える。CTL の確率的拡張である PCTL に対しては、CTL と同様に、PCTL 式の意味は DTMC 上の状態の性質として与えられる。

定義 3.37 (PCTL の意味論) DTMC $\mathcal{D} = \langle S, \mathbf{P}, P_{init}, AP, \Lambda \rangle$, 状態 $s \in S$, 各状態を始点としたパス $\rho \in \bigcup_{s \in S} \text{Paths}_s^{\mathcal{D}}$, AP 上の PCTL 状態式 φ , PCTL パス式 ψ に対して、充足関係 $\models$ は以下のように帰納的に定義される。

$$\mathcal{D}, s \models p \Leftrightarrow p \in \Lambda(s),$$

$$\mathcal{D}, s \models \neg \varphi \Leftrightarrow \mathcal{D}, s \not\models \varphi,$$

$$\mathcal{D}, s \models \varphi_1 \vee \varphi_2 \Leftrightarrow \mathcal{D}, s \models \varphi_1 \text{ or } \mathcal{D}, s \models \varphi_2,$$

$$\mathcal{D}(s) \models P_{\sim c}[\psi] \Leftrightarrow \Pr_{(s)}^{\mathcal{D}}(\{\rho \in \text{Paths}_{(s)}^{\mathcal{D}} \mid \mathcal{D}, \rho \models \psi\}) \sim c$$

$$\mathcal{D}, \rho \models X\varphi \Leftrightarrow \mathcal{D}, \rho[1] \models \varphi$$

$$\mathcal{D}, \rho \models \varphi_1 U^I \varphi_2 \Leftrightarrow \exists i \in I. (\mathcal{D}, \rho[i] \models \varphi_2 \text{ and } \forall j < i. \mathcal{D}, \rho[j] \models \varphi_1).$$

■
以後は、(状態 s を始点としたときの) $\mathcal{D}$ 上で φ を満たすパスの生起確率を $\text{Prob}_{(s)}^{\mathcal{D}}(\varphi)$ と書く.

$$\text{Prob}_{(s)}^{\mathcal{D}}(\psi) = \Pr_{(s)}^{\mathcal{D}}(\{\rho \in \text{Paths}_{(s)}^{\mathcal{D}} \mid \mathcal{D}, \rho \models \psi\}) \quad (3.52)$$

連続確率論理 CSL

連続確率論理 CSL[11, 12] は, PCTL に実時間性も導入した連続時間確率時間論理である.

CSL の構文は PCTL とほぼ同じであり, 以下のように定義される (注意 3.42).

定義 3.38 (CSL の構文) AP を原子命題の集合とする. このとき, AP 上の CSL の式 φ は以下のように帰納的に定義される.

$$\begin{aligned} \varphi &::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid P_{\sim c}[\psi], \\ \psi &::= X^I\varphi \mid \varphi U^I\varphi. \end{aligned}$$

ただし, $p \in AP$, $c \in [0, 1]$, $\sim \in \{<, >, \leq, \geq\}$, $I \subseteq \mathbb{R}_{\geq 0}$ は非負実数上の区間とする. ■

CSL は PCTL の実時間拡張であるため, 時間演算子の時間制約は非負実数上の区間として与えられる. 構文上で PCTL と異なるもう一つの点は, X 演算子にも時間区間制約がある点である. これは, CTMC では次の状態への遷移するまでの時間が固定されていないためである. 直観的には, $X^I\varphi$ は「次の状態遷移が生起する (現在を基準とした相対) 時刻が時間区間 I に含まれ, かつ, その状態で φ を満たす」ことを表している. なお, CSL に関する各種略記は PCTL と同様のものを用いる.

離散時間確率時間論理である PCTL の意味が DTMC 上で与えられていたのに対して, 連続時間確率時間論理である PCTL の意味は CTMC 上で与えられる.

定義 3.39 (CSL の意味論) CTMC $\mathcal{C} = \langle S, \mathbf{R}, P_{init}, AP, \Lambda \rangle$, 状態 $s \in S$, 各状態を始点としたパス $\rho \in \bigcup_{s \in S} \text{Paths}_s^{\mathcal{C}}$, AP 上の CSL 状態式 φ , CSL パス式

ψ に対して、充足関係 $\models$ は以下のように帰納的に定義される。

$$\begin{aligned}
\mathcal{C}, s \models p &\Leftrightarrow p \in \Lambda(s), \\
\mathcal{C}, s \models \neg\varphi &\Leftrightarrow \mathcal{C}, s \not\models \varphi, \\
\mathcal{C}, s \models \varphi_1 \vee \varphi_2 &\Leftrightarrow \mathcal{C}, s \models \varphi_1 \text{ or } \mathcal{C}, s \models \varphi_2, \\
\mathcal{C}(\cdot, s) \models P_{\sim c}[\psi] &\Leftrightarrow \Pr_{(s)}^{\mathcal{D}}(\{\rho \in \text{Paths}_{(s)}^{\mathcal{C}} \mid \mathcal{C}, \rho \models \psi\}) \sim c \\
\mathcal{C}, \rho \models X^I \varphi &\Leftrightarrow \text{time}(\rho, 0) \in I \text{ and } \mathcal{C}, \rho @ [\text{time}(\rho, 0)] \models \varphi \\
\mathcal{C}, \rho \models \varphi_1 U^I \varphi_2 &\Leftrightarrow \exists t_1 \in I. (\mathcal{C}, \rho @ [t_1] \models \varphi_2 \text{ and } \forall t_2 < t_1. \mathcal{C}, \rho @ [t_2] \models \varphi_1).
\end{aligned}$$

■

以後は、(状態 s を始点としたときの) $\mathcal{C}$ 上で φ を満たすパスの生起確率を $\text{Prob}_{(s)}^{\mathcal{C}}(\varphi)$ と書く。

$$\text{Prob}_{(s)}^{\mathcal{C}}(\psi) = \Pr_{(s)}^{\mathcal{C}}(\{\rho \in \text{Paths}_{(s)}^{\mathcal{C}} \mid \mathcal{C}, \rho \models \psi\}) \quad (3.53)$$

例 3.40 ここで PCTL と CSL で記述できる性質をいくつか紹介する。

- $P_{>0}[\psi]$: ψ を満たすパスが 0 より大きい確率で生起する。
 - この式は CTL 式 $E[\psi]$ とは意味が異なる。なぜならば、例 3.12 の DTMC には常に「表」が出るパスは存在しているが、その生起確率は 0 である。
- $P_{>1/2}[\Diamond^{\leq 5} P_{\leq 1/3}[\psi]]$: ψ を満たすパスの生起確率が $1/3$ 以下であるような状態に、5 ステップ以内に到達する確率が $1/2$ より大きい。
- $P_{>2/3}[\Box P_{>0}[\Diamond \varphi]]$: φ を満たす状態へ到達する確率が 0 でない状態が常に出現し続けるパスの生起確率が $2/3$ より大きい。 ■

注意 3.41 (定時 Δ 演算子) 任意の CTMC $\mathcal{C}$ の任意の状態 s について、次の遷移が丁度ある時刻 $t \in (0, \infty)$ であるようなパスが生起する確率

$$\Pr_s^{\mathcal{C}}\left(\bigcup_{s' \in S} \text{Cyl}_s^{\mathcal{C}}(s, [t, t], s')\right)$$

は 0 である。よって、任意の状態式 φ に対して、 $\mathcal{C}, s \models P_{\leq 0}[X^t \varphi]$ かつ $\mathcal{C}, s \models P_{\leq 0}[\Delta^t \varphi]$ であり、CSL では「 φ が丁度 t 単位時間後に成り立つ確率」に関する性質は記述できない。 ■

注意 3.42 (定常状態分布演算子 S) CSL の一般的な定義 [12] では, 定常状態分布演算子 S ももつ. 直観的には, この式 $S_{\sim c}[\varphi]$ は「 φ を満たす状態の定常状態分布の期待値が閾値条件 $\sim c$ を満たす」ことを表す. 形式的には, 有限状態 CTMC $\mathcal{C}$ 上で以下のように意味付けられる.

$$\mathcal{C}, s \models S_{\sim c}[\varphi] \Leftrightarrow \text{Expect}[X_s] \sim c. \quad (3.54)$$

ここで, $\text{Expect}[X_s]$ は以下のように定義される可積分な確率変数 $X_s : \text{Paths}_s^{\mathcal{C}} \rightarrow [0, 1]$ の期待値である.

$$X_s(\rho) = \lim_{k \rightarrow \infty} \frac{\mathfrak{L}(\{t < k \mid \mathcal{C}, \rho @ [t] \models \varphi\})}{k}. \quad (3.55)$$

ただし, $\mathfrak{L}$ は Lebesgue 測度である.

なお, 有限状態 CTMC 上では, ほとんどすべてのふるまいは何らかの BSCC に到達し, その BSCC に依存した定常状態分布をもつため, 極限值 $X_s(\rho)$ はほとんどすべてのパス ρ に対して定義できる. ■

注意 3.43 (PCTL と CSL の報酬拡張) PCTL と CSL には報酬 (利得) の概念を導入した拡張がいくつか提案されている [13, 10]. それらの拡張では, モデル上で定義されている報酬の与え方のもとの, パス上の累積報酬に関する期待値に関する性質を記述できる. ■

時間制約上界

PCTL/CSL の意味論から, $\mathcal{D}/\mathcal{C}, \rho \models \psi$ の真偽は, パス ρ の時間 $\#(\psi)$ までのプレフィックスのみに依存する. ただし, $\#$ は以下のように定義される関数である.

$$\#(\psi) = \begin{cases} 0 & \text{if } \psi = a \in AP, \\ \#(\psi_1) & \text{if } \psi = \neg\psi_1, \\ \max\{\#(\psi_1), \#(\psi_2)\} & \text{if } \psi = \psi_1 \vee \psi_2, \\ \#(\psi) + \sup I & \text{if } \psi = X^I \psi_1, \\ \max\{\#(\psi_1), \#(\psi_2)\} + \sup I & \text{if } \psi = \psi_1 \cup^I \psi_2. \end{cases} \quad (3.56)$$

ただし, 離散時間上では, 時間区間 I は閉区間であり, X 式にかんしては $I = [1, 1]$ であるものとする.

この $\#(\psi)$ を ψ (及び状態式 $P_{\sim c}[\psi]$) の**時間制約上界**といい, $\#(\psi) = \infty$ のとき ψ は**時間非有界**であるという.

充足可能性

CTL 状態式 φ に対して、 $\mathcal{T} \models \varphi$ であるような状態遷移系 $\mathcal{T}$ が存在するとき、 φ は充足可能であるという。ここで、CTL 式が充足可能であるならば、かつそのときに限り、 $\mathcal{T} \models \varphi$ であるような有限状態の状態遷移系 $\mathcal{T}$ が存在する [38]。この性質を小モデル性 (small model property) という。よって、CTL の充足可能性判定問題は、そのような有限状態の状態遷移系を構成可能であるか否かを判定することにより計算できる。

定理 3.44 ([46]) CTL の充足可能性判定問題は EXPTIME-完全である。 ■

CTL と同様に、PCTL (CSL) 状態式 φ に対して、 $\mathcal{D} \models \varphi$ ($\mathcal{C} \models \varphi$) であるような DTMC $\mathcal{D}$ (CTMC $\mathcal{C}$) が存在するとき、 φ は充足可能であるという。最も単純な確率時間論理である PCTL の充足可能性判定が決定可能であるか否かは未解決問題であるが、PCTL の定性的フラグメント、すなわち、確率パス限量子 $\mathcal{P}_{\sim c}$ の閾値 c が 0 または 1 であるようなフラグメントに関しては、小モデル性をもたないが、充足可能性判定は決定可能であることが示されている。

定理 3.45 ([21]) PCTL の定性的フラグメントの充足可能性判定問題は決定可能である。 ■

モデル検査

CTL のモデル検査 ($\mathcal{T} \models \varphi$) は、検査式 φ 中に表れる部分状態式 φ' が「検査対象である状態遷移系 $\mathcal{T} = \langle S, \rightarrow, S_{init}, AP, \Lambda \rangle$ の各状態 $s \in S$ において成り立つ ($\mathcal{T}, s \models \varphi'$) か否か」を再帰的に判定することで効率的に計算できる。

定理 3.46 ([38, 37]) CTL モデル検査は多項式時間で決定可能である。 ■

CTL の確率拡張時間論理である PCTL/CSL でも、行列やベクトルの数値計算をベースとした手法 [51, 11, 12, 59] により多項式時間で決定できる。また、統計的手法を用いた近似的モデル検査手法 [92] も開発されている。

定理 3.47 ([51, 11, 12, 59]) DTMC/CTMC に対する PCTL/CSL モデル検査は多項式時間で決定可能である。 ■

確率的モデル検査器としては、PRISM[70, 58] や Ymer[88, 89] などが開発されている。

3.3 オートマトン

ω -オートマトン (ω -automata) は、与えられた無限長語の受理/不受理⁶を判定する状態遷移機械の一種である。オートマトンは、有限の状態の集合とそれらの間の遷移関係からなり、与えられた語の文字を順に読み、現在の状態から次の状態へ遷移してゆく。オートマトンの遷移関係は、状態遷移系とは異なり、現在の状態だけでなく読んだ文字によって次にどの状態へ遷移できるかを規定する。また、オートマトンには**受理条件** (acceptance condition) が設定されており、語に対するふるまいの特徴が受理条件を満たすかによって語の受理/不受理を判定する。オートマトンが受理する言語の集合、すなわち、認識する言語のクラスは、設定されている受理条件に依存している。 ω -オートマトンの分野では、 ω -正規言語を認識できる ω -**正規オートマトン** (ω -regular automata) が広く研究されてきた。 ω -オートマトンという呼称は、多くの場合、これらのオートマトンを指すことも多いが、全く異なる種類の受理条件をもつ ω -オートマトンも近年になって研究されるようになってきている [6]。

本節では、 ω -オートマトンの定義を与える。また、 ω -正規オートマトンと、平均利得によって受理判定を行う**多閾値平均利得オートマトン** (multi-threshold mean-payoff ω -automata, MTMPA, あるいは単に MPA) について簡単に紹介する。そして、これらのオートマトンによって規定できる言語のクラスについても述べる。

3.3.1 ω -オートマトン

ω -オートマトンは、形式的には以下のように定義される。

定義 3.48 (ω -オートマトン) ω -オートマトン $\mathcal{A}$ は五つ組 $\langle Q, \Sigma, \hookrightarrow, q_{init}, Acc \rangle$ である。ここで、

- Q は有限の状態集合,
- Σ は有限のアルファベット,
- $\hookrightarrow \subseteq Q \times \Sigma \times Q$ はラベル付き遷移関係,

⁶広義には、無限長語に対して、何らかの値を返す機械である。ブール値 (受理/不受理) でない値を返す、つまり、定量的な言語を扱うオートマトンは、[43, 57, 30, 6, 28, 27] などで研究されている。

- $q_{init} \in Q$ は初期状態.
- Acc は受理条件を規定するためのパラメータである.
 - 受理条件については, 3.3.2 項と 3.3.3 項で詳しく述べる. ■

オートマトンのふるまいは, 現在の状態と読んだ文字から遷移可能な状態を非決定的に選択していくことで得られる遷移列として定義される. これを**実行** (run) と呼ぶ.

定義 3.49 (実行) $\mathcal{A} = \langle Q, \Sigma, \hookrightarrow, q_{init}, F \rangle$ を ω -オートマトンとする. このとき, 語 $\sigma = a_0 a_1 \dots$ に対して, 無限遷移列 $r = \langle q_{init}, a_0, q_1 \rangle \langle q_1, a_1, q_2 \rangle \dots \in \hookrightarrow^\omega$ を, σ に対する $\mathcal{A}$ の実行と呼ぶ. 逆に, 実行 r に対して, σ を r 上の語, 状態列 $\rho = q_{init} q_1 \dots$ を r 上のパスといい, それぞれ $\text{word}(r)$, $\text{path}(r)$ と書く.

また, $\mathcal{A}$ のすべての実行, 語, パスの集合をそれぞれ $\text{Runs}^{\mathcal{A}}$, $\text{Words}^{\mathcal{A}}$, $\text{Paths}^{\mathcal{A}}$ と書く. ■

一般に, それぞれの状態とそのとき読んだ文字によって遷移可能な状態は一つとは限らないため, 語に対する実行は複数存在しうる. そして, ω -オートマトンは, 語に対する実行が受理条件 Acc を満たすか否かによってその語を受理するか否かを判定する. 受理条件を満たす実行を**受理実行**といい, 受理実行が存在する語を ω -オートマトンは**受理**する. 別の言い方をすれば, 受理実行 r 上の無限語 $\text{word}(r)$ を受理する. オートマトン $\mathcal{A}$ によって受理される語の集合, つまり, $\mathcal{A}$ によって**認識**される言語を $L(\mathcal{A})$ と書く.

決定性と全域性

一般にオートマトンがある状態である文字を読んだときには, 複数の状態へ遷移可能である. ただし, そのようなオートマトンは認識する言語に関する性質の解析が困難な場合がある.

オートマトンの遷移関係が, 任意の状態 q と文字 a に対して遷移先が一つ以下であるとき, すなわち,

$$\forall q \in Q. \forall a \in \Sigma. (|\{q' \mid \langle q, a, q' \rangle \in \hookrightarrow\}| \leq 1)$$

であるとき, そのオートマトンは**決定的** (deterministic) であるという.

また, 任意の状態 q と文字 a に対して遷移先が一つ以上あるとき, すなわち,

$$\forall q \in Q. \forall a \in \Sigma. (|\{q' \mid \langle q, a, q' \rangle \in \hookrightarrow\}| \geq 1)$$

であるとき, そのオートマトンは**全域的** (total) であるという.

到達可能性と強連結成分

ラベルなし遷移関係と同様に、ラベル付き遷移 $\langle q, a, q' \rangle \in \hookrightarrow$ を $q \xrightarrow{a} q'$ と表記する。ここで、前状態 q を $pre(\delta)$ 、文字 a を $letter(\delta)$ 、後状態 q' を $post(\delta)$ で表す。また、ラベルについて言及する必要がない場合、単に $q \hookrightarrow q'$ と書く。そして、オートマトン上の到達可能関係や (M/B)SCC も、それぞれ定義 3.6 と定義 3.7 と同様に定義される。ただし、遷移関係 $\hookrightarrow$ をもとにした推移閉包 $\hookrightarrow^+$ では、必要に応じて、有限回遷移列（有限長実行） $\rho = q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} \dots \xrightarrow{a_n} q_{n+1}$ からなる推移閉包関係を $q_1 \xrightarrow{a_1 \dots a_n^+} q_{n+1}$ のようにラベル付きで記述する。また、その有限回遷移列 ρ のことを**実行断片**（run fragment）といい、 $|\rho|$ でその長さ n を表す。

そして、開始状態と終端状態が等しい実行断片 $q_1 \xrightarrow{a_1} \dots \xrightarrow{a_n} q_1$ を**閉路**（cycle）と呼ぶ。さらに、途中で同じ状態が複数回出現しない、すなわち、 $\forall i < j. q_i \neq q_j$ を満たす閉路 ρ を**単純閉路**（simple cycle）と呼び、SCC $\langle Q', \hookrightarrow' \rangle$ 上のすべての単純閉路の集合を $C_{\text{simp}}^A(\langle Q', \hookrightarrow' \rangle)$ と書くこととする。ただし、集合 $C_{\text{simp}}^A(\langle Q', \hookrightarrow' \rangle)$ では、閉路 $\delta_1 \dots \delta_n$ と閉路 $\delta_n \delta_1 \dots \delta_{n-1}$ を区別しないものとする。ここで、遷移関係 $\hookrightarrow$ は有限であるため、任意の閉路は単純閉路の有限回の連結/挿入で構成されており、任意の実行 $r = r_0 r_1 r_2 \dots$ は実行断片 r_0 の後ろに無限個の閉路 $r_1, r_2, \dots$ を連結した形をもつ。

3.3.2 ω -正規オートマトン

ω -正規オートマトンは、最もよく研究されている ω -オートマトンであり、 ω -正規言語と同じ表現力を持っている。 ω -正規言語を認識するための受理条件はいくつもあるが、ここでは、Büchi 条件あるいは Rabin 条件をもつオートマトンのみを示す。

ω -正規オートマトンの受理条件は、実行 r 上のパス $\rho = \text{path}(r)$ に無限にしばしば現れる状態の集合 $\text{Inf}(\text{path}(r))$ を用いて与えられる。

$$\text{Inf}(\rho) = \{q \in Q \mid \forall i. \exists j > i. (\rho[j] = q)\}. \quad (3.57)$$

Büchi/Rabin 受理条件は、これを用いて以下のように定義される。

定義 3.50 (Büchi 受理条件と Büchi オートマトン) ω -オートマトン $\langle Q, \Sigma, \hookrightarrow, q_{\text{init}}, \text{Acc} \rangle$ の受理条件パラメータ Acc が無限にしばしば訪れるべき状態の集合 $F \subseteq 2^Q$ で与えられているとき、そのオートマトンを **Büchi オートマトン** (BA) と呼ぶ。

このオートマトンは、以下のように定義される **Büchi 受理条件** $\text{Büchi}_F(r)$ を満たす実行 r を受理する.

$$\text{Büchi}_F(r) = \text{Inf}(\text{path}(r)) \cap F \neq \emptyset. \quad (3.58)$$

■

なお、この F は **受理状態集合** と呼ばれる.

定義 3.51 (Rabin 受理条件と Rabin オートマトン) ω -オートマトン $\langle Q, \Sigma, \hookrightarrow, q_{\text{init}}, \text{Acc} \rangle$ の受理条件パラメータ Acc が「実行上で多くても有限回しか出現すべきでない状態の集合 f_i 」と「実行上で無限回出現すべき状態の集合 f'_i 」の対の和集合 $F = \bigcup_i \{\langle f_i, f'_i \rangle\} \in 2^{2^Q \times 2^Q}$ で与えられているとき、そのオートマトンを **Rabin オートマトン** (RA) と呼ぶ.

このオートマトンは、以下のように定義される **Rabin 受理条件** $\text{Rabin}_F(r)$ を満たす実行 r を受理する.

$$\text{Rabin}_F(r) = \exists \langle f_i, f'_i \rangle \in F. (\text{Inf}(\text{path}(r)) \cap f_i = \emptyset \wedge \text{Inf}(\text{path}(r)) \cap f'_i \neq \emptyset) \quad (3.59)$$

■

なお、 F の要素 $\langle f_i, f'_i \rangle$ は **Rabin 対** と呼ばれる.

例 3.52 BA $\langle \{q_0, q_1, q_2\}, \{a, b, c\}, \hookrightarrow, q_0, \{q_0\} \rangle$ を考える (図 3.5). ただし,

$$\begin{aligned} \hookrightarrow = \{ & \langle q_0, a, q_1 \rangle, \langle q_0, b, q_0 \rangle, \langle q_0, c, q_0 \rangle, \\ & \langle q_1, a, q_2 \rangle, \langle q_1, b, q_0 \rangle, \langle q_1, c, q_2 \rangle, \\ & \langle q_2, a, q_2 \rangle, \langle q_2, b, q_2 \rangle, \langle q_2, c, q_2 \rangle \} \end{aligned}$$

とする.

このオートマトンでは、 $(ab)^\omega$ や $ccb(cabb)^\omega$ などの a の後には b が続くような語を受理する. なぜならば、そのような語に対しては、 q_0 を無限にしばしば訪れる実行が存在するからである. そして、 $(abacb)^\omega$ などの a の後には b が続かない部分をもつ語は受理しない. なぜならば、どのような状態にいても、文字 a を読んで遷移した後で文字 b を読んで遷移できる状態は q_2 のみであり、その後 q_0 を訪れることができないためである.

なお、このオートマトンは決定的かつ全域的である. ■

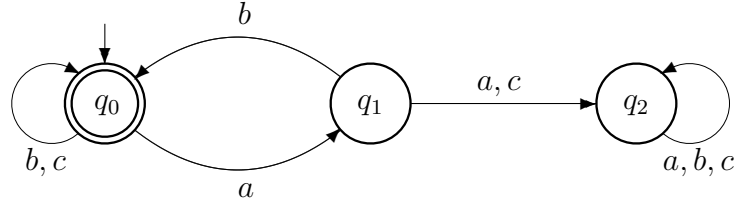


図 3.5: 例 3.52 の Buchi オートマトン

3.3.3 多閾値平均利得オートマトン

近年になって研究されるようになってきた利得（重み）付きの ω -オートマトンは，古典的な ω -オートマトンが実行の特徴として捉える「ある状態を無限にしばしば訪れる」という性質ではなく，平均利得などによってふるまいを特徴付ける．特徴の捉え方にはいくつもアプローチがあるが，重みを持つ意味を直観的に理解しづらい場合もある．Alur らは，実行に対する平均利得の集積点集合に関する閾値条件からなる受理条件をもつ全域的決定性オートマトンを定義している [6]．

[6] で定義されているオートマトンでは，遷移関係 $\hookrightarrow$ 上の重み関数 $W : \hookrightarrow \rightarrow \mathbb{R}^d$ によって与えられた実行上の多次元重みの列，すなわち，トレースの平均利得によって受理/不受理を決定する．利得系で定義されたパス上の利得トレースと利得トレースと同様に，重み関数によって与えられる実行上の利得トレースと平均利得とトレースは以下のように定義される．

定義 3.53 (実行上の利得トレースと平均利得トレース) $\mathcal{A} = \langle Q, \Sigma, \hookrightarrow, q_0, Acc \rangle$ をオートマトン， $W : \hookrightarrow \rightarrow \mathbb{R}^d$ を $\mathcal{A}$ の遷移関係上の重み関数とする．

このとき， $\mathcal{A}$ の実行 $r = \delta_0 \delta_1 \dots \in \text{Runs}^{\mathcal{A}}$ に対して，以下のように定義されるトレース $\text{payoff}_W(r)$ を r 上の利得トレースという．

$$\text{payoff}_W(r)[i] = W(\delta_i). \quad (3.60)$$

また，以下のように定義されるトレース $\text{mp}_W(r)$ を r 上の平均利得トレースという．

$$\text{mp}_W(r)[i] = \frac{1}{i+1} \sum_{j=0}^i W(\delta_j). \quad (3.61)$$

■
 W の i 番目の次元の重み関数を $W[i]$ と書く. すなわち, $W[i](\delta) = \pi_i W(\delta)$ とする. 3.1.4 項でも述べたように, 実行 r の集積点集合 $\mathcal{Acc}(\text{mp}_W(r))$ に対する i 次元成分への射影区間 $\pi_i(\mathcal{Acc}(\text{mp}_W(r)))$ の最大値 (最小値) は, 平均利得トレースの i 次元成分射影列 $\text{mp}_{W[i]}(r)$ の上極限 $\limsup_{n \rightarrow \infty} \text{mp}_{W[i]}(r)[n]$ (下極限 $\liminf_{n \rightarrow \infty} \text{mp}_{W[i]}(r)[n]$) と一致している. そして [6] では, 実行の集積点集合に対するある次元成分への射影区間の最大値 (または最小値) の閾値条件 $\max \pi_i(\cdot) \sim c$ (または $\min \pi_i(\cdot) \sim c$) の Boolean 結合で与えられる受理条件をもったオートマトンが提案されていた.

本論文では, そのようなオートマトンを以下のように定義する. ただし, [6] の定義ではオートマトンは全域的決定性でなくてはならないが, 本論文ではそのような制限がなくてもよいオートマトンとして定義する.

定義 3.54 (多閾値平均利得受理条件と MTMPA) ω -オートマトン $\langle Q, \Sigma, \hookrightarrow, q_{\text{init}}, \text{Acc} \rangle$ の受理条件パラメータ Acc が以下の対 $\langle W, G \rangle$ で与えられているとき, そのオートマトンを **多閾値平均利得 ω -オートマトン**

- $W : \hookrightarrow \rightarrow \mathbb{R}^d$ は遷移に d -次元実ベクトルに関連付ける重み関数.
- $G : 2^{\mathbb{R}^d} \rightarrow \text{Bool}$ は, 実行の平均利得の集積点集合 ($\subseteq \mathbb{R}^d$) に関する述語の Boolean 結合 (集積点条件) である.
 - ただし, G の原子述語は, 集積点集合の i 次元成分への射影区間の最大値または最小値に関する閾値条件 $\text{opt } \pi_i(\cdot) \sim c$ である. ここで, $\text{opt} \in \{\max, \min\}$, $c \in \mathbb{R}$, $\sim \in \{<, >, \leq, \geq\}$ とする.

このオートマトンは, 以下のように定義される **多閾値平均利得受理条件** (multi-threshold mean-payoff acceptance condition, MTMP 受理条件, あるいは単に平均利得受理条件, MP 受理条件) $\text{MP}_{\langle W, G \rangle}(r)$ を満たす実行 r を受理する.

$$\text{MP}_{\langle W, G \rangle}(r) = G(\mathcal{Acc}(\text{mp}_W(r))). \quad (3.62)$$

■

例 3.55 例 3.52 の BA の受理条件を以下の重み関数 W と多閾値平均利得条件 G の対 $\langle W, G \rangle$ に置き換えた MTMPA を考える.

$$W(\delta) = \begin{cases} 1 & \text{if } \delta \in \{\langle q_2, x, q_2 \rangle \mid x \in \{a, b, c\}\}, \\ 0 & \text{otherwise,} \end{cases} \quad (3.63)$$

$$G(\cdot) = \max \pi_1(\cdot) \leq 0. \quad (3.64)$$

このとき、状態 q_2 に到達するすべての実行の利得トレースは 1 に収束するため、受理されない。そうでないすべての実行の利得トレースは 0 に収束するため、受理される。さらに状態 q_0 を無限にしばしば訪れることにもなるため、この MTMPA は例 3.52 の BA と等価である。■

全称閾値条件と存在閾値条件

集積点条件 G を構成する原子命題のうち、 $\max \pi_i(\cdot) < c$, $\max \pi_i(\cdot) \leq c$, $\min \pi_i(\cdot) > c$, $\min \pi_i(\cdot) \geq c$ の形の原子命題は、 i 次元成分への射影区間の「すべて」の値が閾値をみたすという条件である。そのため、これらの閾値条件を**全称閾値条件** (universal threshold condition) と呼ぶ。

一方、 $\max \pi_i(\cdot) < c$, $\max \pi_i(\cdot) \leq c$, $\min \pi_i(\cdot) > c$, $\min \pi_i(\cdot) \geq c$ の形の原子命題は、 i 次元成分への射影区間の「ある」の値が閾値をみたすという条件である。そのため、これらの閾値条件を**存在閾値条件** (existential threshold condition) と呼ぶ。

注意 3.56 ([6] で定義されているオートマトンとの関係) Alur らは、[6] の中で全域的決定性 MTMP オートマトンとそれによって規定される言語のクラス、閉包性、空判定手法について研究している。全域的決定性 MTMP オートマトンが認識する言語は**多閾値平均利得言語** (multi-threshold mean-payoff language) と呼ばれる。この言語のクラスは、例 3.55 で示したように一部の ω -正規言語は規定することができるが、 ω -正規言語のクラスとは比較不能である。また、ブール演算に関して閉じており、空判定問題も決定可能である。■

多閾値平均利得受理条件の等価性

対 $\langle W, G \rangle$ を受理条件パラメータとする d -次元 MTMPA と、 $\mathcal{A}'$ の遷移関数上の重み関数 W' 、集積点条件 G' を考える。このとき、 $L(\mathcal{A}) = L(\mathcal{A}[\langle W, G \rangle / \langle W', G' \rangle])$ ならば、 $\mathcal{A}$ 上で $\langle W, G \rangle$ と $\langle W', G' \rangle$ は等価といい、 $\langle W, G \rangle \approx_{\mathcal{A}} \langle W', G' \rangle$ と書く。

上極限/下極限の基本的な性質から、集積点条件 G に含まれる i 番目次元に関する原子命題を $p_1, \dots, p_n$ (ただし、 $p_m = \text{opt}_m \pi_i(\cdot) \sim_m c_m$) としたとき、

以下のような等価関係が成り立つ.

$$\begin{aligned}
\langle W, G \rangle &\approx_{\mathcal{A}} \langle W[(W[i] + c')/W[i]], G[(c_m + c)/c_m \text{ for all } m] \rangle \\
&\approx_{\mathcal{A}} \langle W[(-W[i])/W[i]], G[(\overline{\text{opt}}_m \pi_i(\cdot) \sim_m -c_m)/p \text{ for all } m] \rangle \\
&\approx_{\mathcal{A}} \langle W[(c' \cdot W[i])/W[i]], G[(c \cdot c_m)/c_m \text{ for all } m] \rangle \\
&\approx_{\mathcal{A}} \langle W[W[i], W[j]/W[j], W[i]], G[\pi_i, \pi_j/\pi_j, \pi_i] \rangle \\
&\approx_{\mathcal{A}} \langle \langle W[1], \dots, W[d], W[i] \rangle, G[(\text{opt}_m \pi_{d+1}(\cdot) \sim_m c_m)/p_m] \rangle \quad (3.65)
\end{aligned}$$

ただし, $c \in \mathbb{R}_{>0}$, $(c \cdot W[i])(\delta) = c \cdot W[i](\delta)$, $(W[i] + c)(\delta) = W[i](\delta) + c$, $(-W[i])(\delta) = -W[i](\delta)$, $\overline{\text{opt}} \neq \text{opt}$ とする.

よって, 任意の MTMP 受理条件パラメータ対 $\langle W, G \rangle$ は, すべての原子述語が「集積点集合のある次元の最小値に関する閾値 0 の閾値条件」である等価な対 $\langle W', G' \rangle \approx_{\mathcal{A}} \langle W, G \rangle$ に変換できる. また, G は閾値条件の Boolean 結合で与えられた述語であるため, 否定を含まない積和標準形に等価変換できる. 従って, 集積点条件 G は以下のような形に限定しても一般性は失われない.

$$G(\cdot) = \bigvee_i \left(\bigwedge_j \min \pi_{k_{ij}}(\cdot) \sim_{ij} 0 \right). \quad (3.66)$$

そして, 以降では, この G を 4 つのインデックス集合 $H_i^>, H_i^{\geq}, H_i^<, H_i^{\leq} \subseteq \{1, \dots, d\}$ の組 $\langle H_i^>, H_i^{\geq}, H_i^<, H_i^{\leq} \rangle$ の集合で略記する.

$$G(\cdot) \equiv \{ \langle H_i^>, H_i^{\geq}, H_i^<, H_i^{\leq} \rangle \mid k_{ij} \in H_i^{\sim} \Leftrightarrow \sim_{ij} \text{ is } \sim \}. \quad (3.67)$$

直観的には, $H^{\sim}$ は「その射影区間の最小値が閾値条件 ~ 0 を満たすべき次元 k 」の集合である.

3.3.4 言語クラス

本項では, 3.3.2 項と 3.3.3 項で与えたオートマトンによって認識される言語や LTL で規定される言語のクラスについて述べる. 本論文では, オートマトンの呼称として, 決定性をもつことを “D”, 受理条件が Büchi 条件であることを “B”, 受理条件が Rabin 条件であることを “R”, 受理条件が多閾値平均利得条件であることを “MTMP”, オートマトンを “A” で表記する. 例えば, 決定性 Rabin オートマトンは DRA と表記する. そしてこれらのオートマトンが認識する言語のクラスを $\mathcal{L}(\cdot)$ で表す. 例えば, Büchi オートマトンの認識する言語のクラスは $\mathcal{L}(\text{BA})$ と表記する.

BA や RA が認識できる言語を ω -正規言語 (ω -regular languages) と呼ぶ. ω -正規言語のクラスは, 空文字言語 $\{\epsilon\}$ を除く正規言語クラス (付録 A) の ω -Kleene 閉包として定義される.

定義 3.57 (ω -Kleene 閉包と ω -正規言語) $\mathcal{L} \subseteq 2^{\Sigma^*}$ を有限長言語の族 (クラス) とする. このとき, 言語族 $\mathcal{L}$ の ω -Kleene 閉包は以下の ω -KC(L)

$$\omega\text{-KC}(L) = \left\{ \bigcup_i (U_i \cdot V_i^\omega) \mid \forall i. U_i, V_i \in \mathcal{L} \setminus \{\emptyset\} \right\}. \quad (3.68)$$

そして, REG を正規言語の族としたとき, $\omega\text{-KC}(\text{REG} \setminus \{\epsilon\})$ で定義される言語族を ω -正規言語族といい, $\omega\text{-REG}$ と書く. ■

オートマトンと LTL, ω -正規言語のクラスについては, 以下のような事実が知られている.

定理 3.58 (BA, RA, DMTMPA[6], LTL と ω -正規言語の表現力)

- $\mathcal{L}(\text{BA}) = \mathcal{L}(\text{RA}) = \mathcal{L}(\text{DRA}) = \omega\text{-REG}$.
- $\mathcal{L}(\text{DBA}) \subsetneq \omega\text{-REG}$, $\mathcal{L}(\text{LTL}) \subsetneq \omega\text{-REG}$.
- $\mathcal{L}(\text{DBA}) \not\subseteq \mathcal{L}(\text{LTL})$, $\mathcal{L}(\text{LTL}) \not\subseteq \mathcal{L}(\text{DBA})$.
- $\mathcal{L}(\text{DMTMPA}) \not\subseteq \omega\text{-REG}$, $\omega\text{-REG} \not\subseteq \mathcal{L}(\text{DMTMPA})$ ⁷. ■

LTL で定義できる言語は counter-free (star-free) ω -正規言語とも呼ばれ, そのクラスは $\omega\text{-REG}$ の真部分集合であることが知られている. そして, [6] では, 多閾値平均利得言語のクラス $\mathcal{L}(\text{DMTMPA})$ が ω -正規言語のクラスと比較不能であることを示している.

LTL から ω -正規オートマトンへの変換

LTL は線形時間的性質を直観的に記述できるものの, LTL 式のままでは検証に利用しづらい場合がある. そのため, その式の規定する言語を認識するオートマトン, すなわち, 式と等価なオートマトンへ変換し, それをオートマトン理論で用いられる手法によって解析することで, しばしば形式的検証が行われている. そのような手法をオートマトンベース手法という.

LTL とオートマトンの関係については, 以下の定理が成り立つ.

⁷[6] では全域的な DMTMPA に関して示しているが, 同様の議論が全域的でないものでも成り立つ.

定理 3.59 (LTL から BA への変換 [84, 85]) 任意の LTL 式 φ に対して, $L(\varphi)$ を認識する, $\mathcal{O}(2^{|\varphi|})$ サイズの BA を構成可能である. ■

LTL から BA への変換器としては, LTL2BA[62] や SPOT[78] などが開発されている.

また, オートマトンベースの検証手法ではしばしばオートマトンの決定化が必要である. 有限長語上の有限状態オートマトンは, 冪集合構成法 (powerset construction) と呼ばれる単純な手法で決定化できる. 一方, ω -オートマトンは, 「ある状態を無限にしばしば訪れる」という条件を用いて受理/不受理を判定するため, 一般に冪集合構成法では決定化できない. BA から DRA への変換では, Safra の構成法 [73] (Safra's construction) と呼ばれる, 非常に複雑な手続きが必要である.

定理 3.60 (BA から DRA への変換 [73]) 任意の BA $\mathcal{A}$ に対して, $L(\mathcal{A})$ を認識する, $\mathcal{O}(2^{|\mathcal{A}|})$ サイズ (最小上界) の DRA を構成可能である. ■

この変換器としては, ltl2dstar[55, 63] 等が開発されている.

3.4 リアクティブシステム

OS や組み込み制御系などの他モジュールやユーザから入力に対して過去の情報のみを用いて適切に応答しなければならないシステムを**リアクティブシステム** (reactive system, RS) という. こうしたシステムは, 不具合が深刻な損害を与えることも多く, 特に高い安全性が求められている. システムの仕様を記述する際には命題とアクション (あるいはイベント) を同一視することが多いが, RS 仕様の記述では, 原子命題 (アクションあるいはイベント) 集合 AP を RS 側が管理する**出力原子命題集合** OAP と環境側が管理する**入力原子命題集合** IAP の二種類に分けて記述に用いる. ただし, $AP = OAP \cup IAP$ かつ $OAP \cap IAP = \emptyset$ である. RS は外部環境からの (未来の) 入力 $b \in 2^{IAP}$ を直接制御することはできないため, 過去の情報のみを用いてその時点の出力 $a \in 2^{OAP}$ を適切に選択しなければならない.

形式的には, RS と環境は, 互いに相手から与えられる (自身にとっては非決定的な) アクションに対して確率的に応答するシステム, つまり, ある種の **Markov 決定過程** (Markov decision process, MDP) としてモデル化される.

定義 3.61 (MDP) MDP は五つ組 $\langle S, \Sigma_I, \Sigma_O, P, s_0 \rangle$ である. ただし,

- S は可算の状態集合,
- Σ_I/Σ_O は $\Sigma_I \cap \Sigma_O = \emptyset$ である入/出力アクションの集合,
- $P : S \times \Sigma_I \times \Sigma_O \rightarrow ([0, 1] \times S)$ は遷移確率関数,
- $s_0 \in S$ は初期状態である.

また, 任意の状態 s と入力アクション $A \in \Sigma_I$ について, 以下の条件が成り立っているものとする.

$$\sum_{B \in \Sigma_O} \text{Prob}(P(s, A, B)) = 1. \quad (3.69)$$

■

$P(s, A, B)$ は, 現在の状態 s で入力アクション $A \in \Sigma_I$ が与えられたときにアクション $B \in \Sigma_O$ を出力する確率 p とその際に遷移する先の状態 s' の対 $\langle p, s' \rangle$ を与える関数である.

RS は, MDP の中でも入力 $a \in 2^{IAP}$ に対して出力 $b \in 2^{OAP}$ を返すものであり, 以下のように定義される.

定義 3.62 (MDP RS) RS $\mathcal{R}$ は, 入/出力アクション Σ_I/Σ_O , 遷移確率関数 P に関して, 以下の制約をもつ MDP である.

- $\Sigma_I = 2^{IAP}$,
- $\Sigma_O = 2^{OAP}$,
- 初期状態 s_0 に関して, 任意の入力アクション $A, A' \in \Sigma_I$ と出力アクション $B \in \Sigma_O$ に関して,

$$P(s_0, A, B) = P(s_0, A', B). \quad (3.70)$$

■

初期状態 s_0 の遷移確率関数に関する制約から, RS の最初の出力は入力に依存しない. RS と環境の相互作用開始時にはどちらか一方 (この定義では RS) が入力なしで出力しなければならないためこの最初の入力を無視することで相互作用開始時の同期を行う.

なお, 無限状態をもつ RS は現実には実装できないため, 実用的な RS は有限状態である必要がある. そのような状態が有限の RS を**有限メモリ** (finite memory), 無限の RS を**無限メモリ** (infinite memory) という.

一方で, 環境は, $2^{OAP}/2^{IAP}$ を入/出力アクションとする MDP である.

定義 3.63 (MDP 環境) 環境 $\mathcal{E}$ は、入/出力アクション Σ_I/Σ_O に関して、以下の制約をもつ MDP である。

- $\Sigma_I = 2^{OAP}$,
- $\Sigma_O = 2^{IAP}$. ■

環境に関しても、状態数が有限/無限のときには有限/無限メモリであるという。

RS $\mathcal{R}$ と環境 $\mathcal{E}$ のふるまいは、互いに相手の出力（自身の入力）に対して確率的な出力（相手の入力）と、それに伴う状態遷移の繰り返しにより構成される。ただし、既に述べたように、RS の最初の入力を無視することで RS と環境の相互作用の同期を行う。

形式的には、RS と環境の相互作用は DTMC としてモデル化できる。

定義 3.64 (RS と環境のふるまいモデル) $\mathcal{R} = \langle S, 2^{IAP}, 2^{OAP}, P, s_0 \rangle$ を MDP RS, $\mathcal{E} = \langle S', 2^{OAP}, 2^{IAP}, P', s'_0 \rangle$ を MDP 環境とする。このとき、RS $\mathcal{R}$ と環境 $\mathcal{E}$ のふるまいは、DTMC $\mathcal{D}^{\mathcal{R}, \mathcal{E}} = \langle S \times S' \times 2^{IAP \cup OAP}, \mathbf{P}, P_{init}, IAP \cup OAP, \Lambda \rangle$ によってモデル化される。ただし、

- それぞれの遷移確率関数 $P(s_1, b_1, a_2) = \langle p, s_2 \rangle$ と $P'(s'_1, a_2, b_2) = \langle p', s'_2 \rangle$ に対して、

$$\mathbf{P}(\langle s_1, s'_1, a_1 \cup b_1 \rangle, \langle s_2, s'_2, a_2 \cup b_2 \rangle) = p \cdot p',$$

- それぞれの初期状態 s_0 と s'_0 に関する遷移確率関数 $P(s_0, b_0, a_1) = \langle p, s_1 \rangle$ と $P'(s'_0, a_1, b_1) = \langle p', s'_1 \rangle$ に対して、

$$P_{init}(\langle s_1, s'_1, a \cup b \rangle) = p \cdot p',$$

- $\Lambda(\langle s, s', a \cup b \rangle) = a \cup b$. ■

よって、この DTMC のふるまいを解析することで、RS と環境の相互作用が持つ性質を検証することができる。

注意 3.65 (純粋 RS と確率的 RS) MDP RS $\mathcal{R} = \langle S, 2^{IAP}, 2^{OAP}, P, s_0 \rangle$ が以下を満たすとき、 $\mathcal{R}$ は純粋 (pure) であるといい、それでないときには確率的 (randomized) であるという。

$$\forall s \in S. \forall b \in 2^{IAP}. \exists a \in 2^{OAP}. P(s, b, a) = 1. \quad (3.71)$$

実現可能性判定/合成においては、純粋 RS のみに注目する場合が多い。 ■

実現可能性

RS の仕様は、充足可能性を満たすだけでは不十分である。なぜならば、過去の情報のみを用いて仕様を満たすように適切に出力を選択し続けなければならないためである。言語 $L \subseteq (2^{IAP \cup OAP})^\omega$ で規定された仕様を満たすように適切に出力を選ぶことが可能であるとき、すなわち、その仕様を実現する RS が存在するとき、仕様 L は**実現可能** (realizable) あるいは**プログラム可能** (programmable) であるという [68, 1, 72].

定義 3.66 (実現可能性 [68, 1, 72]) $L \subseteq (2^{IAP \cup OAP})^\omega$ を RS 仕様とする。このとき、以下を満たす (有限メモリ) RS $\mathcal{R}$ が存在するとき、 L は (有限メモリ) 実現可能であるという。

$$\exists \mathcal{R}. \forall \mathcal{E}. \text{Words}^{\mathcal{D}^{\mathcal{R}, \mathcal{E}}} \subseteq L. \quad (3.72)$$

ただし、本研究では、仕様 L を「ほとんど確かに」満たすように適切に出力を選ぶ RS に注目する。すなわち、以下のように定義される**確率的実現可能性** (probabilistic/almost-sure realizability) について考える。

定義 3.67 (確率的実現可能性) $L \subseteq (2^{IAP \cup OAP})^\omega$ を RS 仕様とする。このとき、以下を満たす (有限メモリ) RS $\mathcal{R}$ が存在するとき、 L は (有限メモリ) 確率的実現可能であるという。

$$\exists \mathcal{R}. \forall \mathcal{E}. \Pr^{\mathcal{D}^{\mathcal{R}, \mathcal{E}}}(\{\sigma \in \text{Words}^{\mathcal{D}^{\mathcal{R}, \mathcal{E}}} \mid \sigma \in L\}) = 1. \quad (3.73)$$

そのような $\mathcal{R}$ を (有限メモリ) 実現プログラム、あるいは単に実現と呼び、 L を実現するすべての (有限メモリ) 実現の集合を $\text{Real}_{(\text{fin})}^L$ と書く。 ■

以下では、この確率的実現可能性を単に実現可能性と呼ぶ。

ここで、RS $\mathcal{R}$ と環境 $\mathcal{E}$ が固定されているとき、そのふるまいは DTMC $\mathcal{D}^{\mathcal{R}, \mathcal{E}}$ でモデル化できるため、確率空間 $\langle \text{Paths}^{\mathcal{D}^{\mathcal{R}, \mathcal{E}}}, \mathcal{F}^{\mathcal{D}^{\mathcal{R}, \mathcal{E}}}, \Pr^{\mathcal{D}^{\mathcal{R}, \mathcal{E}}} \rangle$ 上で可積分な確率変数 $X : (2^{IAP \cup OAP})^\omega \rightarrow \mathbb{R}$ に対する期待値 $\text{Expect}^{\mathcal{R}, \mathcal{E}}[X]$ は以下のように与えられる。

$$\text{Expect}^{\mathcal{R}, \mathcal{E}}[X] = \int_{\sigma \in \text{Words}^{\mathcal{D}^{\mathcal{R}, \mathcal{E}}}} X(\sigma) \, d\Pr^{\mathcal{D}^{\mathcal{R}, \mathcal{E}}}(\sigma). \quad (3.74)$$

そのため、そのような確率変数 X で与えられる目的関数に関する最適実現プログラムは以下のように定義できる。なお、本研究では目的関数は大きいほどよいものと仮定する。

定義 3.68 (最適実現プログラム) $L \subseteq (2^{IAP \cup OAP})^\omega$ を RS 仕様, $X : (2^{IAP \cup OAP})^\omega \rightarrow \mathbb{R}$ を可積分な目的関数とする. このとき, 以下を満たす L を実現する RS $\mathcal{R} \in \text{Real}^L$ を, X に関する最適実現プログラムという.

$$\inf_{\mathcal{E}} \{\text{Expect}^{\mathcal{R}, \mathcal{E}}[X]\} = \sup_{\mathcal{R}' \in \text{Real}^L} \inf_{\mathcal{E}} \{\text{Expect}^{\mathcal{R}', \mathcal{E}}[X]\}. \quad (3.75)$$

また, $\varepsilon > 0$ に対して以下を満たす RS $\mathcal{R} \in \text{Real}_{\text{fin}}^L$ を, X に関する有限メモリ (ε -) 最適実現プログラムという.

$$\inf_{\mathcal{E}} \{\text{Expect}^{\mathcal{R}, \mathcal{E}}[X]\} + \varepsilon \geq \sup_{\mathcal{R}' \in \text{Real}_{\text{fin}}^L} \inf_{\mathcal{E}} \{\text{Expect}^{\mathcal{R}', \mathcal{E}}[X]\}. \quad (3.76)$$

注意 3.69 (有限メモリ最適性) 仕様が実現可能である場合には, 有限メモリでないものを含めた場合には, 目的関数の極大値を与える (つまり, 最適な) プログラムが存在する. しかしながら, 有限メモリのプログラムに限定した場合には, 厳密に最適なプログラムは一般には存在しない. そのため, 本論文では, 有限メモリ ε -最適なプログラムのことを有限メモリ最適なプログラムと定義する. ■

定義 3.67 では, 任意の環境に対しても仕様を実現するプログラムについて言及している. また, 定義 3.68 では, どのような環境であっても少なくとも保証される性能を最大化する実現プログラムを最適であると定義している. このように, ありとあらゆる運用状況を想定して実現可能性判定/合成を行う手法を**最悪ケース** (worst-case) の判定/合成手法と呼ぶ. 古典的な LTL 実現可能性判定問題では, この最悪ケースの検証を行うことが一般的である. そして, LTL を含む ω -正規仕様については, (確率的) 実現可能であるときには有限メモリ純粋 RS が合成できることが知られている.

しかしながら, 前述の通り, RS の運用状況が具体的に分かっている, すなわち, 既知の有限メモリ MDP 環境を想定することも多い. そのような具体的な運用状況を想定して実現可能性判定/合成を行う手法を**平均ケース** (average-case) の判定/合成手法と呼ぶ. 形式的には平均ケース実現可能性/最適実現は以下のように定義される.

定義 3.70 (平均ケース実現可能性) $L \subseteq (2^{IAP \cup OAP})^\omega$ を RS 仕様, $\mathcal{E}$ を環境とする. このとき, 以下を満たす (有限メモリ) RS $\mathcal{R}$ が存在するとき, $\mathcal{E}$ のもとで L は (有限メモリ) 実現可能であるという.

$$\exists \mathcal{R}. \Pr^{\mathcal{D}^{\mathcal{R}, \mathcal{E}}}(\{\sigma \in \text{Words}^{\mathcal{D}^{\mathcal{R}, \mathcal{E}}} \mid \sigma \in L\}) = 1. \quad (3.77)$$

また, そのような $\mathcal{R}$ を $\mathcal{E}$ のもとでの (有限メモリ) 実現プログラムと呼び, $\mathcal{E}$ のもとで L を実現するすべての (有限メモリ) プログラムの集合を $\text{Real}_{(\text{fin})}^{L, \mathcal{E}}$ と書く. ■

定義 3.71 (平均ケース最適実現プログラム) $L \subseteq (2^{IAP \cup OAP})^\omega$ を RS 仕様, $\mathcal{E}$ を環境, $X : (2^{IAP \cup OAP})^\omega \rightarrow \mathbb{R}$ を可積分な目的関数とする. このとき, 以下を満たす L を実現する RS $\mathcal{R} \in \text{Real}_{(\text{fin})}^{L, \mathcal{E}}$ を, $\mathcal{E}$ のもとでの X に関する最適実現プログラムという.

$$\text{Expect}^{\mathcal{R}, \mathcal{E}}[X] = \sup_{\mathcal{R}' \in \text{Real}_{(\text{fin})}^{L, \mathcal{E}}} \{\text{Expect}^{\mathcal{R}', \mathcal{E}}[X]\}. \quad (3.78)$$

また, $\varepsilon > 0$ に対して以下を満たす L を実現する有限メモリ RS $\mathcal{R} \in \text{Real}_{(\text{fin})}^{L, \mathcal{E}}$ を, $\mathcal{E}$ のもとでの X に関する有限メモリ (ε -) 最適実現プログラムという.

$$\text{Expect}^{\mathcal{R}, \mathcal{E}}[X] + \varepsilon \geq \sup_{\mathcal{R}' \in \text{Real}_{(\text{fin})}^{L, \mathcal{E}}} \{\text{Expect}^{\mathcal{R}', \mathcal{E}}[X]\}. \quad (3.79)$$

■

3.5 ゲーム

ゲーム理論は, 複数の主体の意思決定に関する理論であり, 特に経済学的へ応用が盛んに研究されている. そして, 実現可能性判定においては, RS と環境のふるまいはしばしば**展開型二人交互手番ゲーム** (extensive-form two-player alternate-turn game, 単にゲーム) として形式化される. このゲームは, ある種の二部状態遷移系であり, オートマトンの受理条件と同様の**勝利条件** (winning condition) が設定されている. それぞれの状態では二人の意思決定主体 (プレイヤー 0 とプレイヤー 1), のどちらか一方プレイヤー i ($i \in \{0, 1\}$) が次の遷移先を決めることができる. その遷移先は他方のプレイヤー $\bar{i}$ ($\bar{i} \neq i$) が遷移先を決めることができる状態になっている. そして, 交代で遷移先を決めていくことでゲームが進行する. ゲームの勝敗は, そのゲームの進行 (プレイ) の特徴によって決定される. 実現可能性判定で用いる RS と環境のゲームでは, 仕様を満たすふるまいに対応するようにゲームが進行しているときに RS 側のプレイヤーが勝利するように勝利条件が設定される. これにより, 「仕様が実現可能性であること」と「RS 側のプレイヤーが勝利する意思決定戦略が存在すること」を対応させることができる. 本節では, ゲーム及びそれに関連する概念の定義を与える. なお, 仕様からそれに対応するゲームを構成する手法については 9 章で

説明する.

形式的には、(引き分けなし) 展開型二人交互手番ゲームは以下のように定義される.

定義 3.72 (ゲーム) ゲーム $\mathcal{G}$ は、八つ組 $\langle Q_0, Q_1, \Sigma_0, \Sigma_1, \Delta_0, \Delta_1, q_{init}, \text{Win} \rangle$ である. ただし,

- Q_n ($n \in \{0, 1\}$) はプレイヤー n が遷移先を選択する状態の有限集合,
 - $Q_0 \cap Q_1 = \emptyset$,
- Σ_n はプレイヤー n のアクション集合,
 - $\Sigma_0 \cap \Sigma_1 = \emptyset$,
- $\Delta_n : Q_n \times \Sigma_n \rightarrow Q_{\bar{n}}$ ($\bar{n} \neq n$) はプレイヤー n が遷移先を決定する状態の遷移関数,
- $q_{init} \in Q_0$ は初期状態,
- Win は勝利条件を規定するためのパラメータである.

なお、この勝利条件は前節のオートマトン受理条件と同様のものとして与えられる. ■

このゲーム $\mathcal{G}$ は、状態空間が二部グラフである決定性オートマトン⁸ $\mathcal{A}_{\mathcal{G}} = \langle Q_0 \times Q_1, \Sigma_0 \times \Sigma_1, \Delta_0 \times \Delta_1, q_{init}, \text{Win} \rangle$ と見なすことができる.

その $\mathcal{A}_{\mathcal{G}}$ の実行 r はゲームでは**プレイ** (play) と呼ばれ、 $\mathcal{G}$ のすべてのプレイの集合を $\text{Plays}^{\mathcal{G}}$ と書く. オートマトンの場合に似て、ゲームのプレイは、プレイヤー 0 とプレイヤー 1 から交互に与えられるアクション (文字) をもとに決定的な遷移関係に従って状態を遷移していくことで進行し、Büchi 受理条件などで与えられた勝利条件を満たすプレイではプレイヤー 0 が**勝利** (プレイヤー 1 は敗北) する. また、勝利条件を満たさないプレイではプレイヤー 1 が勝利 (プレイヤー 0 は敗北) する. この定義では、初期状態 q_{init} は Q_0 の要素であるため、定義 3.62, 定義 3.63, 定義 3.64 より、RS と環境がそれぞれがプレイヤー 0 とプレイヤー 1 に対応するように与えられている.

そして、 ω -オートマトンの場合と同様に、勝利条件 WIN が Büchi/Rabin 受理条件で与えられているときには Büchi/Rabin ゲームなどと呼ぶ. また、オー

⁸状態空間が二部グラフであるオートマトンは、ターンベースオートマトンとも呼ばれる.

トマトンの呼称法と同様に、受理条件が Büchi 条件であることを “B”，受理条件が Rabin 条件であることを “R”，受理条件が多閾値平均利得条件であることを “MTMP” で表し、ゲームを “G” で表記する．例えば，Rabin ゲームは RG と書く．

なお，到達可能関係や強連結成分などは，状態遷移系やオートマトンと同様に定義されるものとする．また，ゲーム $\langle Q_0 \times Q_1, \Sigma_0 \times \Sigma_1, \Delta_0 \times \Delta_1, q_{init}, \text{Win} \rangle$ を単に五つ組 $\langle Q, \Sigma, \Delta, q_{init}, \text{Win} \rangle$ と略記する．

戦略

ゲームのプレイ中，両プレイヤーは自身が勝利するための行動を選択していく．その行動方針のことを**戦略**⁹ (strategy) という．形式的には，以下のよう
に定義される．

定義 3.73 (戦略) $\mathcal{G} = \langle Q, \Sigma, \Delta, q_{init}, \text{Win} \rangle$ をゲームとする．このとき，プレイヤー n ($n \in \{0, 1\}$) の戦略は，三つ組 $\langle M, \mu_n, m_{init} \rangle$ である．ただし，

- M は可算のメモリ空間，
- $\mu_n : Q_n \times M \times \Sigma_n \rightarrow ([0, 1] \times M)$ は出力確率関数，
- 初期メモリ $m_{init} \in M$ は初期メモリである．

また，プレイヤー n が遷移先を決定する任意の状態 $q \in Q_n$ ，とメモリ状態 $m \in M_n$ について以下の条件が成り立っているものとする．

$$\sum_{A \in \Sigma_n} \text{Prob}(\mu_n(q, m, A)) = 1. \quad (3.80)$$

ただし， $\mu_n(q, m, A) = \langle p, m' \rangle$ に関して， $\text{Prob}(\mu_n(q, m, A))$ は p を， $\text{Mem}(\mu_n(q, m, A))$ は m' を与える関数である．

そして， $|M| = 1$ のときメモリレス， $|M| < \infty$ のとき有限メモリ， $|M| = \infty$ のとき無限メモリであるという． ■

ここで，メモリ $m \in M$ はプレイの過去情報を表現したものである．メモリ空間 M_i を $\text{Plays}^{\mathcal{G}}$ のプレフィックスの集合としたときには，プレイ上のすべての過去情報を用いて次の状態遷移確率を決定する無限メモリ戦略と見なすことができる．一方，有限メモリ戦略は，プレイの過去情報を有限個に分類しそれを用

⁹政策 (policy) などとも呼ばれる．

いて次の状態遷移確率を決定する．なお、 $\langle M, \mu_n, m_{\text{init}} \rangle$ がメモリレスである場合には、単に μ_n と書くこととする．またそのとき、 $\mu_n(q, m_{\text{init}}, A) = \langle p, m_{\text{init}} \rangle$ も単に $\mu_n(q, A) = p$ とみなすこととする．

ここで、RS と環境のふるまいと同様に、ゲーム上でそれぞれ特定の戦略に従うプレイヤー 0 とプレイヤー 1 のふるまいは DTMC によってモデル化できる．

定義 3.74 (ゲーム上で両プレイヤーが特定戦略に従う際のふるまい) $\mathcal{G} = \langle Q, \Sigma, \Delta, q_{\text{init}}, \text{Win} \rangle$ をゲーム、 $\text{Str}_0 = \langle M, \mu_0, m_0 \rangle$ をプレイヤー 0 の戦略、 $\text{Str}_1 = \langle M', \mu_1, m'_0 \rangle$ をプレイヤー 1 の戦略とする．

このとき、ゲームのプレイは DTMC

$$\mathcal{D}^{\mathcal{G}, \text{Str}_0, \text{Str}_1} = \langle (Q_0 \times \Sigma_0 \times Q_1) \cup (Q_1 \times \Sigma_1 \times Q_0) \times M \times M', \mathbf{P}, P_{\text{init}}, \emptyset, \Lambda_\emptyset \rangle$$

としてモデル化できる．ただし、

- $q_0 \in Q_0$, $\Delta_0(q_0, a) = q_1$, $\mu_0(q_0, m_1, a) = \langle p, m_2 \rangle$ に対して、

$$\mathbf{P}(\langle \langle q'_1, b, q_0 \rangle, m_1, m' \rangle, \langle \langle q_0, a, q_1 \rangle, m_2, m' \rangle \rangle) = p,$$

- $q_1 \in Q_1$, $\Delta_1(q_1, b) = q_0$, $\mu_1(q_1, m'_1, b) = \langle p, m'_2 \rangle$ に対して、

$$\mathbf{P}(\langle \langle q'_0, a, q_1 \rangle, m, m'_1 \rangle, \langle \langle q_1, b, q_0 \rangle, m, m'_2 \rangle \rangle) = p,$$

- $\Delta_0(q_{\text{init}}, a) = q_1$, $\mu_0(q_{\text{init}}, m_{\text{init}}, a) = \langle p, m \rangle$ に対して、

$$P_{\text{init}}(\langle \langle q_{\text{init}}, a, q_1 \rangle, m, m'_{\text{init}} \rangle) = p,$$

- $\Lambda_\emptyset$ は常に $\emptyset$ を返す関数である． ■

この DTMC のパスは、ゲームの遷移と両プレイヤーのメモリ状態の三つ組の列であるため、メモリ状態を捨象した列の生起確率を考えることで、ゲームのプレイの生起確率を捉えることができる．ここで、プレイ断片 $\delta_1 \dots \delta_n \in (\Delta_0 \Delta_1)^*(\epsilon \cup \Delta_0)$ についての筒集合を以下のように定義する．

定義 3.75 (ゲーム上のプレイの筒集合) $\mathcal{G} = \langle Q, \Sigma, \Delta, q_{\text{init}}, \text{Win} \rangle$ をゲームとする．このとき、プレフィックスが $\delta_1 \dots \delta_n \in (\Delta_0 \Delta_1)^*(\epsilon \cup \Delta_0)$ であるようなプレイの集合をプレイ断片 $\delta_1 \dots \delta_n$ の筒集合といい、 $\text{Cyl}^{\mathcal{G}}(\delta_1 \dots \delta_n)$ と書く． ■

このとき、MC のパス集合上の確率測度と同様に、ゲームのプレイ集合上の確率測度を定義できる．

定義 3.76 (ゲームのプレイ集合上の確率測度) $\mathcal{G} = \langle Q, \Sigma, \Delta, q_{init}, \text{Win} \rangle$ をゲーム, $\text{Str}_0 = \langle M, \mu_0, m_0 \rangle$ をプレイヤ 0 の戦略, $\text{Str}_1 = \langle M', \mu_1, m'_0 \rangle$ をプレイヤ 1 の戦略とする.

このとき, $\mathcal{F}^{\mathcal{G}}$ を $\text{Plays}^{\mathcal{G}}$ に含まれるすべての筒集合を含む最小の完全加法族としたとき, $\mathcal{G}$ 上の戦略 Str_0 に従うプレイヤ 0 と戦略 Str_0 に従うプレイヤ 1 によって生起されるプレイに関する可測空間 $\langle \text{Plays}^{\mathcal{G}}, \mathcal{F}^{\mathcal{G}} \rangle$ 上の確率測度 $\text{Pr}^{\mathcal{G}, \text{Str}_0, \text{Str}_1} : \mathcal{F}^{\mathcal{G}} \rightarrow [0, 1]$ は以下のように定義できる.

$$\begin{aligned} \text{Pr}^{\mathcal{G}, \text{Str}_0, \text{Str}_1}(\text{Cyl}^{\mathcal{G}}(\delta_1 \dots \delta_n)) = \\ \sum_{\substack{m_0, \dots, m_n \in M, \\ m'_0, \dots, m'_n \in M'}} \text{Pr}^{\mathcal{D}^{\mathcal{G}, \text{Str}_0, \text{Str}_1}}(\text{Cyl}^{\mathcal{G}}(\langle \delta_1, m_0, m'_0 \rangle \dots \langle \delta_n, m_n, m'_n \rangle)). \end{aligned} \quad (3.81)$$

■

ここで, プレイヤ 1 の任意の戦略に対してプレイヤ 0 が必ず勝利できる戦略を**勝利戦略** (winning strategy) と呼ぶ.

定義 3.77 (プレイヤ 0 の勝利戦略) $\mathcal{G} = \langle Q, \Sigma, \Delta, q_{init}, \text{Win} \rangle$ をゲームとする. このとき, 以下を満たす Str_0 をプレイヤ 0 の勝利戦略という.

$$\forall \text{Str}_1. \langle \delta_0, m_0, m'_0 \rangle \langle \delta_1, m_1, m'_1 \rangle \dots \in \text{Paths}^{\mathcal{D}^{\mathcal{G}, \text{Str}_0, \text{Str}_1}}. \delta_0 \delta_1 \dots \text{が Win を満たす.} \quad (3.82)$$

また, 以下を満たす Str_0 をプレイヤ 0 のほとんど確かな勝利戦略という.

$$\forall \text{Str}_1. \text{Pr}^{\mathcal{G}, \text{Str}_0, \text{Str}_1}(\{r \in \text{Plays}^{\mathcal{G}} \mid r \text{ は Win を満たす} \}) = 1. \quad (3.83)$$

■

本研究では, 確率的環境を想定した実現可能性 (定義 3.70) を扱うため, ここでもプレイヤ 1 の既知の戦略に対して勝利するための戦略に注目する. そのような特定の敵戦略 Str_1 に対してはほとんど確かに勝利できることが保証されている戦略 Str_0 を Str_1 に対する**対抗戦略** (counter strategy) と呼ぶ.

定義 3.78 (対抗戦略) $\mathcal{G} = \langle Q, \Sigma, \Delta, q_{init}, \text{Win} \rangle$ をゲームとする. このとき, プレイヤ 1 の戦略 Str_1 に対して, 以下を満たす (有限メモリ/メモリレス) 戦略 Str_0 を (有限メモリ/メモリレス) 対抗戦略という.

$$\text{Pr}^{\mathcal{G}, \text{Str}_0, \text{Str}_1}(\{r \in \text{Plays}^{\mathcal{G}} \mid r \text{ は Win を満たす} \}) = 1. \quad (3.84)$$

また, $\mathcal{G}$ 上の μ_1 に対するすべての (有限メモリ/メモリレス) 対抗戦略の集合を $\text{Cnt}_{(\text{fin}/\text{ML})}^{\mathcal{G}, \text{Str}_1}$ と書く. ■

ここで、RS と環境のふるまいと同様に、 $\mathcal{G}$ 上の戦略 Str_0 と Str_1 について、可積分な確率変数 $X : (\Delta_0 \Delta_1)^\omega \rightarrow \mathbb{R}$ に対する期待値 $\text{Expect}^{\mathcal{G}, \text{Str}_0, \text{Str}_1}[X]$ は以下のように与えられる.

$$\text{Expect}^{\mathcal{G}, \text{Str}_0, \text{Str}_1}[X] = \int_{r \in \text{Plays}^{\mathcal{G}}} X(r) \, d\text{Pr}^{\mathcal{G}, \text{Str}_0, \text{Str}_1}(r). \quad (3.85)$$

そのため、そのような確率変数 X で与えられる目的関数に関する最適対抗戦略は以下のように定義できる.

定義 3.79 (最適対抗戦略) $\mathcal{G} = \langle Q, \Sigma, \Delta, q_{\text{init}}, \text{Win} \rangle$ をゲーム, $X : (\Delta_0 \Delta_1)^\omega \rightarrow \mathbb{R}$ を可積分な目的関数とする.

このとき、プレイヤー 1 の戦略 Str_1 に対して、以下を満たす対抗戦略 $\text{Str}_0 \in \text{Cnt}^{\mathcal{G}, \text{Str}_1}$ を X に関する最適対抗戦略という.

$$\text{Expect}^{\mathcal{G}, \text{Str}_0, \text{Str}_1}[X] = \sup_{\text{Str}'_0 \in \text{Cnt}^{\mathcal{G}, \text{Str}_1}} \{ \text{Expect}^{\mathcal{G}, \text{Str}'_0, \text{Str}_1}[X] \}. \quad (3.86)$$

また、 $\varepsilon > 0$ に対して以下を満たす有限メモリ/メモリレスな対抗戦略 $\mu_0 \in \text{Cnt}_{\text{fin/ML}}^{\mathcal{G}, \text{Str}_1}$ を Str_1 に対する、 X に関する有限メモリ/メモリレス (ε -) 最適対抗戦略という.

$$\text{Expect}^{\mathcal{G}, \text{Str}_0, \text{Str}_1}[X] + \varepsilon = \sup_{\mu'_0 \in \text{Cnt}_{\text{fin/ML}}^{\mathcal{G}, \text{Str}_1}} \{ \text{Expect}^{\mathcal{G}, \text{Str}'_0, \text{Str}_1}[X] \}. \quad (3.87)$$

■

第II部

確率頻度時間論理とその検証手法

第4章 確率頻度時間論理PFTL

生命過程や待ち行列などの現実的なシステムはランダム性を持ったモデルとして捉えられることが多く、その定量的な解析にしばしば確率モデル検査が用いられている。これまでに開発されている確率的モデル検査器は、様々な種類の確率的モデルを解析でき、確率的モデル検査は最も成功している定量的形式解析手法の一つである。

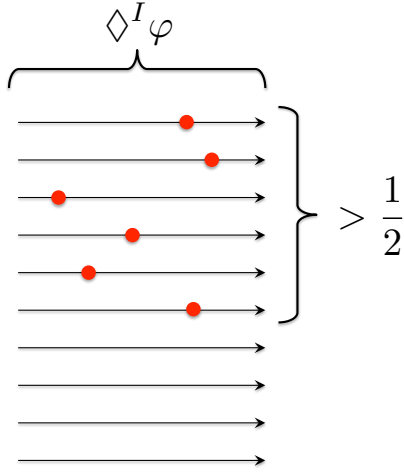
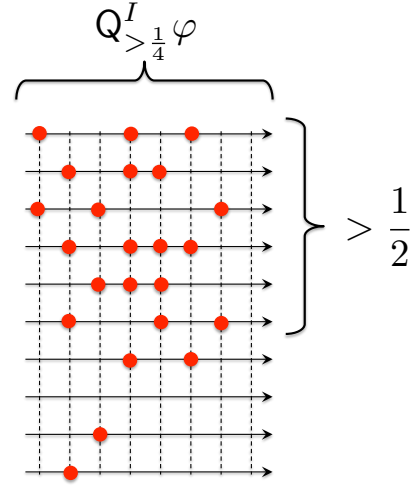
CTLのパス限量子が「すべてのパスで ψ を満たす ($A[\psi]$)」や「あるパスが ψ を満たす ($E[\psi]$)」という分岐の定性的性質を記述可能であったのに対して、確率時間論理の確率パス限量子は「 ψ を満たすパスの生起確率が閾値 p を超える ($P_{>p}[\psi]$)」などの分岐の定量的性質を記述可能である。しかしながら、既存の確率時間論理では、一つの時間系列に関する性質 ψ については、「次に ψ を満たす ($X\psi$)」や「 ψ_2 を満たすまでは常に ψ_1 を満たし続ける ($\psi_1 U \psi_2$)」、「いつか ψ を満たす ($\Diamond\psi$)」、「常に ψ を満たす ($\Box\psi$)」といった性質しか捉うことができない。例えば、

- (1). 「“ある時間区間 I 中いつか φ を満たすパス”の生起確率が50%より大きい (図4.1)」 ($= P_{>1/2}[\Diamond^I \varphi]$) ことは記述できるが、
- (2). 「“ある時間区間 I 中で25%より多い時間点で φ を満たすパス”の生起確率が50%より大きい」という、「ある性質を満たす時間の割合」は記述できない。

これはパス式を記述する演算子が定性的なレベルでしか記述できないためである。また、この(2)に似た性質としては次の式が考えられる。

$$(3). P_{>1/2}[\Box^{I-1} P_{>1/4}[X\varphi]].$$

この式は、「時間区間 $I-1$ の間、“25%より大きい確率で次に φ が成り立つ状態”が常に出現するパス」が50%より大きい確率で生起することを表しているが、そのパス上で実際に φ がどれだけ成り立っているか（「 φ を満たす時間の割合」）については言及できていない。そのため、現実のシステムのふるまいやシミュレーションで得られたパスの揺らぎをもった性質を直観的に記述する

図 4.1: $P_{>\frac{1}{2}}[\Diamond^I \varphi]$.図 4.2: $P_{>\frac{1}{2}}[\mathcal{Q}_{>\frac{1}{4}}^I \varphi]$.

ための言語として適しているとは言えない。

そこで本研究では、既存の確率時間論理に頻度演算子 $\mathcal{Q}$ を新たに導入した確率頻度時間論理 PFTL (Probabilistic Frequency Temporal Logic) を開発した [80]¹。この頻度演算子 $\mathcal{Q}$ は、パス上である性質がどれだけ成り立つかを直接記述できる新しい時間演算子である。つまり、確率時間論理がパス限量子の定量的な拡張であったのに対して、この拡張は時間演算子の定量的な拡張である。この頻度演算子によって、一つの時間系列上の揺らぎをもったふるまいの特性を直観的に記述できる。例えば、 $P_{>1/2}[\mathcal{Q}_{>1/4}^I \varphi]$ によって (2) の性質を記述できる (図 4.2)。

本章ではこの PFTL の構文と意味論の定義を与える。そして、5 章で PFTL のモデル検査手法について述べる。

本章の構成

まず、4.1 節で確率頻度時間論理 PFTL の構文の定義を与える。そして 4.2 節/4.3 節で PFTL の離散時間/連続時間上の意味論を与える。また、4.4 節では、PFTL で記述できる性質の例を示す。

¹本稿で与える意味論は [80] とは若干異なる。

4.1 構文

確率頻度時間論理 PFTL[80] は確率時間論理に頻度演算子 Q を導入した CTL* 型の時間論理である。この頻度演算子によって、揺らぎをもったふるまいの特性を直観的に記述できる。例えば、 $Q_{\geq 0.8}^{[0,10]}\varphi$ は「10 ステップ中で φ が成り立つ頻度が 80% 以上である（8 個以上の時刻で φ が成り立つ）」ことを表す。また、利便性も考慮して、この演算子は条件付き頻度を記述できる二項演算子として定義する。

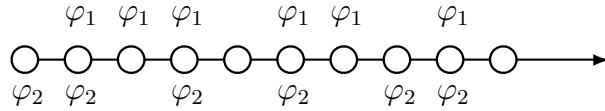
形式的には、以下のように定義される。

定義 4.1 (PFTL の構文) AP を原子命題の集合とする。このとき、PFTL の式 φ は以下のように帰納的に定義される。

$$\begin{aligned}\varphi &::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid P_{\sim c}[\psi], \\ \psi &::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid X^I\psi \mid \psi U^I\psi \mid Q_{\sim c}^I[\psi|\psi].\end{aligned}$$

ただし、 $p \in AP$, $c \in [0, 1]$, $\sim \in \{<, >, \leq, \geq\}$, $I \subseteq \mathbb{R}_{\geq 0}$ とする。 ■

直観的には、 $Q_{\sim c}^I[\psi_1|\psi_2]$ は「時間区間 I 上で、 ψ_2 が成り立っている条件の下で ψ_1 も成り立つ条件付き頻度が閾値条件 $\sim c$ を満たす」ことを表す。例えば、以下のような時間系列において、「最初の 10 ステップの条件 φ_2 の下で φ_1 が成り立つ条件付き頻度」は $2/3$ である。



また、PFTL では、LTL や PCTL, CSL での略記のほか、以下のような略記も用いてよいことにする。

$$Q_{\sim c}^I\psi \equiv Q_{\sim c}^I[\psi|\top].$$

直観的には、 $Q_{\sim c}^I\psi$ は、「時間区間 I 中の ψ の条件なし頻度が閾値条件 $\sim c$ を満たす」ことを表す。

4.2 離散時間意味論

本節では PFTL の離散時間での意味論を与える。

これは PCTL の意味論と同様に DTMC 上の性質として定義される。ただし、PFTL は CTL* 型の時間論理であるためにパス式に関する意味が LTL のように与えられる。

新たに導入された頻度演算子の意味を与える前に、まず条件 ψ_2 のもとで ψ_1 を満たす条件付き頻度の定義を与える。離散時間上においては、注目している時間区間 I が有界かつその中に ψ_2 を満たす時刻が存在するならば、その条件付き頻度は「時間区間 I 中の $\psi_1 \wedge \psi_2$ を満たす時刻数」の「時間区間 I 中の ψ_2 を満たす時刻数」に対する比として定義できる。一方、時間区間 I 中に ψ_2 を満たす時刻が存在しない場合には条件付き頻度は定義できない。

定義 4.2 (離散時間上の有界時間区間における条件付き頻度) 有界時間区間 J に対して、条件 ψ_2 のもとで ψ_1 を満たす条件付き頻度 $\text{Freq}_\rho^{\mathcal{D}}(\psi_1, \psi_2, J)$ は以下のように定義される。

$$\text{Freq}_\rho^{\mathcal{D}}(\psi_1, \psi_2, J) = \begin{cases} \frac{|\{k \in J \mid \mathcal{D}, \rho[k \dots] \models \psi_1 \wedge \psi_2\}|}{|\{k \in J \mid \mathcal{D}, \rho[k \dots] \models \psi_2\}|} & \text{if } \exists k \in J. \mathcal{D}, \rho[k \dots] \models \psi_2, \\ \text{undefined} & \\ \text{otherwise.} & \end{cases} \quad (4.1)$$

■

なお、時間区間 I が非有界である場合、時間区間 $I \cap [0, j]$ における条件付き頻度は $j \rightarrow \infty$ のとしたときに収束するとは限らない。そこで、時間区間 I が非有界である場合の頻度演算子の意味は、上極限/下極限を用いて定義する。

定義 4.3 (PFTL の離散時間意味論) $\mathcal{D} = \langle S, \mathbf{P}, P_{\text{init}}, AP, \Lambda \rangle$ を DTMC とする。このとき、 $\mathcal{D}$ と状態式の充足関係は定義 3.37 と同様であり、 $\mathcal{D}$ とパス式

の充足関係 $\models$ は以下のように帰納的に定義される².

$$\begin{aligned}
\mathcal{D}, \rho \models \varphi &\Leftrightarrow \mathcal{D}, \rho[0] \models \varphi \text{ for a state formula } \varphi, \\
\mathcal{D}, \rho \models \neg\psi &\Leftrightarrow \mathcal{D}, \rho \not\models \psi, \\
\mathcal{D}, \rho \models \psi_1 \vee \psi_2 &\Leftrightarrow \mathcal{D}, \rho \models \psi_1 \text{ or } \mathcal{D}, \rho \models \psi_2, \\
\mathcal{D}, \rho \models X\psi &\Leftrightarrow \mathcal{D}, \rho[1 \dots] \models \psi \\
\mathcal{D}, \rho \models \psi_1 \mathbf{U}^I \psi_2 &\Leftrightarrow \exists i \in I. (\mathcal{D}, \rho[i \dots] \models \psi_2 \text{ and } \forall j < i. \mathcal{D}, \rho[j \dots] \models \psi_1), \\
\mathcal{D}, \rho \models \mathbf{Q}_{\sim c}^I[\psi_1 | \psi_2] &\Leftrightarrow \begin{cases} (\exists k \in I. \mathcal{D}, \rho[k \dots] \models \psi_2) \Rightarrow \text{Freq}_{\rho}^{\mathcal{D}}(\psi_1, \psi_2, I) \sim c \\ \text{if } \sup I < \infty, \\ (\forall j > \inf I. \exists k > j. \mathcal{D}, \rho[k \dots] \models \psi_2) \Rightarrow \\ \left[\liminf_{j \rightarrow \infty} \text{Freq}_{\rho}^{\mathcal{D}}(\psi_1, \psi_2, I \cap [0, j]), \right. \\ \left. \limsup_{j \rightarrow \infty} \text{Freq}_{\rho}^{\mathcal{D}}(\psi_1, \psi_2, I \cap [0, j]) \right] \subseteq \{x \mid x \sim c\} \\ \text{otherwise.} \end{cases}
\end{aligned}$$

ただし、 I は $\mathbb{N}$ 上の区間とする. ■

有界時間区間 J ($\sup J < \infty$) 上の頻度は単純に定義できるが、非有界時間区間 J' ($\sup J' = \infty$) 上での頻度の捉え方にはいくつかの選択がある. 定義 4.3 では、非有界時間区間上の頻度は有限プレフィックスの頻度の極限である. そして、その頻度の定義のもとで、条件付き頻度を「条件式 ψ_2 が無限にしばしば成り立つ」という前提のもとで定義している. 実際には「ある $j \in J'$ が存在し $\mathcal{D}, \rho[j \dots] \models \psi_2$ 」という前提が満たされれば条件付き頻度は定義できる³. この場合、いつか常に ψ_2 を満たさないパスでは、非有界時間区間上の頻度が特定の有限区間上の頻度のみで決定される. ここで、 $\mathcal{D}$ が有限状態であれば、ほとんどすべてのパスがいつか何れかの BSCC に到達し、そこに含まれる状態のみを無限にしばしば訪れる. そして、その BSCC の定常状態分布に従った頻度をもつため、「無限にしばしば ψ_2 を満たすこと」と「条件式 ψ_2 を満たす状態へ到達すること」はほとんどすべてのパス上で一致する. 本研究では、(1) 解析が容易であり、(2) ほとんどすべてのパスのほとんどすべての時間上の性質を反映するという理由から、この定義 4.3 を採用する.

また、非有界時間区間 I ($\sup I = \infty$) 上の条件付き頻度式 $\mathbf{Q}_{\sim c}^I[\psi_1 | \psi_2]$ に対

²時間非有界な Q 式の意味は [80] とは異なる.

³[80] では、この定義を基に時間非有界な Q 式の意味を定義している.

して、極限を用いない以下のような定義を与えることも可能である。

$$\mathcal{D}, \rho \models Q_{\sim c}^I[\psi_1 | \psi_2] \Leftrightarrow \left(\forall j > i. \exists k > j. \mathcal{D}, \rho[k \dots] \models \psi_2 \Rightarrow \right. \\ \left. \exists j > i. \forall k > j. \text{Freq}_\rho^{\mathcal{D}}(\psi_1, \psi_2, I \cap [0, k]) \sim c \right). \quad (4.2)$$

これは「条件式 ψ_2 が無限にしばしば成り立つという前提のもとで、充分長いすべての時間区間の条件付き頻度が閾値を満たす」という定義である。ここでも前提条件の設定にはいくつか選択肢があるが、どのような場合でも解析が非常に困難である。特に、 $Q_{\sim c} Q_{\sim c'} \psi$ のように非有界時間頻度式を重ねた式は、その意味を直観的に理解することが非常に困難である。そのため、この定義は採用していない。

4.3 連続時間意味論

本節では PFTL の連続時間での意味論を与える。

ここでは CSL の意味論と同様に、非爆発的 CTMC の性質として意味を与える。連続時間意味論も離散時間の場合と同様に定義されるが、有限時間区間 I ($\inf I \neq \sup I$) 上の条件式 ψ_2 のもとで式 φ_1 が成り立つ条件付き頻度の捉え方にはいくつか選択肢がある。一般に、連続時間上の「時間の長さ」は Lubesgue 測度 $\mathfrak{L}$ を用いて捉えることができる。そのため、有界時間区間 I 中で ψ_2 が成り立つ時間の長さが Lubesgue 測度で 0 より大きい場合には、「時間区間 I 中の $\psi_1 \wedge \psi_2$ を満たす時間の Lubesgue 測度での長さ」の「時間区間 I 中の ψ_2 を満たす時間の Lubesgue 測度での長さ」に対する比として定義できる。これは、非爆発的 CTMC のほとんどすべてのパス上では任意の有界時間区間中には有限回の遷移しか生起しないためである。この Lubesgue 測度のみをもとにして条件付き頻度を定義することもできるが、 ψ_2 が成り立つ時間の長さが Lubesgue 測度で 0 であっても、 ψ_2 が有限個の時刻で成り立つ場合もあり得る。つまり、 ψ_2 を満たす「時間の長さ」が Lubesgue 測度で 0 の場合でも、 $\psi_1 \wedge \psi_2$ を満たす時刻数の ψ_2 を満たす時刻数に対する比を条件付き頻度として定義できる。ここで、任意の有限時間区間中には有限回の遷移しか生起しないため、そのような時刻数は有限でしかない。そこで本研究では、 ψ_2 の成り立つ時間の長さが Lubesgue 測度 $\mathfrak{L}$ で 0 より大きい場合には Lubesgue 測度 $\mathfrak{L}$ 、Lubesgue 測度 $\mathfrak{L}$ で 0 であってもそのような時刻が存在する場合には時刻数をもとに条件付き頻度を定義する。

定義 4.4 (連続時間上の有界時間区間における条件付き頻度) 有界時間区間 J について, 条件 ψ_2 のもとで ψ_1 を満たす条件付き頻度 $\text{Freq}_\rho^C(\psi_1, \psi_2, J)$ は以下のように定義される.

$$\text{Freq}_\rho^C(\psi_1, \psi_2, J) = \begin{cases} \frac{\mathcal{L}(\{k \in J \mid \mathcal{C}, \rho@[k \dots] \models \psi_1 \wedge \psi_2\})}{\mathcal{L}(\{k \in J \mid \mathcal{C}, \rho@[k \dots] \models \psi_2\})} & \text{if } \mathcal{L}(\{k \in J \mid \mathcal{C}, \rho@[k \dots] \models \psi_2\}) > 0, \\ \frac{|\{k \in J \mid \mathcal{C}, \rho@[k \dots] \models \psi_1 \wedge \psi_2\}|}{|\{k \in J \mid \mathcal{C}, \rho@[k \dots] \models \psi_2\}|} & \text{if } 0 < \{k \in J \mid \mathcal{C}, \rho@[k \dots] \models \psi_2\} < \infty, \\ \text{undefined} & \\ \text{otherwise.} & \end{cases} \quad (4.3)$$

■

なお, 非有界時間区間上の条件付き頻度は, 離散時間意味論と同様に, 「条件式 ψ_2 が無限にしばしば成り立つ」という前提のもとで頻度を定義する.

定義 4.5 (PFTL の連続時間意味論) $\mathcal{C} = \langle S, R, P_{init}, AP, \Lambda \rangle$ を CTMC とする. このとき, $\mathcal{C}$ と状態式の充足関係は定義 3.39 と同様であり, $\mathcal{C}$ とパス式の充足関係 $\models$ は以下のように帰納的に定義される⁴.

$$\begin{aligned} \mathcal{C}, \rho &\models \varphi &\Leftrightarrow \mathcal{C}, \rho[0] &\models \varphi \text{ for a state formula } \varphi, \\ \mathcal{C}, \rho &\models \neg \psi &\Leftrightarrow \mathcal{C}, \rho &\not\models \psi, \\ \mathcal{C}, \rho &\models \psi_1 \vee \psi_2 &\Leftrightarrow \mathcal{C}, \rho &\models \psi_1 \text{ or } \mathcal{C}, \rho \models \psi_2, \\ \mathcal{C}, \rho &\models X^I \psi &\Leftrightarrow \text{time}(\rho, 0) \in I &\text{ and } \mathcal{C}, \rho@[time(\rho, 0) \dots] \models \psi, \\ \mathcal{C}, \rho &\models \psi_1 U^I \psi_2 &\Leftrightarrow \exists t_1 \in I. (\mathcal{C}, \rho@[t_1 \dots] &\models \psi_2 \text{ and } \forall t_2 < t_1. \mathcal{C}, \rho@[t_2 \dots] \models \psi_1), \\ \mathcal{C}, \rho &\models Q_{\sim c}^I[\psi_1 | \psi_2] &\Leftrightarrow \begin{cases} (\exists k \in I. \mathcal{C}, \rho@[k \dots] \models \psi_2) \Rightarrow \text{Freq}_\rho^C(\psi_1, \psi_2, I) \sim c \\ \text{if } \sup I < \infty, \\ (\forall j > \inf I. \exists k > j. \mathcal{C}, \rho@[k \dots] \models \psi_2) \Rightarrow \\ \left[\liminf_{j \rightarrow \infty} \text{Freq}_\rho^C(\psi_1, \psi_2, I \cap [0, j]), \right. \\ \left. \limsup_{j \rightarrow \infty} \text{Freq}_\rho^C(\psi_1, \psi_2, I \cap [0, j]) \right] \subseteq \{x \mid x \sim c\} \\ \text{otherwise.} \end{cases} \end{aligned}$$

ただし, I は $\mathbb{R}_{>0}$ 上の区間とする.

■

⁴離散時間野場合同様に, 時間非有界な Q 式の意味は [80] とは異なる.

ここで、任意の式について無限時間長の連続時間パス上でその式を満たす時間の集合は Lebesgue 可測であり、また、Lebesgue 測度が0となる集合 $\{k \in I \mid \mathcal{C}, \rho @ [k \dots] \models \psi\}$ の大きさが無限である場合を考える必要はない。なぜならば、本研究では CTMC は非爆発的であると仮定しており、 ψ の部分式の真偽が有界時間区間内では有限回しか変化せず、有界時間区間 I 中の ψ を満たす時刻の集合は有限個の（幅が0の）時間区間の和集合からなる。

また、PFTL パス式 ψ の時間制約上界 $\#(\psi)$ は、式 (3.56) と同様に、以下の場合を追加したものとして定義される。

$$\#(\psi) = \max\{\#(\psi_1), \#(\psi_2)\} + \sup I \quad \text{if } \psi = Q_{\sim c}^I \langle \psi_1 \mid \psi_2 \rangle. \quad (4.4)$$

注意 4.6 (PFTL の頻度演算子と CSL の定常状態分布演算子) PFTL の頻度式 $Q_{\sim c} \varphi$ は、「パス全体の φ を満たす時間の頻度」に関する性質、すなわち、「パス上の定常状態分布」に関する性質を記述するパス式である。一方、CSL の定常状態分布演算子 S (注意 3.42) も定常状態分布に関する性質を記述できるが、この演算子で記述される式は「定常状態分布の期待値」に関する性質を記述する状態式である。 ■

4.4 仕様記述例

本節では、PFTL で記述できる性質の例を、既存時間論理により記述可能な性質と比較しながら示す。

条件なし頻度

本項では、時間論理の様相演算子として広く用いられている $\Diamond$ 演算子及び $\Box$ 演算子により表現可能な性質と PFTL の条件なし頻度演算式により表される性質の関係について述べる。

until 演算子の略記として定義される $\Diamond$ 演算子と $\Box$ 演算子は、「いつか（ある時間で）…」や「常に（すべての時間で）…」という定性的線形時間性質のみを表現可能である。一方、PFTL で新たに導入された演算子 $Q_{\sim c}$ は「頻度閾値条件 $\sim c$ を満たす割合の時間で…」という定量的線形時間性質を表現可能である。ここで、PFTL の意味論における頻度の定義（定義 4.2 と定義 4.4）では、「すべての時刻で ψ を満たす」（「 ψ を満たす時間の割合が0より大きい」）ならば「 ψ を満たす時間の割合が1である」（「ある時刻で ψ を満たす」）であ

るが、その逆は必ずしも成り立たない。即ち、条件なし頻度式 $Q_{\sim c}\psi$ に関して、古典的時間演算子 $\Diamond$ と $\Box$ を用いた式との間には以下の含意関係が成り立つ。

$$Q_{>c}^I\psi \rightarrow \Diamond^I\psi \quad (4.5)$$

$$\Box^I\psi \rightarrow Q_{\geq c}^I\psi \quad (4.6)$$

$$Q_{>c}\psi \rightarrow \Box\Diamond\psi \quad (4.7)$$

$$\Diamond\Box\psi \rightarrow Q_{\geq c}\psi \quad (4.8)$$

以上のように、条件なし頻度式を用いることで、 $\Diamond$ と $\Box$ の組み合わせでは表現できない強さの性質を記述可能である。

定量的相関関係

本節では、頻度演算子 Q と確率パス限量子 P が表現する性質の違いについて述べる。

φ を原子命題の Boolean 結合とし、以下のような PFTL パス式を考える。

$$Q_{\sim c}^I(\psi \mid \varphi). \quad (4.9)$$

この PFTL パス式は「時間区間 I 中で条件 φ のもとで ψ も満たす条件付き頻度が閾値条件 $\sim c$ を満たす」という、パス上の φ と ψ の定量的相関関係を表している。PFTL パス式 (4.9) と似た線形時間的性質は、CSL では例えば以下のようなパス式で記述できる。

$$\Box^I(\varphi \rightarrow P_{\sim c}[\psi]). \quad (4.10)$$

この CSL パス式 (4.10) は、「パス上の時間区間 I 中で φ を満たすすべての状態について、その状態を始点として“ ψ を満たすパスに分岐する確率”が閾値条件 $\sim c$ を満たす」というパス上に現れる状態における φ と $P_{\sim c}[\psi]$ の定性的相関関係を表す。なお、 I が非有界であるときには、PFTL パス式 (4.9) は、「 φ であるすべての状態 s について、“ s を始点として ψ を満たすパスに分岐する確率”を“ s の出現頻度”で重み付けした加重平均的な確率」、即ち、「 φ を満たす状態における平均的な“ ψ を満たすパスに分岐する確率”が閾値条件 $\sim c$ を満たす」ことを表していると解釈できる。

また、CSL では、式 (4.10) の部分式 φ の代わりに時間演算子を用いたパス式を用いることは構文上許されていない。一方で、PFTL では、式 (4.9) の部分式 φ の代わりにパス式を用いてもよい。そのため、頻度演算子 Q を用いる

ことで、二つのパス式が表現する線形時間的性質の間の定量的相関関係を表現できる。例えば、以下のようなパス式は「“5 単位時間以内に φ が成り立つ時間” の 8 割以上で ψ が成り立ち、かつ、“5 単位時間以内に φ が成り立たない時間” の 2 割以下でしか ψ は成り立たない」ことを表す。

$$Q_{\geq 0.8} \langle \psi \mid \Diamond^{\leq 5} \varphi \rangle \wedge Q_{\leq 0.2} \langle \psi \mid \neg \Diamond^{\leq 5} \varphi \rangle. \quad (4.11)$$

確率的周期性

頻度演算子 Q を用いることで、確率的周期的のような複雑な時間的性質も簡潔に表現することができる。

例えば、以下のような PFTL パス式を考える。

$$Q_{\sim q}^I \langle \Diamond^{\leq 1} \Delta^J \psi_1 \mid \Delta^{\leq 1} \psi_2 \rangle. \quad (4.12)$$

この式は「時間区間 I 中では、丁度 1 単位時間後に初めて ψ_2 が成り立つとき、丁度 1 単位時間後からみて ψ_1 が初めて成り立つ時間が J に含まれる頻度が閾値条件 $\sim q$ を満たす」ことを表す。これは、 ψ_1 の立ち上がり間隔が 1 単位時間より長いパス上では、「 ψ_2 の立ち上がり後に ψ_1 が立ち上がるまでの時間が J に含まれる頻度」に関する性質を表している。前項でも述べたように、条件付き頻度演算子 Q を用いることで、このような線形時間的性質に注目した平均的な分岐の可能性を直観的に捉えることができる。そして、 $\psi_2 = \psi_1$ のとき、この式は「 ψ_1 の生起間隔」に関する定量的な性質を表している。そのため、頻度演算子 Q を用いることで、「 ψ_1 で特徴付けられる時間的性質が成り立つ周期」に関する定量的性質を記述することができる。

第5章 PFTLモデル検査

モデル検査は、システムを抽象化したモデルが仕様を満たすか網羅的に自動検証する手法である。無限個の状態を網羅的に検証することが一般には困難であるため、通常の場合、検査の対象は有限状態モデルである。時間演算子の直前にパス限量子を置かなくてはならないCTL型の時間論理はモデル検査が容易である場合が多い。一方で、LTLなどの時間演算子を複数用いてパス式を記述できる時間論理では、多くの場合、モデル検査は充足可能性判定と同程度に難しいことが知られている。

CTLの確率的拡張論理であるPCTL/CSLのモデル検査では、離散時間/連続時間Markov連鎖（DTMC/CTMC）などの完全に確率的にふるまうモデルだけでなく、Markov決定過程（MDP）や確率時間オートマトンなどの一部に非決定的選択をもつモデルも検査対象にできる[58]。また、PRISM[70, 58]などの確率的モデル検査器では、状態式 $P_{\sim c}[\psi]$ を入力としたモデル検査問題だけでなく、 $P_{=?}[\psi]$ という特殊な形の質問式を入力とした**充足確率問題**（satisfaction probability problem）を解くことができる¹。

- 確率的モデル検査問題

- 入力：モデル M と検査式 $P_{\sim c}[\psi]$
- 出力：「 $M \models P_{\sim c}[\psi]$ か否か」 ($\in \{\top, \perp\}$)

- 充足確率問題

- 入力：モデル M と質問式 $P_{=?}[\psi]$ (ψ はパス式)
- 出力：確率 $\Pr^M(\{\rho \in \text{Paths}^M \mid M, \rho \models \psi\})$ ($\in [0, 1]$)

充足確率問題は、モデルの確率に関する性能解析問題の一種であり、モデルの定量的性能評価に用いることができる。

¹なお、LTLなどの線形時間論理の確率的モデル検査は、式 ψ に対して検査式 $P_{\sim c}[\psi]$ または質問式 $P_{=?}[\psi]$ を入力とするモデル検査のことを指す。

本章では、PFTL モデル検査手法について、CTL 型フラグメント PFCTL のモデル検査と、LTL 型フラグメント PFLTL のモデル検査に分けて与える [80, 96]. PFCTL は、DTMC と CTMC のどちらに対しても既存の CTL 型確率時間論理のモデル検査手法 [51, 11, 12, 59] と同様に、行列/ベクトル演算を中心とした数値計算アプローチで厳密にモデル検査ができる. 一方、PFLTL モデル検査は、DTMC に対しては確率的 LTL モデル検査 [40, 41, 24] に帰着することで厳密に検証することができる. すなわち、離散時間 PFTL モデル検査が決定可能である. これは、離散時間上では以下の二点が成り立つためである.

- 離散時間上の有限区間の頻度に言及する式は、next 演算子を用いて表現できる.
- 時間が離散的であるため、状態遷移と時間変化の粒度が等しい.

しかしながら、連続時間上ではこのような性質は成り立たないため、CTMC に対しては厳密にモデル検査することは困難である. そこで本章では、CTMC に対する PFLTL モデル検査については統計的アプローチ [92] による近似的な時間有界モデル検査手法を与える.

本章の構成

5.1 節と 5.2 節で、それぞれ PFCTL モデル検査と PFLTL 型モデル検査について述べる. そして、5.3 節で CTMC モデルに対する統計的手法を用いた LTL 型フラグメントモデル検査の適用例を示す.

5.1 CTL 型フラグメントモデル検査

本節では、有限状態 DTMC/CTMC モデルに対する、PFTL の CTL 型フラグメントモデル検査について述べる.

CTL 型の構文とは、時間演算子の直前にはパス限量子を置かなくてはならない構文である. PFTL の CTL 型フラグメントを PFCTL といい、以下のよう定義される.

定義 5.1 (PFCTL の構文) AP を原子命題の集合とする。このとき、PFCTL の式 φ は、以下のように帰納的に定義される。

$$\begin{aligned}\varphi &::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid P_{\sim c}[\psi] \\ \psi &::= X^I\varphi \mid \varphi U^I\varphi \mid Q_{\sim c}^I[\varphi|\varphi].\end{aligned}$$

ただし、 $p \in AP$, $c \in [0, 1]$, $\sim \in \{<, >, \leq, \geq\}$, I は自然数上の区間（離散時間）あるいは非負実数の区間（連続時間）とする。 ■

一般に CTL 型時間論理のモデル検査は、状態式 φ を充足する状態の集合 $\text{Sat}(\varphi)$ を、 φ の部分状態式を満たす状態の集合から再帰的に計算していくことで決定される。確率時間論理では、状態式 φ が $P_{\sim c}[\psi]$ の形であるときには、それぞれの状態 s を初期状態として ψ を満たすパスが生起する確率 $\text{Prob}_s^D(\psi)$ を計算し、それが閾値条件 $\sim c$ を満たすか判定する。離散時間/連続時間 PFCTL モデル検査も、PCTL/CSL[51, 11, 12] などの CTL 型確率時間論理モデル検査[51, 11, 12, 59] と同様に、充足集合 $\text{Sat}(\varphi)$ は以下のように定義される。

$$\begin{aligned}\text{Sat}(p) &= \{s \in S \mid p \in \Lambda(s)\}, \\ \text{Sat}(\neg\varphi) &= S \setminus \text{Sat}(\varphi), \\ \text{Sat}(\varphi_1 \wedge \varphi_2) &= \text{Sat}(\varphi_1) \cap \text{Sat}(\varphi_2), \\ \text{Sat}(P_{\sim c}[\psi]) &= \{s \in S \mid \text{Prob}_s^{D/c}(\psi) \sim c\}.\end{aligned}$$

そして、部分状態式の充足集合 $\text{Sat}(\varphi_1)$ と $\text{Sat}(\varphi_2)$ を決定した後に、充足確率ベクトル $\overrightarrow{\text{Prob}}^{D/c}(X^I\varphi_1)$, $\overrightarrow{\text{Prob}}^{D/c}(\varphi_1 U^I\varphi_2)$, $\overrightarrow{\text{Prob}}^{D/c}(Q_{\sim c}^I[\varphi_1|\varphi_2])$ を計算することで再帰的手続きを進めていくことができる。

有限状態 DTMC/CTMC に対するこれらの充足確率ベクトルの計算法をそれぞれ 5.1.1 項と 5.1.2 項に示す。

5.1.1 DTMC に対するモデル検査

本項では、有限状態 DTMC $\mathcal{D} = \langle A, \mathbf{P}, P_{init}, AP, \Lambda \rangle$ に対して、充足確率ベクトル $\overrightarrow{\text{Prob}}^D(X\varphi_1)$, $\overrightarrow{\text{Prob}}^D(\varphi_1 U^I\varphi_2)$, $\overrightarrow{\text{Prob}}^D(Q_{\sim c}^I[\varphi_1|\varphi_2])$ の計算手法 [80] を示す。ここで、自然数区間 I については $[t, t']$ あるいは $[t, \infty)$ という形であると仮定する。任意の離散時間区間はこれらのうちのどちらかの形で表現できるため、一般性は失われない。

$X\varphi_1$ の充足確率ベクトルの計算

この場合、遷移確率行列 $\mathbf{P}$ を用いて以下のように単純に計算できる.

$$\overrightarrow{\text{Prob}}^{\mathcal{D}}(X\varphi_1) = \mathbf{P} \cdot \vec{v}(\varphi_1). \quad (5.1)$$

ただし,

$$v_s(\varphi_1) = \begin{cases} 1 & \text{if } \text{Sat}(\varphi_1), \\ 0 & \text{otherwise.} \end{cases} \quad (5.2)$$

$\varphi_1 \mathbf{U}^I \varphi_2$ の充足確率ベクトルの計算

ここでは四つの場合に分けて考える.

- $I = [0, \infty)$ のとき, 求めるべき充足確率ベクトルは, $\mathcal{D}$ 上でいつか φ_2 を満たし, かつ, それまでの間 φ_1 を満たしながら遷移し続ける時間非有界な条件付き到達確率ベクトルである. よって以下のように計算できる.

$$\overrightarrow{\text{Prob}}(\varphi_1 \mathbf{U}^{[0, \infty)} \varphi_2) = \overrightarrow{\text{Reach}}^{\mathcal{D}[\mathbf{P}/\mathbf{P}_{\varphi_1 \mathbf{U} \varphi_2}]}(\text{Sat}(\varphi_2)). \quad (5.3)$$

ここで, $\mathbf{P}_{\varphi_1 \mathbf{U} \varphi_2}$ は以下のように定義される, いつか φ_2 を満たしそれまでの間 φ_1 を満たしながら遷移し続けるための条件付き遷移確率を表す行列である.

$$\mathbf{P}_{\varphi_1 \mathbf{U} \varphi_2}(s, s') = \begin{cases} 1 & \text{if } s = s' \text{ and } s \in \text{Sat}(\varphi_2), \\ \mathbf{P}(s, s') & \text{if } s \in \text{Sat}(\varphi_1) \setminus \text{Sat}(\varphi_2) \\ & \text{and } s' \in \text{Sat}(\varphi_2), \\ 0 & \text{otherwise.} \end{cases} \quad (5.4)$$

- $I = [0, t]$ のとき, 求める充足確率ベクトルは, 時間区間 $[0, t]$ 以内に φ_2 を満たしそれまでの間 φ_1 を満たしながら遷移し続ける時間有界な条件付き到達確率ベクトルである. これは式 (5.2) と (5.4) を用いて以下のように計算できる.

$$\overrightarrow{\text{Prob}}(\varphi_1 \mathbf{U}^{[0, t]} \varphi_2) = (\mathbf{P}_{\varphi_1 \mathbf{U} \varphi_2})^t \cdot \vec{v}(\varphi_2). \quad (5.5)$$

- $I = [t, t']$ ($t > 0$) のとき, 求める充足確率ベクトルは, $\mathcal{D}$ 上で最初の $t-1$ 時間までは常に φ_1 をみたし, その後の $t'-t$ 時間に φ_1 を満たしながら φ_2 を満たす状態への時間有界な条件付き到達確率である. よって以下のように計算できる.

$$\overrightarrow{\text{Prob}}(\varphi_1 \mathbf{U}^{[t, t']} \varphi_2) = (\mathbf{P}_{\varphi_1})^t \cdot \overrightarrow{\text{Prob}}^{\mathcal{D}}(\varphi_1 \mathbf{U}^{[0, t'-t]} \varphi_2). \quad (5.6)$$

ここで,

$$\mathbf{P}_{\varphi_1}(s, s') = \begin{cases} \mathbf{P}(s, s') & \text{if } s \in \text{Sat}(\varphi_1), \\ 0 & \text{otherwise.} \end{cases} \quad (5.7)$$

- $I = [t, \infty)$ のときは, $I = [t, t']$ の場合と同様に以下のように計算できる.

$$\overrightarrow{\text{Prob}}(\varphi_1 \mathbf{U}^{[t, \infty)} \varphi_2) = (\mathbf{P}_{\varphi_1})^t \cdot \overrightarrow{\text{Prob}}(\varphi_1 \mathbf{U}^{[0, \infty)} \varphi_2). \quad (5.8)$$

$\mathbf{Q}_{\sim c}^I[\varphi_1 | \varphi_2]$ の充足確率ベクトルの計算

- $I = [t, \infty)$ のとき, ほとんどすべてのパスはいずれかの BSCC B に到達し, その状態出現頻度は定常状態分布 $\overrightarrow{\text{Freq}}^{\mathcal{D}}(B)$ と一致する. よって, 以下の式 (5.9) または (5.10) を満たす BSCC に含まれる状態から生起するほとんどすべてのパスはこの頻度式を満たす.

$$B_i \cap \text{Sat}(\varphi_2) = \emptyset, \quad (5.9)$$

$$\frac{\sum_{s' \in B_i \cap \text{Sat}(\varphi_1) \cap \text{Sat}(\varphi_2)} \text{Freq}_{s'}^{\mathcal{D}}(B_i)}{\sum_{s' \in B_i \cap \text{Sat}(\varphi_2)} \text{Freq}_{s'}^{\mathcal{D}}(B_i)} \sim c. \quad (5.10)$$

そのような BSCC B_i の和集合を $\text{SatBSCC}^{\mathcal{D}}(\mathbf{Q}_{\sim c}^I[\varphi_1 | \varphi_2])$ とする. このとき, 求めるべき確率ベクトルは, $\text{SatBSCC}^{\mathcal{D}}(\mathbf{Q}_{\sim c}^I[\varphi_1 | \varphi_2])$ への到達確率として得ることができる. 従って,

$$\overrightarrow{\text{Prob}}^{\mathcal{D}}(\mathbf{Q}_{\sim c}^{[t, \infty)}[\varphi_1 | \varphi_2]) = \overrightarrow{\text{Reach}}^{\mathcal{D}}(\text{SatBSCC}^{\mathcal{D}}(\mathbf{Q}_{\sim c}^I[\varphi_1 | \varphi_2])). \quad (5.11)$$

- $I = [0, t]$ のとき, それぞれの状態を始点として k ステップで φ_2 と $\varphi_1 \wedge \varphi_2$ を満たす状態数をそれぞれ i 回と j 回訪れるパスの生起確率を与えるべ

クトル $\vec{v}^{\varphi_1|\varphi_2}(i, j, k)$ (式 (5.13)-(5.16)) を用いて目的の充足確率ベクトルの計算を以下のように計算できる.

$$\overrightarrow{\text{Prob}}^{\mathcal{D}}(\mathbf{Q}_{\sim_c}^{[0,t]}[\varphi_1|\varphi_2]) = \sum_{i \leq t} \left(\sum_{\substack{j \leq i, \\ j \sim_c i}} \vec{v}^{\varphi_1|\varphi_2}(i, j, t) \right). \quad (5.12)$$

ただし,

$$v_s^{\varphi_1|\varphi_2}(0, 0, 0) = \begin{cases} 1 & \text{if } s \notin \text{Sat}(\varphi_2), \\ 0 & \text{otherwise,} \end{cases} \quad (5.13)$$

$$v_s^{\varphi_1|\varphi_2}(1, 0, 0) = \begin{cases} 1 & \text{if } s \in \text{Sat}(\varphi_2) \setminus \text{Sat}(\varphi_1), \\ 0 & \text{otherwise,} \end{cases} \quad (5.14)$$

$$v_s^{\varphi_1|\varphi_2}(1, 1, 0) = \begin{cases} 1 & \text{if } s \in \text{Sat}(\varphi_2) \cap \text{Sat}(\varphi_1), \\ 0 & \text{otherwise,} \end{cases} \quad (5.15)$$

$$v_s^{\varphi_1|\varphi_2}(i, j, k) = \begin{cases} \sum_{s' \in S} P(s, s') \cdot v_{s'}^{\varphi_1|\varphi_2}(i-1, j-1, k-1) & \\ \quad \text{if } s \in \text{Sat}(\varphi_2) \cap \text{Sat}(\varphi_1), & \\ \sum_{s' \in S} P(s, s') \cdot v_{s'}^{\varphi_1|\varphi_2}(i-1, j, k-1) & \\ \quad \text{if } s \in \text{Sat}(\varphi_2) \setminus \text{Sat}(\varphi_1), & \\ \sum_{s' \in S} P(s, s') \cdot v_{s'}^{\varphi_1|\varphi_2}(i, j, k-1) & \\ \quad \text{otherwise.} & \end{cases} \quad (5.16)$$

- $I = [t, t']$ ($t > 0$) のとき, until 演算子の場合とは異なり, 時間 t の間は任意の状態を訪れてよい, 以下のように与えることができる.

$$\overrightarrow{\text{Prob}}(\mathbf{Q}_{\sim_c}^{[t,t']}[\varphi_1|\varphi_2]) = \mathbf{P}^t \cdot \overrightarrow{\text{Prob}}^{\mathcal{D}}(\mathbf{Q}_{\sim_c}^{[0,t'-t]}[\varphi_1|\varphi_2]). \quad (5.17)$$

5.1.2 CTMC に対するモデル検査

本項では, 有限状態 CTMC $\mathcal{C}$ に対して, 充足確率ベクトル $\overrightarrow{\text{Prob}}^{\mathcal{C}}(\mathbf{X}^I \varphi_1)$, $\overrightarrow{\text{Prob}}^{\mathcal{C}}(\varphi_1 \mathbf{U}^I \varphi_2)$, $\overrightarrow{\text{Prob}}^{\mathcal{C}}(\mathbf{Q}_{\sim_c}^I[\varphi_1|\varphi_2])$ の計算手法 [80] を示す. ここで, $\mathcal{C}$ のパラメータ λ の一様化 DTMC を $\text{univ}_{\lambda}(\mathcal{C}) = \langle S, \mathbf{P}, P_{\text{init}}, AP, \Lambda \rangle$ とする.

$X^I \varphi_1$ の充足確率ベクトルの計算

この場合、筒集合上の確率測度 Pr を用いて以下のように単純に計算できる.

$$\text{Prob}_s^C(X^I \varphi_1) = \sum_{s' \in \text{Sat}(\varphi_1)} \text{Pr}_s^C(C_s^C(s, I, s')). \quad (5.18)$$

 $\varphi_1 U^I \varphi_2$ の充足確率ベクトルの計算

ここでは五つの場合に分けて考える.

- (1). $\inf I = 0$ かつ $0 \notin I$ のとき, $\varphi_1 U^I \varphi_2$ を満たすパスの初期状態は少なくとも φ_1 を満たす. そしてさらに $\varphi_1 U^{\{0\} \cup I} \varphi_2$ を満たすパスが $\varphi_1 U^I \varphi_2$ を満たす. よって, 以下のように計算できる.

$$\overrightarrow{\text{Prob}}^C(\varphi_1 U^I \varphi_2) = \mathbf{I}_{\varphi_1} \cdot \overrightarrow{\text{Prob}}^C(\varphi_1 U^{\{0\} \cup I} \varphi_2). \quad (5.19)$$

ここで,

$$\mathbf{I}_{\varphi_1}(s, s') = \begin{cases} 1 & \text{if } s = s' \text{ and } s \in \text{Sat}(\varphi_1), \\ 0 & \text{otherwise.} \end{cases} \quad (5.20)$$

- (2). $I = [0, \infty)$ のとき, 求める充足確率ベクトルは, $\mathcal{C}$ 上でいつか φ_2 を満たし, かつ, それまでの間 φ_1 を満たしながら遷移し続ける時間非有界な条件付き到達確率ベクトルである. これは一様化 DTMC $\text{univ}_\lambda(\mathcal{C})$ 上での条件付き到達確率と一致するため, 以下のように計算できる.

$$\overrightarrow{\text{Prob}}^C(\varphi_1 U^{[0, \infty)} \varphi_2) = \overrightarrow{\text{Prob}}^{\text{univ}_\lambda(\mathcal{C})}(\varphi_1 U \varphi_2). \quad (5.21)$$

- (3). $I = [0, t]$ または $I = [0, t)$ ($0 < t < \infty$) のとき, 求める充足確率ベクトルは, $\text{univ}_\lambda(\mathcal{C})$ 上でそれぞれの遷移の生起するタイミングがパラメータ λ の指数分布に従うとき, φ_2 を満たし, かつ, それまでの間 φ_1 を満たしながら遷移し続ける時間有界な条件付き到達確率ベクトルである. よって, 式 (5.2) と (5.4) を用いて, 以下のように計算できる.

$$\overrightarrow{\text{Prob}}(\varphi_1 U^{[0, t]} \varphi_2) = \left(\sum_{i=0}^{\infty} \mathfrak{Pois}_{\lambda \cdot t}(i) \cdot (\mathbf{P}_{\varphi_1 U \varphi_2})^i \right) \cdot \vec{v}(\varphi_2). \quad (5.22)$$

ここで, Poisson 確率 $\mathfrak{Pois}_{\lambda \cdot t}(i)$ は平均して単位時間あたり λ 回起こるイベントが t 単位時間中に i 回生起する確率である (式 (3.20), 付録 B.3 節の定義 B.8).

- (4). $I = [t, t']$ または $I = [t, t')$ ($0 < t < t' < \infty$) のとき, 求める充足確率ベクトルは, $\text{univ}_\lambda(\mathcal{C})$ 上で最初の t 時間は常に φ_1 をみたし, その後の $t' - t$ 時間に φ_1 を満たしながら φ_2 を満たす状態への時間有界な条件付き到達確率である. よって以下のように計算できる.

$$\begin{aligned} \overrightarrow{\text{Prob}}(\varphi_1 U^{[t, t']} \varphi_2) = \\ \left(\sum_{i=0}^{\infty} \mathfrak{Pois}_{\lambda \cdot t}(i) \cdot (\mathbf{P}_{\square \varphi_1})^i \right) \cdot \mathbf{I}_{\varphi_1} \cdot \overrightarrow{\text{Prob}}(\varphi_1 U^{[0, t'-t]} \varphi_2). \end{aligned} \quad (5.23)$$

ここで,

$$\mathbf{P}_{\square \varphi_1}(s, s') = \begin{cases} \mathbf{P}(s, s') & \text{if } s, s' \in \text{Sat}(\varphi_1), \\ 0 & \text{otherwise.} \end{cases} \quad (5.24)$$

- (5). $I = [t, \infty)$ のときは, $I = [t, t']$ と同様に以下のように計算できる.

$$\begin{aligned} \overrightarrow{\text{Prob}}(\varphi_1 U^{[t, \infty)} \varphi_2) = \\ \left(\sum_{i=0}^{\infty} \mathfrak{Pois}_{\lambda \cdot t}(i) \cdot (\mathbf{P}_{\square \varphi_1})^i \right) \cdot \mathbf{I}_{\varphi_1} \cdot \overrightarrow{\text{Prob}}(\varphi_1 U^{[0, \infty)} \varphi_2). \end{aligned} \quad (5.25)$$

$Q_{\sim c}^I[\varphi_1 | \varphi_2]$ の充足確率ベクトルの計算

PFCTL では, 頻度式の部分式 φ_1 と φ_2 は状態式に限定されるため, 区間 I 上で φ_2 が可算無限個の時間点で成り立つことはない. そのため, 条件 φ_2 のもとで φ_1 を満たす条件付き頻度は, ほとんどすべてのパス上で区間の両端の開閉には影響されない. よって, 時間区間 I は $[t, \infty)$, $[0, t]$, $[t, t']$ の三つの場合を示せば充分である. 以下, この三つの場合に分けて示す.

- (1). $I = [t, \infty)$ のとき, 離散時間の場合と同様に, ほとんどすべてのパスはいずれかの BSCC に到達し, その状態出現頻度は定常状態分布と一致する. また, $\text{univ}_\lambda(\mathcal{C})$ は遷移の発生が一様な指数分布に従うために, $\mathcal{C}$ と $\text{univ}_\lambda(\mathcal{C})$ の定常状態分布は一致する. 従って,

$$\overrightarrow{\text{Prob}}^c(Q_{\sim c}^{[t, \infty)}[\varphi_1 | \varphi_2]) = \overrightarrow{\text{Prob}}^{\text{univ}_\lambda(\mathcal{C})}(Q_{\sim c}^{[t, \infty)}[\varphi_1 | \varphi_2]) \quad (5.26)$$

- (2). $I = [0, t]$ のとき, 以下の条件が成り立つパスを考える. ただし, $0 < c < 1$ とする.

- 一様化 DTMC の遷移が時間区間 $[0, t]$ の中で k 回の遷移が生起している.
- * 区間 $[0, t][0, t]$ 中で φ_2 を満たす状態数を i ($1 \leq i \leq k+1$) とする.
- * その i 個の状態に滞在している時間の長さを t' とする.
- * その i 個の状態のうち φ_1 を満たしている状態数を j ($1 \leq j \leq i$) とする.

ここで、これらの仮定を満たすという条件の下で条件付き頻度式 $Q_{<c}^I[\varphi_1|\varphi_2]$ を満たす条件付き確率を三つの場合に分けて考える.

- (i). k 回目の遷移で訪れた状態が $\varphi_1 \wedge \varphi_2$ を満たす場合. 仮定と頻度式を同時に満たす確率は、 l 回分の遷移が区間 $[(1-c) \cdot t', t']$ に生起し、残りの $i-l-1$ 回分の遷移が区間 $[0, (1-c) \cdot t']$ に生起する確率 P_l の $l \geq j$ に関する和 $P (= \sum_{l=j}^{i-1} P_l)$ に等しい. なぜならば、 j 回分以上の遷移が $c \cdot t'$ 時間中に生起しているならば $\varphi_1 \wedge \varphi_2$ を満たす j 個の状態の滞在時間が $c \cdot t'$ より小さいためである. この確率 P_l は、平均して単位時間あたり λ 回起こるイベントが x 単位時間中に y 回生起する確率、すなわち、Poisson 確率 $\mathfrak{Pois}_{\lambda \cdot x}(y)$ (式 (3.20), 付録 B.3 節の定義 B.8) を用いて、

$$\begin{aligned}
 P_l &= \mathfrak{Pois}_{\lambda \cdot (1-c) \cdot t'}(i-l-1) \cdot \mathfrak{Pois}_{\lambda \cdot c \cdot t'}(l) \\
 &= \exp(-\lambda \cdot (1-c) \cdot t') \cdot \frac{(\lambda \cdot (1-c) \cdot t')^{(i-l-1)}}{(i-l-1)!} \\
 &\quad \cdot \exp(-\lambda \cdot c \cdot t') \cdot \frac{(\lambda \cdot c \cdot t')^l}{l!} \\
 &= \exp(-\lambda t') \cdot \frac{(\lambda \cdot t')^{i-1}}{(i-1)!} \cdot \frac{(i-1)! \cdot c^l \cdot (1-c)^{i-l-1}}{l! \cdot (i-l-1)!} \\
 &= \mathfrak{Pois}_{\lambda \cdot t'}(i-1) \cdot \mathfrak{Binom}_{c, i-1}(l). \tag{5.27}
 \end{aligned}$$

ただし、 $\mathfrak{Binom}_{p,x}(y)$ は、成功確率 p の Bernoulli 試行を独立に x 回行った際に、 y 回成功する確率、すなわち、二項確率

$$\mathfrak{Binom}_{p,x}(y) = \frac{x!}{y!(x-y)!} \cdot p^y \cdot (1-p)^{x-y}. \tag{5.28}$$

である (付録 B.3 節の定義 B.5) .

よって確率 P は,

$$P = \sum_{l=j}^{i-1} P_l = \mathfrak{Pois}_{\lambda \cdot t'}(i-1) \cdot \sum_{l=j}^{i-1} \mathbf{Binom}_{c, i-1}(l). \quad (5.29)$$

ここで, Poisson 確率 $\mathfrak{Pois}_{\lambda \cdot t'}(i-1)$ は, 時間 t' 中に $i-1$ 回の遷移が生起する確率であるため, 考慮していた条件付き確率は累積二項確率 $\sum_{l=j}^{i-1} \mathbf{Binom}_{c, i-1}(l)$ と一致する.

- (ii). k 回目の遷移で訪れた状態が $\neg\varphi_1 \wedge \varphi_2$ を満たす場合も, (1) と同様である.
- (iii). k 回目の遷移で訪れた状態が $\neg\varphi_2$ を満たす場合, 時間長 t' はちょうど i 回分の遷移が生起した時間を意味している. そのため, 仮定と頻度式を同時に満たす確率は, $i-l-1$ 回分の遷移が区間 $[0, (1-c) \cdot t']$ に生起し, 残りの時間 $c \cdot t'$ にさらに $l+1$ 回の遷移がちょうど生起する (つまり全体で i 回目の遷移がちょうど時間 t' で生起する) 確率 P'_l の $l \geq j$ に関する和 P' ($= \sum_{l=j}^{i-1} P'_l$) に等しい. この確率 P'_l は, Poisson 確率と, 平均して単位時間あたり λ 回起こるイベントが時間 x にちょうど y 回目の遷移が生起する確率, すなわち, Erlang 確率

$$\mathfrak{Erl}_{\lambda, x}(y) = \frac{\lambda^x \cdot y^{x-1} \cdot \exp(-\lambda \cdot y)}{(x-1)!} \quad (5.30)$$

(付録 B.3 節の定義 B.9) を用いて,

$$\begin{aligned} P'_l &= \mathfrak{Pois}_{\lambda \cdot (1-c) \cdot t'}(i-l-1) \cdot \mathfrak{Erl}_{\lambda, l+1}(c \cdot t') \\ &= \exp(-\lambda \cdot (1-c) \cdot t') \cdot \frac{(\lambda \cdot (1-c) \cdot t')^{(i-l-1)}}{(i-l-1)!} \\ &\quad \cdot \exp(-\lambda \cdot c \cdot t') \cdot \frac{\lambda^{l+1} \cdot (c \cdot t')^l}{l!} \\ &= \exp(-\lambda t') \cdot \frac{\lambda^i \cdot (t')^{i-1}}{(i-1)!} \cdot \frac{(i-1)! \cdot c^l \cdot (1-c)^{i-l-1}}{l! \cdot (i-l-1)!} \\ &= \mathfrak{Erl}_{\lambda, i}(t') \cdot \mathbf{Binom}_{c, i-1}(l). \end{aligned} \quad (5.31)$$

よって確率 P' は,

$$P' = \sum_{l=j}^{i-1} P'_l = \mathfrak{Erl}_{\lambda, i}(t') \cdot \sum_{l=j}^{i-1} \mathbf{Binom}_{c, i-1}(l). \quad (5.32)$$

ここで, Erlang 確率 $\text{Erl}_{\lambda,i}(t)$ は, i 回目の遷移が時間 t にちょうど生起する確率であるため, 考慮していた条件付き確率はやはり累積二項確率 $\sum_{l=j}^{i-1} \text{Binom}_{c,i-1}(l)$ と一致する.

以上により, (i)-(iii) の条件付き確率は累積二項確率 $\sum_{l=j}^{i-1} \text{Binom}_{c,i-1}(l)$ である. この事実と, それぞれの状態を始点として k ステップで φ_2 と $\varphi_1 \wedge \varphi_2$ を満たす状態をそれぞれ i 回と j 回訪れるパスの生起確率を与えるベクトル $\vec{v}^{\varphi_1|\varphi_2}(i, j, k)$ (式 (5.13)-(5.16)) を用いて目的の充足確率ベクトルの計算を以下のように計算できる.

$$\begin{aligned} \overrightarrow{\text{Prob}}^c(Q_{\sim c}^{[0,t]}[\varphi_1|\varphi_2]) = \\ \sum_{k=0}^{\infty} \sum_{i=0}^{k+1} \sum_{j=0}^i \left(\text{Pois}_{\lambda \cdot t}(k) \cdot B_{\sim c}(i, j) \cdot \vec{v}^{\varphi_1|\varphi_2}(i, j, k) \right). \end{aligned} \quad (5.33)$$

ただし,

$$B_{\sim c}(i, j) = \begin{cases} 1 & \text{if } i = 0, (i = j \text{ and } 1 \sim c) \text{ or} \\ & (j = 0 \text{ and } 0 \sim c), \\ \sum_{l=j}^{i-1} \text{Binom}_{c,i-1}(l) & \text{if } 0 < j < i, c \in (0, 1) \text{ and } \sim \in \{<, \leq\}, \\ \sum_{l=0}^{j-1} \text{Binom}_{c,i-1}(l) & \text{if } 0 < j < i, c \in (0, 1) \text{ and } \sim \in \{>, \geq\}, \\ 0 & \text{otherwise.} \end{cases} \quad (5.34)$$

- (4). $I = [t, t']$ ($t > 0$) のとき, 途中で訪れた状態は考慮しなくても良いため, 過渡状態確率行列 Π_t^c を用いて以下のように計算できる.

$$\overrightarrow{\text{Prob}}^c(Q_{\sim c}^{[t,t']}[\varphi_1|\varphi_2]) = \Pi_t^c \cdot \overrightarrow{\text{Prob}}^c(Q_{\sim c}^{[0,t'-t]}[\varphi_1|\varphi_2]) \quad (5.35)$$

5.2 LTL 型フラグメントモデル検査

本節では, DTMC/CTMC モデルに対する, PFTL の LTL 型フラグメントモデル検査について述べる.

LTL 型の構文であるとは, パス限量子を部分式として用いることができない構文である. PFTL の LTL 型フラグメントを PFLTL といい, 以下のように定義される.

定義 5.2 (PFTL の構文) AP を原子命題の集合とする. このとき PFTL の式 φ は, 以下のように帰納的に定義される.

$$\begin{aligned}\varphi &::= P_{\sim c}[\psi] \\ \psi &::= p \mid \neg\psi \mid \psi \vee \psi \mid X^I\psi \mid \psi U^I\psi \mid Q_{\sim c}^I[\psi \mid \psi].\end{aligned}$$

ただし, $p \in AP$, $c \in [0, 1]$, $\sim \in \{<, >, \leq, \geq\}$, I は自然数上の区間 (離散時間) あるいは非負実数の区間 (連続時間) とする. ■

DTMC/CTMC に対する PFTL モデル検査手法をそれぞれ 5.2.1 項と 5.2.2 に示す.

5.2.1 DTMC に対するモデル検査

本項では, 有限状態 DTMC モデルに対する PFTL の LTL 型フラグメント PFTL のモデル検査/充足確率問題について述べる. この離散時間 PFTL モデル検査/充足確率問題が決定可能であるということは, 充足集合 $\text{Sat}(P_{\sim c}[\psi])$ /充足確率ベクトル $\text{Prob}^{\mathcal{D}}(\psi)$ を決定/計算できるため, 離散時間 PFTL モデル検査/充足確率問題が決定/計算可能であることを意味する. LTL の確率的モデル検査/充足確率問題については, 指数時間アルゴリズム [40, 41, 24] が開発されており, 本論文では離散時間 PFTL モデル検査を確率的 LTL モデル検査に帰着して決定できることを示す.

時間有界な頻度式 Q^I については, それと等価な式を時間区間の長さ $|I|$ に対して指数オーダー個の next 演算子 X を用いて記述できる. 例えば, $Q_{\geq 1/2}^{\leq 3}\varphi$ は, 以下のような等価な LTL 式で表現できる.

$$\begin{aligned}(\varphi \wedge X\varphi) \vee (X(\varphi \wedge X\varphi)) \vee (XX(\varphi \wedge X\varphi)) \vee \\ (\varphi \wedge XX\varphi) \vee X(\varphi \wedge XX\varphi) \vee (\varphi \wedge XXX\varphi)\end{aligned}$$

そのため, 容易に確率的 LTL モデル検査に帰着できる.

時間非有界な頻度式 Q については, 有限状態 DTMC がもつ以下の三点の特徴を利用して確率的 LTL モデル検査/充足確率問題に帰着する.

- 有限状態マルコフ連鎖のほとんどすべてのパスは, 何れかの BSCC B に到達する.

- そして、ほとんどすべてのパスは到達した BSCC B の定常状態分布に従って状態出現頻度 $\overrightarrow{\text{Freq}}^{\mathcal{D}}(B)$ (式 (3.9) の解) が決まる, すなわち, 状態出現頻度が収束する.
- 離散時間上では, 状態と時間の粒度が等しい.

これより, 以下の補題が成り立つ.

補題 5.3 $\mathcal{D} = \langle S, \mathbf{P}, P_{init}, AP, \Lambda \rangle$ を DTMC とする. $\mathcal{D}$ の BSCC $\langle B, \hookrightarrow \rangle \in \text{BSCC}^{\mathcal{D}}$ に到達するほとんどすべてのパス $\rho \in \text{Paths}^{\mathcal{D}}$ に対して, LTL 式 ψ を満たす時点の出現頻度 $\text{Freq}_B^{\mathcal{D}}(\psi)$ は以下のように収束する.

$$\begin{aligned} \text{Freq}_B^{\mathcal{D}}(\psi) &= \lim_{k \rightarrow \infty} \frac{|\{i \in [0, k] \mid \mathcal{D}, \rho[i \dots] \models \psi\}|}{k} \\ &= \sum_{s \in B} (\text{Freq}_s^{\mathcal{D}}(B) \cdot \text{Prob}_s^{\mathcal{D}}(\psi)). \end{aligned} \quad (5.36)$$

■

証明. B に到達したほとんどすべてのパス上では, 状態の出現頻度は $\overrightarrow{\text{Freq}}^{\mathcal{D}}(B)$ に収束する. また, 状態 $s \in B$ は無限にしばしば出現するため, ほとんどすべてのパス上で状態 s に対題していて, かつ, φ を満たす頻度は $\text{Freq}_s^{\mathcal{D}}(B) \cdot \text{Prob}_s^{\mathcal{D}}(\varphi)$ である. 以上により式 (5.36) が成り立つ. □

ここで, 時間非有界な頻度式 $\psi = \mathbf{Q}_{\sim c}[\psi_1 | \psi_2]$ に対して, ψ_2 のもとでの ψ_1 の条件付き頻度が定義できない, または, 閾値条件 $\sim c$ を満たす $\mathcal{D}$ 上の BSCC $\langle B, \hookrightarrow \rangle$ の和集合を $\text{SatBSCC}^{\mathcal{D}}(\psi)$ と書くこととする.

$$\text{SatBSCC}^{\mathcal{D}}(\psi) = \left\{ B \in \text{BSCC}^{\mathcal{D}} \mid \text{Freq}_B^{\mathcal{D}}(\psi_2) > 0 \Rightarrow \frac{\text{Freq}_B^{\mathcal{D}}(\psi_1 \wedge \psi_2)}{\text{Freq}_B^{\mathcal{D}}(\psi_2)} \sim c \right\}. \quad (5.37)$$

このとき, 補題 5.3 から, DTMC のほとんどすべてのパス上では, 「時間非有界な頻度式 ψ を充足すること」と「 $\text{SatBSCC}^{\mathcal{D}}(\psi)$ へ到達すること」は一致する.

補題 5.4 $\mathcal{D} = \langle S, \mathbf{P}, P_{init}, AP, \Lambda \rangle$ を DTMC, ψ を時間非有界な頻度式とする. このとき, 以下が成り立つ.

$$\text{Prob}^{\mathcal{D}'}(\{\rho \in \text{Paths}^{\mathcal{D}'} \mid \mathcal{D}', \rho \models \Diamond p_\psi \leftrightarrow \psi\}). \quad (5.38)$$

ただし, $\mathcal{D}' = \langle S, \mathbf{P}, P_{init}, AP \cup \{p_\psi\}, \Lambda' \rangle$, p_ψ は新しい補助命題, Λ' は以下のよう定義されるラベリング関数とする.

$$\Lambda'(s) = \begin{cases} \Lambda(s) \cup \{p_\psi\} & \text{if } s \in \text{SatBSCC}^{\mathcal{D}}(\psi), \\ \Lambda(s) & \text{otherwise.} \end{cases} \quad (5.39)$$

さらに, $\mathcal{D}'$ と PFTL パス式 φ , $\varphi' = \varphi[\psi/\diamond p_\psi]$ について, 以下を満たす.

$$\text{Prob}_s^{\mathcal{D}}(\varphi) = \text{Prob}_s^{\mathcal{D}'}(\varphi'). \quad (5.40)$$

■

証明. 式 (5.38) は補題 5.3 により明らか. 式 (5.40) は式 (5.38) と $\mathcal{D}'$ の定義から AP 上の任意の PFTL パス式 ψ' について, $\mathcal{D}, \rho \models \psi'$ ならば, かつそのときに限り, $\mathcal{D}', \rho \models \psi'$ であることから明らか. □

補題 5.4 から, PFTL モデル検査/充足確率問題では, 時間非有界な頻度式を定性的到達性制約で置き換えることができる. ここで, 頻度式 ψ に注目して, 補題 5.4 の対 $\langle \mathcal{D}, \varphi \rangle$ から得られる対 $\langle \mathcal{D}', \varphi' \rangle$ を $\langle \mathcal{D}, \varphi \rangle \downarrow_\psi$ と書くこととする. この変形を繰り返すことで, DTMC に対する PFTL モデル検査/充足確率問題を古典的 LTL のそれらの問題に帰着できる.

定理 5.5 DTMC に対する PFTL モデル検査/充足確率問題は二重指数時間で決定可能/計算可能である.

証明. $\mathcal{D}$ を DTMC, φ を n 個の時間非有界頻度式 $\psi_1, \dots, \psi_n$ をもつ PFTL 式とする. 補題 5.4 から, $\mathcal{D}$ が φ を充足する確率は, n 回の変形により得られる対 $\langle \mathcal{D}', \varphi' \rangle = (\dots (\langle \mathcal{D}, \varphi \rangle \downarrow_{\psi_1}) \dots) \downarrow_{\psi_n}$ について, $\mathcal{D}'$ が φ' を充足する確率と等しい.

それぞれの変形 $\downarrow_{\psi_i}$ では, 式 (5.36) を計算する必要がある. そのためには, BSCC に属するそれぞれの状態について LTL 充足確率問題 $\text{Prob}_s^{\mathcal{D}}(\varphi_i)$ を解けばよい. [40] と [41] から, LTL 充足確率問題は指数時間で解くことができるため, それぞれの変形は, $|\mathcal{D}|$ について多項式時間かつ $|\varphi|$ について指数時間で計算できる.

変形の結果として得られた $\mathcal{D}'$ と φ' のサイズは増えることはないため, $\mathcal{D}'$ に対する φ' の確率的モデル検査/充足確率問題は, $|\mathcal{D}|$ について線形/多項式時間かつ $|\varphi|$ について指数時間で計算できる.

以上から, 全体では $|\mathcal{D}|$ について多項式時間, $|\varphi|$ について指数時間, 時間有界な頻度式の最大時間区間長について二重指数時間, n について線形時間で確率的モデル検査/充足確率問題を決定/計算できる. □

5.2.2 CTMC に対する統計的モデル検査

離散時間においては、状態遷移と時間変化の粒度が等しいため、 X 演算子をもつ時間論理に時間区間制約を追加しても本質的には表現力は変化しない。そのため、古典的な線形時間論理である LTL の確率的モデル検査は、到達確率解析などにより判定が行われる [40, 41, 24]。一方、連続時間においては、状態遷移と時間変化の粒度が等しくないため、時間区間制約を追加することで時間論理の表現力は本質的に大きくなる。計量時間論理 [56] (Metric Temporal Logic, MTL) のように時間区間制約がある線形実時間論理のモデル検査では部分式の充足時間を厳密に考慮しなければならないため、確率的 MTL モデル検査は困難であり、時間有界検査手法しか提案されていない [35]。

PFLTL は、一般的な時間演算子に加え条件付き頻度演算子 Q をもつため、厳密なモデル検査は非常に困難である。そこで本研究では、統計的手法を用いた時間有界 PFLTL の近似的モデル検査 (統計的 PFLTL モデル検査) を与える [80]。統計的モデル検査は、元々は CSL モデル検査のために開発された、統計的検定/推定による近似的検査手法である [92]。この手法では、検査式 $P_{\sim c}[\psi]$ に対する過誤率 ϵ の判定や質問式 $P_{=?}[\psi]$ に対する厳密な計算はできないが、厳密なモデル検査手法では扱えないようなモデルと検査式に対しても統計的な誤差評価の下で近似的に検査できる。そのため、統計的モデル検査を用いることで、他の近似的解析手法と同様に、工学的に有用な知見を得ることも可能である。

本項では、まず統計的モデル検査の概略を示し、その後、検査の各工程について説明する。

統計的モデル検査の概略

統計的手法を用いたモデル検査は以下の三つの工程からなる。

1. 標本集団 $\{\rho_1, \dots, \rho_n\}$ の無作為抽出
 - モデル C の確率的シミュレーションによる標本 (パス) $\rho_1, \dots, \rho_n$ の生成
2. 各標本 ρ_i の検査パス式 ψ に対する充足判定
 - 検査パス式を記述する言語 (論理) に応じた充足判定手続きによる「 $C, \rho_i \models \psi$ か否か」の計算

3. 各標本に対する充足判定結果を用いた標本集団の統計的評価

- 入力検査式 $P_{\sim p}[\psi]$ のとき、仮説検定による真偽判定
- 入力質問式 $P_{=?}[\psi]$ のとき、区間推定による ψ の充足確率の推定

無限時間長の標本パスを確率的シミュレーションにより実際に生成することはできないため、必然的に生成する標本パスは有限時間長である。そして、その標本パスが検査パス式 ψ を満たすか判定できるならば、統計的モデル検査は可能である。しかしながら、一般に、パスの有限時間長プレフィックスのみで ψ の真偽を判定することはできない。そのため統計的モデル検査では、通常、検査パス式 ψ は時間有界である ($\#(\psi) < \infty$) と仮定する。パス式 ψ が時間有界である場合、 ψ の真偽は標本パスの時刻 $\#(\psi)$ までのプレフィックスのみに依存するため、検査パス式 ψ に対しては時間長 $\#(\psi)$ の標本パスを確率的シミュレーションで生成すればよい。この有限時間長の確率的シミュレーションは、可算無限状態の CTMC であっても、非爆発性モデルであれば容易に計算できる。そして、生成された標本集団を仮説検定や区間推定を用いて統計的に評価し、出力を決定できる。

注意 5.6 (複数の確率パス限量子をもつ式に対する統計的モデル検査) 複数の確率パス限量子を用いて式を記述する CSL の統計的モデル検査では、検査パス式 ψ が部分式 $P_{\sim c}[\psi]$ をもつ場合、まず、それぞれの状態 s に関して $\mathcal{C}, s \models P_{\sim c}[\psi]$ を検査する必要がある。そのため、CSL モデル検査手続きのすべてを統計的手法で行うためには、 ψ 式中のパス限量子の数 N とモデルの状態数 $|S|$ に対して、 $\mathcal{O}(N \times |S|)$ 回の多重検定が必要となる。 ■

注意 5.7 (時間非有界式に対する近似的モデル検査) 統計的手法では、直接的に検査可能な式は時間有界でなければならない。ただし、時間有界式に対する統計的モデル検査に加え、到達確率解析や定常分布解析などを行うことにより、時間非有界式に対する近似的検査が可能となる場合もある [90]。 ■

確率的シミュレーションによる標本パス生成

確率的シミュレーション手法は、解析対象となるモデルの種類のみに依存する。そのため、この工程で用いられる手法は、統計的 PFTL モデル検査に特有のものではない。非爆発的 CTMC の確率的シミュレーションは Gillespie 法 [48] などの一般的な疑似乱数を用いる手法で行うことができるため、本稿では省略する。この確率的シミュレーションは、検査パス式 ψ の時間制約上界

$\#(\psi)$, CTMC の平均退出速度と平均遷移可能状態数に対して線形時間で計算できる.

注意 5.8 (統計的モデル検査を用いて検査可能なモデル) ある種の限定されたクラスの可算無限状態 CTMC に対する CSL モデル検査手法 [71, 50] はこれまでにいくつか提案されているが, 一般の可算無限状態 CTMC に対する厳密な CSL モデル検査手法は提案されていない. それに対して, 時間有界 PFLTL の統計的モデル検査は, 確率的シミュレーションにより離散化した標本パスが生成できるモデルであれば検査可能である. そのため, 統計的モデル検査では, 可算無限状態 CTMC だけでなく連続状態や非 Markov 過程, 確率的ハイブリッドシステム [95], 確率微分方程式などの通常のモデル検査では検査困難なモデルも近似的に検査可能である.

なお, 非爆発的でない CTMC は, 平均退出速度が ∞ に発散しシミュレーションが停止しない可能性があるため, 統計的モデル検査では扱うことができない. また, 非決定性をもつモデルも, パス集合上の確率測度を単純には定義できないため, 統計的モデル検査では検査不能である. ■

標本パスの時間有界 PFLTL パス式充足判定

次に, 与えられた有限時間長標本パスが時間有界 PFLTL パス式 ψ ($\#(\psi) < \infty$) を満たすか判定する手法を示す. この工程は, 検査パス式を記述する言語 (論理) に依存するため, 統計的 PFLTL モデル検査に特有のものである. ここでは, 前節で得られた有限時間長プレフィックス $\langle s_0, t_0 \rangle \dots \langle s_n, t_n \rangle$ をもつパス $\rho \in \text{Paths}^C$ が時間有界 PFLTL パス式 ψ を満たすか否か, 即ち, $\mathcal{C}, \rho \models \psi$ の真偽を判定する場合を考える. ただし, $\#(\psi) < \sum_{i=0}^n t_i = T_n$ とする. また, 本節の時間区間はすべて有界である.

定義 4.5 から, 区間 $[0, T_n)$ 中の ψ を満たす時刻の集合 $\text{Sat}_{\mathcal{C}, \rho}^{<T_n}(\psi)$ に関しては, 以下の関係が成り立つ.

$$\mathcal{C}, \rho \models \psi \Leftrightarrow 0 \in \text{Sat}_{\mathcal{C}, \rho}^{<T_n}(\psi). \quad (5.41)$$

そして, $\text{Sat}_{\mathcal{C}, \rho}^{<T_n}(\psi)$ は, ψ の部分式 ψ_1 と ψ_2 を満たす時刻の集合 $\text{Sat}_{\mathcal{C}, \rho}^{<T_n}(\psi_1)$ と $\text{Sat}_{\mathcal{C}, \rho}^{<T_n}(\psi_2)$ から再帰的に計算できる. ここで, 時間有界 PFLTL パス式 ψ に対して, $\text{Sat}_{\mathcal{C}, \rho}^{<T_n}(\psi)$ は有限個の区間の和集合として表現可能である. そこで実際の計算では, ψ を満たす時刻の集合 $\text{Sat}_{\mathcal{C}, \rho}^{<T_n}(\psi)$ の代わりに, ψ を満たす区間の

有限集合を計算することとし、その区間集合を $\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi)$ で表す.

$$\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi) = \bigcup_{I_i \in \text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi)} I_i. \quad (5.42)$$

また、時間区間集合 $\{I_1, \dots, I_m\}$ と $\{J_1, \dots, J_l\}$ の同値関係 $\doteq$ を以下のように定義する.

$$\{I_1, \dots, I_m\} \doteq \{J_1, \dots, J_l\} \Leftrightarrow \bigcup_{i=1}^m I_i = \bigcup_{j=1}^l J_j. \quad (5.43)$$

このとき、 $\{I_1, \dots, I_m\}$ と同値な最小の要素数をもつ時間区間集合を、 $\{I_1, \dots, I_m\}$ の同値類の代表元とする. この代表元の要素は、明らかに互いに素、かつ、互いに連結でない. 以後は、 $\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi) = \{I_1, \dots, I_m\}$ と書いたときには、時間区間集合 $\{I_1, \dots, I_m\}$ は $\inf I_i < \inf I_{i+1}$ であるよう各要素にインデックスが付けられた代表元とする.

以下では、 ψ の部分式 ψ_1 と ψ_2 の充足区間集合は既に計算されているものとし、式 ψ の充足区間集合 $\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi)$ の求め方を示す.

- (1). $\psi = a \in AP$ のときには、 $\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi)$ は CTMC $\mathcal{C}$ のラベリング関数 Λ から容易に求められる.

$$\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi) \doteq \{[T_{i-1}, T_i] \mid 0 \leq i \leq n \text{ and } a \in \Lambda(s_i)\}. \quad (5.44)$$

ただし、 $T_i = \sum_{k=0}^i t_k$, $T_{-1} = 0$ とする.

- (2). $\psi = \neg\psi_1$ のとき、時間区間 $[0, T_n]$ から $\text{SatInt}_{\mathcal{C},\rho}(\psi_1) = \{I_1, \dots, I_m\}$ に含まれる区間を取り除いて断片化された区間の集合が $\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\neg\psi_1)$ である. 従って、

$$\begin{aligned} \text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi) \doteq & \{[0, \inf I_1] \setminus I_1, [\sup I_m, T_n] \setminus I_m\} \\ & \cup \{[\sup I_i, \inf I_{i+1}] \setminus (I_i \cup I_{i+1}) \mid 1 \leq i < m\} \end{aligned} \quad (5.45)$$

- (3). $\psi = \psi_1 \vee \psi_2$ のとき、 $\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi)$ は $\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi_1)$ と $\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi_2)$ の和集合と同値である. 従って、

$$\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi) \doteq \text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi_1) \cup \text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi_2). \quad (5.46)$$

- (4). $\psi = X^I \psi_1$ のとき, $\text{SatInt}_{\mathcal{C}, \rho}^{<T_n}(\psi_1) = \{I_1, \dots, I_m\}$ のある要素 I_i に含まれる, $k+1$ 回目の遷移の発生した時刻 $T_k \in I_i$ について, 「区間 $[T_{k-1}, T_k)$ 中で $t' \in I$ 時間後に $k+1$ 回目の遷移が起こる」ような時刻 t からなる区間

$$X_{k,i} = \{t \in [T_{k-1}, T_k) \mid T_k \in (I+t) \cap I_i\}$$

の集合が $\text{SatInt}_{\mathcal{C}, \rho}^{<T_n}(X^I \psi_1)$ である. 従って,

$$\text{SatInt}_{\mathcal{C}, \rho}^{<T_n}(\psi) \doteq \{X_{k,i} \mid 0 \leq k < n, 1 \leq i \leq m \text{ and } X_{k,i} \neq \emptyset\} \quad (5.47)$$

- (5). $\psi = \psi_1 \cup^I \psi_2$ のとき,

- (5-a). $0 \notin I$ のとき, $\text{SatInt}_{\mathcal{C}, \rho}^{<T_n}(\psi_1) = \{I_1, \dots, I_m\}$ と $\text{SatInt}_{\mathcal{C}, \rho}^{<T_n}(\psi_2) = \{J_1, \dots, J_l\}$ に対して, 区間 I_i で ψ_1 を満たしながら区間 J_j に到達し ψ_2 を満たす場合を考える. このとき, その ψ_2 を満たす時刻は, 区間

$$Y_{i,j} = (I_i \cup \{\sup I_i\}) \cap J_j$$

に含まれる. そのため, 区間 I_i の中で, $t' \in I$ 時間後に区間 $Y_{i,j}$ へ到達するような時刻 t からなる区間

$$Z_{i,j} = \{t \in I_i \mid (I+t) \cap Y_{i,j} \neq \emptyset\}$$

で式 ψ は成り立つ. 従って,

$$\text{SatInt}_{\mathcal{C}, \rho}^{<T_n}(\psi) \doteq \{Z_{i,j} \mid 1 \leq i \leq m, 1 \leq j \leq l \text{ and } Z_{i,j} \neq \emptyset\} \quad (5.48)$$

- (5-b). $0 \in I$ のとき, 明らかに $\text{SatInt}_{\mathcal{C}, \rho}^{<T_n}(\psi_2)$ は $\text{SatInt}_{\mathcal{C}, \rho}^{<T_n}(\psi)$ に含まれる. 従って,

$$\text{SatInt}_{\mathcal{C}, \rho}^{<T_n}(\psi) \doteq \text{SatInt}_{\mathcal{C}, \rho}^{<T_n}(\psi_2) \cup \text{SatInt}_{\mathcal{C}, \rho}^{<T_n}(\psi_1 \cup^{I \setminus \{0\}} \psi_2) \quad (5.49)$$

- (6). $\psi = Q_{\sim c}^I \langle \psi_1 \mid \psi_2 \rangle$ のとき, 頻度閾値条件 $\sim c$ が ≥ 0 または ≤ 1 ならば明らかに,

$$\text{SatInt}_{\mathcal{C}, \rho}^{<T_n}(\psi) = \{[0, T_n)\}. \quad (5.50)$$

そうでないならば, 以下の (6-a)-(6-c) により $\text{SatInt}_{\mathcal{C}, \rho}^{<T_n}(\psi)$ は与えられる.

(6-a). $I = [0, 0]$ のとき, ψ は単なる含意関係である. 従って,

$$\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi) \doteq \begin{cases} \text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi_2 \rightarrow \psi_1) & \text{if } 1 \sim c, \\ \text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi_2 \rightarrow \neg\psi_1) & \text{otherwise.} \end{cases} \quad (5.51)$$

(6-b). $\inf I = 0$ かつ $\sup I > 0$ のとき, 定義 4.5 より, ψ の充足時間集合 $\text{Sat}_{\mathcal{C},\rho}^{<T_n}(\psi)$ は, 以下の条件を満たす.

$$\text{Sat}_{\mathcal{C},\rho}^{<T_n}(\psi) = \{t \in [0, T_n) \mid f_{\mathcal{C},\rho,\psi}^I(t) \sim 0 \text{ or } |\text{Sat}_{\mathcal{C},\rho}^{J+t}(\psi_2)| = 0\} \quad (5.52)$$

ただし, $f_{\mathcal{C},\rho,\psi}^J(t) : [0, T_n) \rightarrow \mathbb{R}$ は以下のように定義される関数である.

$$f_{\mathcal{C},\rho,\psi}^J(t) = \begin{cases} \mathfrak{L}(\text{Sat}_{\mathcal{C},\rho}^{J+t}(\psi_1 \wedge \psi_2)) - c \cdot \mathfrak{L}(\text{Sat}_{\mathcal{C},\rho}^{J+t}(\psi_2)) & \text{if } \mathfrak{L}(\text{Sat}_{\mathcal{C},\rho}^{J+t}(\psi_2)) > 0, \\ |\text{Sat}_{\mathcal{C},\rho}^{J+t}(\psi_1 \wedge \psi_2)| - c \cdot |\text{Sat}_{\mathcal{C},\rho}^{J+t}(\psi_2)| & \text{otherwise.} \end{cases} \quad (5.53)$$

よって, 関数 $f_{\mathcal{C},\rho,\psi}^J(t)$ の値の変化を調べることで, ψ の充足時間区間集合 $\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi)$ を得ることができる. なお, $\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi_1 \wedge \psi_2) = \{J_1, \dots, J_l\}$, $\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi_2) = \{I_1, \dots, I_m\}$ に対して,

$$\text{Sat}_{\mathcal{C},\rho}^J(\psi_1 \wedge \psi_2) = \bigcup_{1 \leq j \leq l} J_j \cap J, \quad (5.54)$$

$$\text{Sat}_{\mathcal{C},\rho}^J(\psi_2) = \bigcup_{1 \leq i \leq m} I_i \cap J \quad (5.55)$$

である. また, この関数 $f_{\mathcal{C},\rho,\psi}^J(t)$ は, ψ_2 や $\psi_1 \wedge \psi_2$ の真偽が変化する時刻とその $\sup I$ 単位時間前, 即ち, 以下のように定義される境界点集合 $\text{Bnd}_{\mathcal{C},\rho}^{<T_n}(\psi)$ を除いて区分的に線形である.

$$\begin{aligned} \text{Bnd}_{\mathcal{C},\rho}^{<T_n}(\psi) = & \{0, T_n - \sup I\} \\ & \cup \bigcup_{1 \leq i \leq m} \{\inf I_i - \sup I, \inf I_i, \sup I_i - \sup I, \sup I_i\} \cap [0, T_n) \\ & \cup \bigcup_{1 \leq j \leq l} \{\inf J_j - \sup I, \inf J_j, \sup J_j - \sup I, \sup J_j\} \cap [0, T_n). \end{aligned} \quad (5.56)$$

このとき、昇順に整列した境界点集合 $\{\tau_1, \dots, \tau_\kappa\} = \text{Bnd}_{\mathcal{C}, \rho}^{<T_n}(\psi)$ に対して、それぞれの点 $\{\tau_k\}$ と开区間 (τ_k, τ_{k+1}) (の部分集合) が $\text{Sat}_{\mathcal{C}, \rho}^{<T_n}(\psi)$ に属するかを判定することで $\text{SatInt}_{\mathcal{C}, \rho}^{<T_n}(\psi)$ を計算できる.

$$\begin{aligned} \text{SatInt}_{\mathcal{C}, \rho}^{<T_n}(\psi) \doteq & \bigcup_{1 \leq k \leq \kappa} \left(\left\{ \{\tau_k\} \mid f_{\mathcal{C}, \rho, \psi}^I(\tau_k) \sim 0 \text{ or } |\text{Sat}_{\mathcal{C}, \rho}^{I+\tau_k}(\psi_2)| = 0 \right\} \right. \\ & \left. \cup \text{SubSatInt}_{\mathcal{C}, \rho, \psi}^{<T_n}(\tau_k, \tau_{k+1}) \right). \end{aligned} \quad (5.57)$$

ここで、 $\text{SubSatInt}_{\mathcal{C}, \rho, \psi}^{<T_n}(\tau_k, \tau_{k+1})$ は、开区間 (τ_k, τ_{k+1}) のうち ψ を満たす時刻からなる部分区間であり、以下のように計算できる.

(6-b-i). $\mathfrak{L}(\text{Sat}_{\mathcal{C}, \rho}^{I+\tau_k}(\psi_2)) = 0$ かつ $\mathfrak{L}(\text{Sat}_{\mathcal{C}, \rho}^{I+\tau_{k+1}}(\psi_2)) > 0$ のとき、関数 $f_{\mathcal{C}, \rho, \psi}^I$ は時刻 τ_k において不連続となり得るが、以下の関係が成り立つ.

$$f_{\mathcal{C}, \rho, \psi}^I(\tau_{k+1}) \sim 0 \Leftrightarrow \forall t \in (\tau_k, \tau_{k+1}). f_{\mathcal{C}, \rho, \psi}^I(t) \sim 0. \quad (5.58)$$

従って,

$$\text{SubSatInt}_{\mathcal{C}, \rho, \psi}^{<T_n}(\tau_k, \tau_{k+1}) = \begin{cases} (\tau_k, \tau_{k+1}) & \text{if } f_{\mathcal{C}, \rho, \psi}^I(\tau_{k+1}) \sim 0 \\ \emptyset & \text{otherwise.} \end{cases} \quad (5.59)$$

(6-b-ii). $\mathfrak{L}(\text{Sat}_{\mathcal{C}, \rho}^{I+\tau_k}(\psi_2)) > 0$ かつ $\mathfrak{L}(\text{Sat}_{\mathcal{C}, \rho}^{I+\tau_{k+1}}(\psi_2)) = 0$ のとき、(6-b-i) と同様に,

$$\text{SubSatInt}_{\mathcal{C}, \rho, \psi}^{<T_n}(\tau_k, \tau_{k+1}) = \begin{cases} (\tau_k, \tau_{k+1}) & \text{if } f_{\mathcal{C}, \rho, \psi}^I(\tau_k) \sim 0 \\ \emptyset & \text{otherwise.} \end{cases} \quad (5.60)$$

(6-b-iii). $\mathfrak{L}(\text{Sat}_{\mathcal{C}, \rho}^{I+\tau_k}(\psi_2)) > 0$ かつ $\mathfrak{L}(\text{Sat}_{\mathcal{C}, \rho}^{I+\tau_{k+1}}(\psi_2)) > 0$ のとき、関数 $f_{\mathcal{C}, \rho, \psi}^I$ は閉区間 $[\tau_k, \tau_{k+1}]$ において連続である. 従って,

$$\begin{aligned} & \text{SubSatInt}_{\mathcal{C}, \rho, \psi}^{<T_n}(\tau_k, \tau_{k+1}) \\ &= \{t \in (\tau_k, \tau_{k+1}) \mid f_{\mathcal{C}, \rho, \psi}^I(t) \sim 0\} \\ &= \left\{ \tau_k + t \in (\tau_k, \tau_{k+1}) \mid t \sim \frac{-f_{\mathcal{C}, \rho, \psi}^I(\tau_k) \cdot (\tau_{k+1} - \tau_k)}{f_{\mathcal{C}, \rho, \psi}^I(\tau_{k+1}) - f_{\mathcal{C}, \rho, \psi}^I(\tau_k)} \right\}. \end{aligned} \quad (5.61)$$

(6-b-iv). $\mathfrak{L}(\text{Sat}_{\mathcal{C},\rho}^{I+\tau_k}(\psi_2)) = \mathfrak{L}(\text{Sat}_{\mathcal{C},\rho}^{I+\tau_{k+1}}(\psi_2)) = 0$ のとき, 関数 $f_{\mathcal{C},\rho,\psi}^I$ は時刻 τ_k と τ_{k+1} において不連続となり得るが, 开区間 (τ_k, τ_{k+1}) 中では変化しない. また, 時刻 $t \in (\tau_k, \tau_{k+1})$ に関して, 以下の関係が成り立つ.

$$\text{Sat}_{\mathcal{C},\rho}^{I+t}(\psi_2) = \text{Sat}_{\mathcal{C},\rho}^{(I \cup \{\sup I\})+\tau_k}(\psi_2), \quad (5.62)$$

$$\text{Sat}_{\mathcal{C},\rho}^{I+t}(\psi_1 \wedge \psi_2) = \text{Sat}_{\mathcal{C},\rho}^{(I \cup \{\sup I\})+\tau_k}(\psi_1 \wedge \psi_2). \quad (5.63)$$

従って,

$$\text{SubSatInt}_{\rho,L,\psi}^{<T_n}(\tau_k, \tau_{k+1}) = \begin{cases} (\tau_k, \tau_{k+1}) & \text{if } f_{\mathcal{C},\rho,\psi}^{I \cup \{\sup I\}}(\tau_k) \sim 0 \text{ or} \\ & \text{Sat}_{\mathcal{C},\rho}^{(I \cup \{\sup I\})+\tau_k}(\psi_2) = \emptyset, \\ \emptyset & \text{otherwise.} \end{cases} \quad (5.64)$$

(6-c). $\inf I > 0$ のとき, それぞれの時点 $t \in [0, T_n)$ について, $t \in \text{Sat}_{\mathcal{C},\rho}^{<T_n}(\psi)$ ならば, かつそのときに限り, $t + \inf I \in \text{Sat}_{\mathcal{C},\rho}^{<T_n}(\mathbf{Q}_{\sim q}^{I-\inf I} \langle \psi_1 \mid \psi_2 \rangle)$ である. 従って, $\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\mathbf{Q}_{\sim q}^{I-\inf I} \langle \psi_1 \mid \psi_2 \rangle) = \{I_1, \dots, I_m\}$ に対して,

$$\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi) \doteq \bigcup_{1 \leq i \leq m} \{I_i - \inf I\}. \quad (5.65)$$

ここで, n 回の遷移が生じたパス ρ に対して, 時間区間 $[0, T_n)$ 中の ψ の充足時間区間集合 $\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi)$ の大きさは,

- $\psi \in AP$ のとき, n 以下,
- $\psi = \neg\psi_1$ のとき, $|\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi_1)| + 1$ 以下,
- $\psi = \psi_1 \vee \psi_2$ のとき, $|\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi_1)| + |\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi_2)|$ 以下,
- $\psi = X^I\psi_1$ のとき, $|\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi_1)|$ 以下,
- $\psi = \psi_1 \mathbf{U}^I \psi_2$ のとき, $|\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi_1)|$ 以下,
- $\psi = \mathbf{Q}_{\sim c}^I \langle \psi_1 \mid \psi_2 \rangle$ のとき, $2 \cdot |\text{Bnd}_{\mathcal{C},\rho}^{<T_n}(\psi)|$ 以下

である。なお、境界点集合 $\text{Bnd}_{\mathcal{C},\rho}^{<T_n}(\psi)$ の大きさは、 $4 \cdot (|\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi_1 \wedge \psi_2)| + |\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi_2)|)$ 以下である。

よって、本節で与えた $\text{SatInt}_{\mathcal{C},\rho}^{<T_n}(\psi)$ 計算アルゴリズムの時間計算量は、検査パス式 ψ の大きさと ρ の時間区間 $[0, T_n)$ 中の遷移回数 n (CTMC の平均的な退出速度に対して線形) に対して線形オーダー、頻度演算子の数に関して高々指数オーダーである。

注意 5.9 (統計的モデル検査で検査可能な更なる拡張確率時間論理) 本統計的モデル検査手法は、有限時間長標本パスがパス式 ψ を満たすか判定できるのであれば、どのような性質 ψ でも検査可能である。そのため、「式の先頭に一つだけ確率パス限量子をもつ」という構文的特徴を持っていれば、頻度演算子 Q 以外の新しい時間演算子をもつ確率時間論理でも本手法と同様のアプローチで検査可能である。 ■

統計的解析

無作為な標本パス生成は Bernoulli 試行であり、パス式 ψ を満たすパスの生起確率 (Bernoulli 試行の成功確率) は一般的な統計的手法を用いて解析できる。そのため、この工程で用いられる手法は統計的 PFLTL モデル検査に特有のものではない。

適用可能な統計的解析手法はいくつもあるが、統計的モデル検査では区間推定 (interval estimation) と逐次確率比検定 [87] (sequential probability ratio test, SPRT) がしばしば用いられている。

- 区間推定：検定に用いる標本数 n と信頼水準 $\alpha \in (0, 1)$ をあらかじめ設定した上で、「 ψ の充足確率」が $1 - \alpha$ の確率で含まれる区間 $I \subseteq [0, 1]$ を推定する。(付録 B.4.1)
- 逐次確率比検定：識別不能幅 $\delta > 0$ 、第一種、第二種過誤率 $\alpha, \beta \in (0, 1)$ をあらかじめ設定した上で、「 $P_{\sim c}[\psi]$ を満たすか否か」を判断するための十分な証拠が得られるまで必要に応じて標本を生成して仮説検定を行う。(付録 B.4.2) 真の確率が $[c - \delta, c + \delta]$ に含まれない場合、誤り率 α, β のもとで正しい仮説を採択する。

注意 5.10 (統計的モデル検査に適用可能な統計的手法) 統計的モデル検査では、Bernoulli 試行の解析に適用可能な統計的手法であればどのような手法も用いることができる。初めて提案された統計的モデル検査 [92] では、式 $P_{\sim c}[\psi]$

を入力検査式とし、SPRT を統計的評価に用いていた。また、確率的モデル検査器 PRISM[58] は SPRT だけでなく区間推定もサポートしている。そして、Bayes 主義的な検定と区間推定を用いる統計的モデル検査手法 [95] も提案されている。

さらに、希少事象に関する統計的評価を効率的に行うことを可能とする重点抽出法などを統計的モデル検査に適用することも可能である [39]。 ■

5.3 適用例

本節では、PFTL モデル検査の有効性を示すため、既存の確率時間論理では記述が困難な定量的性質、及び、通常のモデル検査では検査困難な定量的モデルに対する検査例を示す。ここでは、無限状態 CTMC モデル確率的周期性に関する時間有界 PFTL モデル検査を例として用いる。

5.3.1 被食者-捕食者モデル

被食者と捕食者の個体数の増減関係の決定論的モデルとしてしばしば用いられる Lotka-Volterra 方程式は、同調振動系として知られている。

$$\frac{dx}{dt} = ax - bxy, \quad (5.66)$$

$$\frac{dy}{dt} = -cy + dxy, \quad (5.67)$$

ここで、 $x (> 0)$ は被食者個体数、 $y (> 0)$ は捕食者個体数、 $a (> 0)$ は被食者の自然増加率（正確には真の自然増加速度 a^+ と真の自然減少速度 a^- の差）、 $b (> 0)$ は捕食による被食者の減少率、 $c (> 0)$ は捕食者の自然減少率、 $d (> 0)$ は捕食による捕食者の増加率である。この系は被食者と捕食者の個体数が以下の 1-4 を繰り返すように同調振動するモデルとして知られている。

1. 被食者数の減少により、捕食者数が減少する。
2. 捕食者数の減少により、被食者数が増大する。
3. 被食者数の増大により、捕食者数が増大する。
4. 捕食者数の増大により、被食者数が減少する。

ただし、変数の初期値やパラメータによっては、補食による被食者の絶滅、及び、それに伴う捕食者の絶滅なども起こり得る。

本研究では、Lotka-Volterra 方程式の直観的な意味に基づき、また、両方の個体数が絶滅しにくいように外部系からの被食者の流入速度定数 e (> 0) を導入した、以下のような遷移測度行列 $\mathbf{R}$ と初期分布 P_{init} をもつ CTMC $\langle \mathbb{N}^2, \mathbf{R}, P_{init}, AP, l \rangle$ を考える。ただし、 b と d は元の直観的な意味とは異なり、捕食者の増加を伴わない補食率と捕食者の増加を伴う補食率を表す定数とする。

$$\mathbf{R}(\langle x, y \rangle, \langle x', y' \rangle) = \begin{cases} a^+x + e & \text{if } x' = x + 1 \text{ and } y' = y, \\ a^-x + bxy & \text{if } x' = x - 1 \text{ and } y' = y, \\ cy & \text{if } x' = x \text{ and } y' = y - 1, \\ dxy & \text{if } x' = x - 1 \text{ and } y' = y + 1, \\ \text{exit}(\langle x, y \rangle) & \text{if } x' = x \text{ and } y' = y, \\ 0 & \text{otherwise.} \end{cases} \quad (5.68)$$

$$\text{exit}(\langle x, y \rangle) = -((a^+ + a^-)x + cy + (b + d)xy + e + f). \quad (5.69)$$

ここで、各パラメータを以下のように設定した場合の、確率的シミュレーションを図 5.3.1 に示す²。

$$P_{init}(\langle 800, 800 \rangle) = 1, \quad (5.70)$$

$$a^+ = 2.0 \times 10^{-1}, \quad (5.71)$$

$$a^- = 2.0 \times 10^{-2}, \quad (5.72)$$

$$b = 2.0 \times 10^{-4}, \quad (5.73)$$

$$c = 6.0 \times 10^{-2}, \quad (5.74)$$

$$d = 1.0 \times 10^{-4}, \quad (5.75)$$

$$e = 3.0. \quad (5.76)$$

このモデルは無限状態をもつため、通常モデル検査では検証困難である。なお、式 (5.70)-(5.76) のパラメータ設定の下では各状態 $\langle x, y \rangle$ における被食者数 x の増加速度は $(a^+ - a^-)x - (b + d)xy + e$ であるため、 $x > 0$ かつ $y > 600 + 1000/x$ ($y < 600 + 1000/x$) である状態では、次の状態の x は減少（増加）することが期待される。同様に、捕食者数 y の増加速度は $dxy - cy$ であるため、 $y > 0$

²なお、式 (5.66) と (5.66) で与えられる常微分方程式モデルでは、同様のパラメータに加えて外部系からの被食者の流入速度定数 e (> 0) を導入した場合、捕食者数 600、被食者数 925 に収束する。

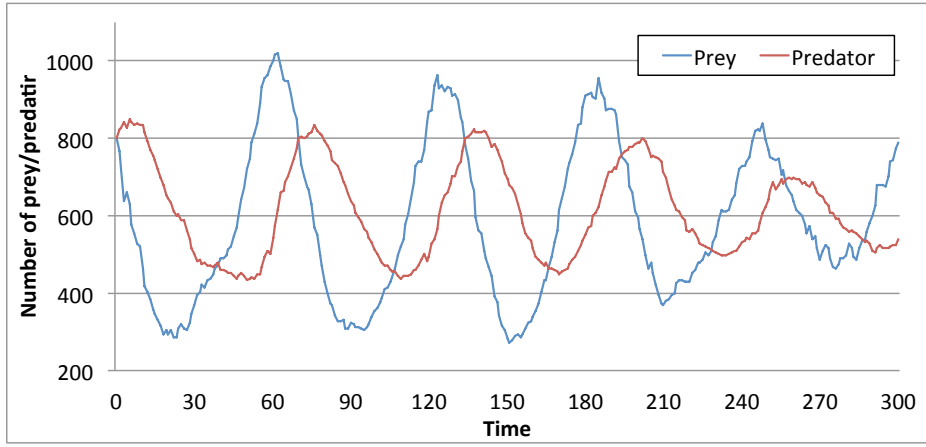


図 5.1: 被食者-捕食者モデルの確率的シミュレーション

かつ $x > 600 - 1/y$ ($x < 600 - 1/y$) である状態では, 次の状態の y は増加 (減少) することが期待される. よって, $\mathcal{C}_{LV}$ のほとんどすべてのパス ρ 上では, 状態列 $state(\rho, 0), state(\rho, 1) \dots$ は概ね点 $\langle 600, 600 \rangle$ の周りを回るように遷移し, その滞在時間列 $time(\rho, 0), time(\rho, 1) \dots$ が 0 に収束することはない. そのため, このパラメータ設定のもとでは $\mathcal{C}_{LV}$ は非爆発的である.

5.3.2 実験 1: 逐次確率比検定による解析

PFTL のパス式では, 図 5.3.1 のような周期的なふるまいはの性質を以下ののような式 ψ で記述できる.

$$\psi_1 = Q_{>q}^I[\Diamond^{=10} \Delta^J \Diamond^{=1} \psi_{cp} | \Diamond^{=1} \psi_{cp}]. \quad (5.77)$$

このパス式は「 ψ_{cp} を丁度満たし始める瞬間と次に ψ_{cp} を丁度満たし始める瞬間の間隔が $J + 10$ に含まれる頻度が q より大きい」という周期性を表している. ここでは, このチェックポイントとなる性質 ψ_{cp} を

$$\psi_{cp} = \Box^{\leq 10} y > x \quad (5.78)$$

とする. ここで, $y > x$ という命題は「捕食者数 y が被食者数 x より大きい」ことを表すものとする. つまり, パス式 ψ_1 は, 「捕食者数 y が被食者数 x を丁度安定して越える瞬間の間隔が $J + 10$ に含まれる頻度が q より大きい」という性質を表すと解釈できる.

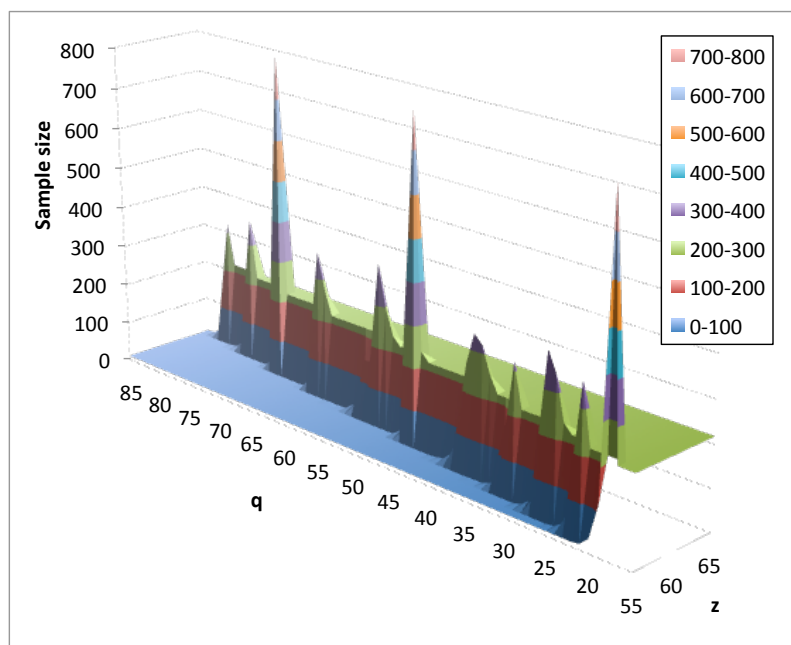


図 5.2: $P_{>0.95}[Q_{>q}^{[0,10^5]}[\diamond=10\triangle^{[0,z]}\diamond=1\psi_{cp}|\diamond=1\psi_{cp}]]$ の逐次確率比検定に要した標本数. ただし, 識別不能幅, 第一種誤り率, 第二種誤り率は 0.01.

識別不能幅, 第一種誤り率, 第二種誤り率を 0.01, $I = [0, 10^5]$, $J+10 = [10, z]$ として $q \in \{10, 15, \dots, 85, 90\}$, $z \in \{55, 56, \dots, 64, 65\}$ を変化させたときの $P_{>0.95}[\psi]$ の逐次確率比検定に要した標本数を図 5.3.2 に示す. 図中の青い領域の標本数を与えた検定では, 生成されたすべて標本パスは ψ_1 を満たさず, ψ_1 を満たすパスの真の生起確率が 0.95 よりも小さいと判定する為に必要な最小個数 (11 個) の標本で検定できた. また, 図中の緑の領域の標本数を与えた検定では, 生成されたすべて標本パスは ψ_1 を満たし, ψ_1 を満たすパスの真の生起確率が 0.95 よりも大きいと判定する為に必要な最小個数 (218 個) の検定で検定できた. 標本数が多い検定は, ψ_1 を満たすパスの真の生起確率が識別不能領域 $[0.94, 0.96]$ に含まれているか, それに近い値をもっていることを表す. 以上から検査する確率閾値と真の確率の差が小さくない場合には, 少ない標本で検証できることが分かる. また, この結果から「捕食者数 y が被食者数 x を丁度安定して越える瞬間」の間隔がおよそ 60 単位時間であることが確かめられる.

5.3.3 実験2：区間推定による解析

次に、「捕食者数が被食者数を丁度安定して越える瞬間の捕食者数」が「次に捕食者数が被食者数を丁度安定して越える瞬間までの時間」に与える影響について考える。

ここでは、以下のような式について調べる。

$$\psi_2 = Q_{>q}^I[\Diamond^{=10}\Delta^J\Diamond^{=1}\psi_{cp}|\Diamond^{=1}(\psi_{cp} \wedge x \geq 800)]. \quad (5.79)$$

$$\psi_3 = Q_{>q}^I[\Diamond^{=10}\Delta^J\Diamond^{=1}\psi_{cp}|\Diamond^{=1}(\psi_{cp} \wedge x < 800)]. \quad (5.80)$$

ここで、命題 $x \geq 800$ と $x < 800$ は「被食者数が800以上」、「被食者数が800より小さい」を表すとする。これらの式は、「捕食者数が被食者数を丁度安定して越える瞬間で $x \geq 800$ ($x < 800$) を満たすとき、次に捕食者数が被食者数を丁度安定して越える瞬間が $J + 10$ に含まれる頻度が q より大きい」という性質を表す。そして、 $I = [0, 10^5]$, $J + 10 = [55, 65]$, $q > 0.55$ に対して、標本数 1000 で ψ_2 の充足確率を信頼水準 0.05 で区間推定したときには、信頼区間 $[0.358, 0.418]$ が得られた。すなわち、真の ψ_2 の充足確率がこの区間に含まれる確率 0.95 である。また、 $I = [0, 10^5]$, $J + 10 = [55, 65]$, $q > 0.85$ に対して、標本数 1000 で ψ_3 の充足確率を信頼水準 0.05 で区間推定したときには、信頼区間 $[0.585, 0.645]$ が得られた。すなわち、真の ψ_3 の充足確率がこの区間に含まれる確率 0.95 である。

この結果から、捕食者数が被食者数を丁度安定して越える瞬間に $x < 800$ である場合は $x \geq 800$ である場合と比べて「次に捕食者数が被食者数を丁度安定して越える瞬間が 55 から 65 単位時間に含まれる」である頻度が有意に高いと判断できる。

考察

この「 ψ_{cp} が立ち上がる時点の状態 $\langle x, y \rangle$ 」によって、次に ψ_{cp} が立ち上がるまでの時間間隔が大きく異なる」という性質は、この被食者-捕食者モデル $\mathcal{C}_{LV}$ の確率的周期性が CSL では記述困難であることを示唆している。

CSL では、時間演算子を確率パス限量子の直後でしか使うことができないため、一つの時系列上の複雑な性質を直接的に記述することはできない。また、「特定の t 単位時間後に丁度ある性質 ψ を満たす ($\Delta^{=t}\psi$)」確率についても意味のある言及はできない。よって、確率的周期性に関する定量的性質を CSL で記述するためには、「チェックポイントとなる (いくつかの) 状態群」を CSL

状態式で特徴付け, 「その状態群間をある時間間隔で巡回する確率がある閾値条件を常に満たす」ことを CSL パス式で表現しなくてはならない. そのような状態式とパス式は, 例えば, 以下のような式である.

$$\varphi_{cp} = (y \leq x) \wedge P_{>p_1}[\Box^{[0,10]+1}y > x], \quad (5.81)$$

$$\psi_4 = \Box^I(\varphi_{cp} \rightarrow P_{>p_2}[\Diamond^{=10}P_{>p_3}[\Delta^{J_1-10}\varphi_{cp}]]). \quad (5.82)$$

直観的には, φ_{cp} は「現在, 捕食者数 y が被食者数 x 以下であり, 1 単位時間後には y が x を安定して越える確率が p_1 より大きい」ことを表し, ψ'_{cyc} は「区間 I では常に, φ_{cp} を満たす状態から” φ_{cp} を初めて満たすまでの時間が $J_1 - 10$ に含まれるパスに分岐する確率が p_3 より大きい状態”へ 10 単位時間後に到達する確率が p_2 より大きい」ことを表す. ここで, ψ'_{cyc} が成り立つならば「 φ_{cp} を満たす状態から再び φ_{cp} を満たすまでの間隔が J_1 ($\subseteq [10, \infty)$) に含まれるパスに分岐する確率が $p_1 \cdot p_2$ より大きい」が, その逆は必ずしも成立しない. 同様の性質は以下のように記述することも可能である.

$$\varphi'_{cp} = (y \geq x) \wedge P_{>p_4}[\Box^{[0,10]+1}y < x], \quad (5.83)$$

$$\psi_5 = \Box^I((\varphi_{cp} \rightarrow P_{>p_5}[\Delta^{J_2}\varphi'_{cp}]) \wedge (\varphi'_{cp} \rightarrow P_{>p_6}[\Delta^{J_3}\varphi_{cp}])). \quad (5.84)$$

直観的には, φ'_{cp} は「現在, 被食者数 x が捕食者数 y 以下であり, 1 単位時間後には x が y を安定して越える確率が p_1 より大きい」ことを表し, ψ_4 は「“ φ_{cp} (φ'_{cp}) が規定する状態群” から “ φ'_{cp} (φ_{cp}) が規定する状態群” へ初めて到達する時間が J_2 (J_3) に含まれる確率が p_5 (p_6) より大きい」ことを表す. 何れにしても, CSL で確率的周期性を記述するためには, 「チェックポイントとなる性質が立ち上がる時間の間隔」を直接的には記述できないだけでなく, 多くの確率閾値パラメータ (ここでは $p_1, \dots, p_6$) に対する値を具体的に割り当てる必要がある. そして, それらは CSL 状態式 φ_{cp} や φ'_{cp} のみたすすべての状態が共通に満たす弱い閾値条件となるようにしか与えることができない. よって, CSL では確率的周期性を適度な強さで簡潔に表現することはできない.

それに対して, PFLTL 式 ψ_1 が表す性質は, ψ_{cp} を満たすすべての時刻の状態の「共通に満たす定量的性質」ではなく「その出現頻度に応じて重み付けされた定量的性質」であり, 少ないパラメータによって適度な強さで確率的周期性を表現している.

第III部

平均利得時間論理とその検証手法

第6章 平均利得線形時相論理 LTL^{mp}

これまでの時間論理を用いた検証及び自動合成では、システムが満たすべき機能に関する定性的仕様がしばしば注目されてきた。例えば、「常に、要求 *request* はいつか処理 *serve* され、それまでの間待ち状態 *wait* である」(式 (6.1)) のようなイベント生起順序に関する制約である。

$$\Box(request \rightarrow (wait \cup serve)). \quad (6.1)$$

しかしながら、このような定性的仕様に対する検証や自動合成は、現実的なシステム開発ではしばしば有効でない。例えば、式 (6.1) を含むような LTL 式仕様からの自動合成では、「既に安全に要求を処理 *serve* できる状況であるにも関わらず待ち *wait* 続ける」ような非効率的にふるまうリアクティブシステム (RS) が生成されてしまう可能性がある。もちろん、安全に処理できる状況でもすぐに処理しない方が結果的にシステム全体の性能の向上につながる可能性もある。定性的仕様からの RS の自動合成においては、合成された RS の質を測る明確な尺度が「内部状態制御の単純さ (状態数)」しかなく、Acacia+[2] などの既存の実現可能性判定/合成器は状態数が少ない RS を合成しようとする。そのため、状況に応じて必要なイベントを適切なタイミングで必要なだけ生起させるような臨機応変な RS ではなく、非効率的で杓子定規な RS が合成されてしまうことになる。従って、形式的な検証や自動合成を現実的なシステム開発に効果的に適用するためには、システムが満たすべき性能に関する定量的仕様にも注目する必要がある。

そこで本研究では、システム動作におけるコスト/利得やイベント生起頻度の平均的な性質を捉えることができる**平均利得**の概念を導入した**平均利得線形時相論理**LTL^{mp} (LTL with Mean-Payoff constraints) を開発した [81, 97]。この LTL^{mp} は、LTL に平均利得に関する性質を規定する**平均利得式** (mean-payoff formula) を追加した平均利得線形時間論理である。ここで、平均利得式とは、モデル実行上で与えられる利得や論理の中で規定された利得に関する平均利

得制約を表す式である。それによって、長期的な頻度や平均コストなどの性質を反映した仕様を規定できる。これまでの利得拡張時間論理と異なる点は、ふるまいの利得に関する性質をモデルに依存しない形で規定できる点である。LTL や PCTL, CSL に利得（報酬）の概念を導入した論理は既にいくつか提案されている [13, 10, 19]。これらの論理ではモデルのふるまい上の平均利得や利得の期待値などに関する性質を記述できる。ただし、これらの論理では利得はモデルの状態や遷移に対して定義されているものであり、モデル検査のための仕様しか記述できなかった。一方 LTL^{mp} では、ふるまいの時間的性質に対して与えられる利得を論理の中で定義できる。例えば、「ふるまい上で φ が成り立つ時間で 1, ψ が成り立つ時間で 2 の利得を与える」ということを設定できる。それにより、モデル検査だけでなく充足可能性/実現可能性判定といった検証のための仕様記述言語として用いることができる。

本章では、 LTL^{mp} の構文と意味論の定義を与え、また、 LTL^{mp} の表現力について論じる。

そして、第7章ではこの LTL^{mp} で規定できる言語を認識できるオートマトンについて、第8章では LTL^{mp} のモデル検査と充足可能性判定、最適化手法について述べる。また、第9章では、 LTL^{mp} 実現可能性判定手法について述べる。

本章の構成

6.1 節で平均利得線形時間論理 LTL^{mp} の構文の定義を与える。そして、6.2 節で LTL^{mp} の意味論と、いくつかの基本的な性質について述べる。6.3 節で LTL^{mp} により記述可能な性質をいくつか紹介する。最後に、6.4 節で、 LTL^{mp} の表現力について論じる。

6.1 構文

本節では LTL^{mp} の構文を与える。

LTL^{mp} は通常の LTL の構文に加えて、平均利得に関する閾値条件を表す平均利得式 (mean-payoff formula) $MP(t) \triangleright 0$ ($\triangleright \in \{>, \geq\}$) をもつ。ここで、 t は各時間で発生する利得を規定する**利得項** (payoff term) であり、実定数 $c \in \mathbb{R}$, 利得系モデル上の利得を参照する**自由変数** (free variable) $v_1, \dots, v_n \in V$, 時間的性質 φ の充足に対する利得を表す**特性変数** (characteristic variable) 1_φ , 及び、それらの代数演算 (和 $+$, 差 $-$, 積 $\cdot$) からなる。

表 6.1: 語 $\sigma = \{p, q\} \{p\} \{q\} \emptyset \{q\} \{p\} \{p, q\} \dots$ に対する利得項に関する利得トレース.

利得項	利得トレース							
$\mathbf{1}_p$	1	1	0	0	0	1	1	...
$\mathbf{1}_q$	1	0	1	0	1	0	1	...
$\mathbf{1}_{p \cup q}$	1	1	1	0	1	1	1	...
v	-1	2	0	2	1	3	-2	...
2	2	2	2	2	2	2	2	...
$2 + \mathbf{1}_p$	3	3	2	2	2	3	3	...
$v \cdot \mathbf{1}_q$	-1	0	0	0	1	0	-2	...
$-\mathbf{1}_p$	-1	-1	0	0	0	-1	-1	...

定義 6.1 (LTL^{mp} の構文 [81]) AP を原子命題の集合, V を自由変数の集合とする. このとき, LTL^{mp} は以下のように帰納的に定義される.

$$\begin{aligned}
\varphi &::= p \mid \theta \mid \neg\varphi \mid \varphi \vee \varphi \mid X\varphi \mid \varphi U \varphi, \\
\theta &::= \text{MP}(t) \triangleright c \mid \neg\theta, \\
t &::= c \mid v \mid \mathbf{1}_\varphi \mid t + t \mid -t \mid t \cdot t.
\end{aligned}$$

ただし, $p \in AP$, $c \in \mathbb{R}$, $v \in V$, $\triangleright \in \{>, \geq\}$ とする. ■

直観的には, 特性変数 $\mathbf{1}_\varphi$ はその時点で φ を満たすとき 1, そうでないとき 0 を与える項である. φ が時間演算子を含む場合, それぞれの時間点において未来のふるまいの性質に依存した値が割り当てられる. この意味で特性変数は束縛されている. 一方, 自由変数 v は, 利得系 $\mathcal{P} = \langle S, \hookrightarrow, S_{\text{init}}, AP, l, w \rangle$ のパス ρ 上の各時間で与えられる利得, すなわち, そのときの状態に対する重みを参照するために用いる. 具体的には, 自由変数 v と重み関数 w の i 番目次元の成分 $w[i]$ が対応している場合, 時間 j で割り当てられる v の値は, ρ 上の i 次元成分利得トレース $\text{payoff}_{w[i]}(\rho)$ の j 番目の要素 $w[i](\rho[j])$ に対応する. そして, 利得項間の演算は各点的演算である. ここで, 語 $\sigma = \{p, q\} \{p\} \{q\} \emptyset \{q\} \{p\} \{p, q\} \dots$ に対してそれぞれの時間で利得項に割り当てられる値の列, すなわち, その利得項に関する利得トレースの例を表 6.1 に示す. ただし, ここでは変数 v に対応するモデルのパス上の利得トレースは $(-1) \ 2 \ 0 \ 2 \ 1 \ 3 \ (-2) \dots$ であるものとする.

なお、利得項 t 中に現れる特性変数 1_φ の台を規定する式 φ のすべてが AP 上の Boolean 結合で与えられている場合、任意の時間 j での利得は、 $\mathcal{P}$ のパス ρ 上の状態 $\rho[j]$ に対する重み $w(\rho[j])$ と文字 $l(\rho[j])$ のみに依存し、未来のふるまいの時間的性質には依存しない。そのため、そのような t は**未来非依存**であるという。

各時間の利得を規定する利得項 t に対して、**平均利得項** $MP(t)$ は利得項 t に関する平均利得トレース（利得トレースの平均トレース）の下極限、平均利得式 $MP(t) \triangleright c$ はその閾値条件を表す。ここで、その利得項 t が平均利得式を含む式の特性変数（例えば $1_{MP(t) \triangleright c}$ ）を持たないとき、その平均利得式は**単純**であるという。また、単純平均利得式とその否定を**平均利得リテラル**と呼ぶ。

平均利得式に関しては、以下のような略記を用いてもよいものとする。

$$MP(t) \leq c \equiv \neg MP(t) > c, \quad (6.2)$$

$$MP(t) < c \equiv \neg MP(t) \geq c, \quad (6.3)$$

$$\overline{MP}(t) \sim c \equiv MP(-t) \sim -c. \quad (6.4)$$

ただし、 $\sim \in \{>, \geq, <, \leq\}$ であり、また、 $\sim$ が $>$ ($\geq$, $<$, $\leq$) のとき、 $\sim$ は $<$ ($\leq$, $>$, $\geq$) である。つまり、平均利得リテラルは $MP(t) \sim 0$ の形をした、平均利得式を含む式の特性変数をもたない式である。また、平均利得項 $\overline{MP}(t)$ は、直観的には、利得項 t の平均利得トレースの上極限を表す。

6.2 意味論

次に、 LTL^{mp} の意味論を与える。これは LTL と同様に語に対する意味として定義する。また、各時間で自由変数 $v \in V$ の値は、変数割当 $\alpha : V \rightarrow \mathbb{R}^\omega$ を通して割り当てられる。

定義 6.2 (LTL^{mp} の意味論 [81]) AP を原子命題集合、 V を自由変数集合、 $\alpha : V \rightarrow \mathbb{R}^\omega$ を V に対する変数割当とする。アルファベット 2^{AP} 上の無限長語 $\sigma = a_0 a_1 \dots \in (2^{AP})^\omega$ と AP 上の LTL^{mp} 式 φ の α の下での充足関係 $\models$ は以下のように帰納的に定義される。

$$\begin{aligned} \sigma, \alpha, i \models p &\Leftrightarrow p \in a_i \text{ for } p \in AP, \\ \sigma, \alpha, i \models \neg \varphi &\Leftrightarrow \sigma, \alpha, i \not\models \varphi, \\ \sigma, \alpha, i \models \varphi_1 \vee \varphi_2 &\Leftrightarrow \sigma, \alpha, i \models \varphi_1 \text{ or } \sigma, \alpha, i \models \varphi_2, \end{aligned}$$

$$\begin{aligned}
\sigma, \alpha, i &\models X\varphi \Leftrightarrow \sigma, \alpha, i+1 \models \varphi, \\
\sigma, \alpha, i &\models \varphi_1 \cup \varphi_2 \Leftrightarrow \exists j \geq i. (\sigma, \alpha, j \models \varphi_2 \text{ and } \forall k \in [i, j). \sigma, \alpha, k \models \varphi_1), \\
\sigma, \alpha, i &\models \mathbf{MP}(t) \triangleright c \Leftrightarrow \llbracket \mathbf{MP}(t) \rrbracket_\sigma^\alpha[i] \triangleright c.
\end{aligned}$$

ただし、平均利得項 $\mathbf{MP}(t)$ と利得項 t の σ と α の下での解釈 $\llbracket \cdot \rrbracket_\sigma^\alpha$ は以下のように帰納的に定義される 1 次元トレースである。

$$\begin{aligned}
\llbracket \mathbf{MP}(t) \rrbracket_\sigma^\alpha[i] &= \liminf_{n \rightarrow \infty} \frac{1}{n} \sum_{j=0}^n \llbracket t \rrbracket_\sigma^\alpha[i+j] \\
\llbracket c \rrbracket_\sigma^\alpha[i] &= c \text{ for } c \in \mathbb{R}, \\
\llbracket v \rrbracket_\sigma^\alpha[i] &= \alpha(v)[i] \text{ for } v \in V, \\
\llbracket \mathbf{1}_\varphi \rrbracket_\sigma^\alpha[i] &= \begin{cases} 1 & \text{if } \sigma, \alpha, i \models \varphi, \\ 0 & \text{otherwise,} \end{cases} \\
\llbracket t_1 + t_2 \rrbracket_\sigma^\alpha[i] &= \llbracket t_1 \rrbracket_\sigma^\alpha[i] + \llbracket t_2 \rrbracket_\sigma^\alpha[i], \\
\llbracket -t \rrbracket_\sigma^\alpha[i] &= -\llbracket t \rrbracket_\sigma^\alpha[i], \\
\llbracket t_1 \cdot t_2 \rrbracket_\sigma^\alpha[i] &= \llbracket t_1 \rrbracket_\sigma^\alpha[i] \cdot \llbracket t_2 \rrbracket_\sigma^\alpha[i].
\end{aligned}$$

また、 $i = 0$ や $V = \emptyset$ のとき、 $\sigma, \alpha, i \models \varphi$ と $\llbracket t \rrbracket_\sigma^\alpha$ から i や α を省略する。 ■

充足可能性

LTL^{mp} 文 φ に対して、 $\sigma \models \varphi$ であるような語 σ が存在するとき、 φ は充足可能であるという。

また、LTL^{mp} 文 φ で定義される言語 $\{\sigma \in (2^{AP})^\omega \mid \sigma \models \varphi\}$ を $L(\varphi)$ 、LTL^{mp} 文で定義できる言語のクラスを $\mathcal{L}(\text{LTL}^{\text{mp}})$ と書く。

モデル検査

d -次元利得系 $\mathcal{P} = \langle S, \rightarrow, S_{\text{init}}, AP, \Lambda, w \rangle$ について、変数対応関数 $\beta : V \rightarrow \mathbb{N}_{>0}$ を通して自由変数 $v \in V$ を重み $w[\beta(v)]$ に対応させたときのモデル充足関係 $\mathcal{P}, \beta \models \varphi$ は、以下のように定義する。

$$\mathcal{P}, \beta \models \varphi \Leftrightarrow \forall \rho \in \text{Paths}^{\mathcal{P}}. (\text{word}_\Lambda(\rho), \alpha_{w, \beta}^\rho \models \varphi). \quad (6.5)$$

ただし、 φ 中に出現するすべての変数 $v \in V$ について、 $1 \leq \beta(v) \leq d$ かつ

$$\alpha_{w, \beta}^\rho(v) = \text{payoff}_{w[\beta(v)]}(\rho) \quad (6.6)$$

とする。なお、以降では変数対応関数 β は固定されているものとし、 $\mathcal{P}, \beta \models \varphi$ を単に $\mathcal{P} \models \varphi$ と書く。

確率的モデル検査/充足確率

d -次元利得付き DTMC $\mathcal{D} = \langle S, \mathbf{P}, P_{\text{init}}, AP, \Lambda, w \rangle$ について、変数対応関数 $\beta : V \rightarrow \mathbb{N}_{>0}$ を通して自由変数 $v \in V$ を重み $w[\beta(v)]$ に対応させたときの LTL^{mp} 式 φ を満たすパスが生起する確率 $\text{Prob}_{(s)}^{\mathcal{D}}(\varphi)$ は、以下のように与えられる。

$$\text{Prob}_{(s)}^{\mathcal{D}}(\varphi) = \Pr_{(s)}^{\mathcal{D}}(\{\rho \in \text{Paths}^{\mathcal{D}}. (\text{word}_{\Lambda}(\rho), \alpha_{w, \beta}^{\rho} \models \varphi)\}). \quad (6.7)$$

また、モデル充足関係 $\mathcal{D} \models \varphi$ は、以下のように定義する。

$$\text{Prob}_{(s)}^{\mathcal{D}} \models \varphi \Leftrightarrow \text{Prob}_{(s)}^{\mathcal{D}}(\varphi) = 1 \quad (6.8)$$

そして、充足関係 $\mathcal{D} \models \varphi$ が成り立つか否か判定することを LTL^{mp} 確率的モデル検査、確率 $\text{Prob}^{\mathcal{D}}(\varphi)$ を求める問題を LTL^{mp} 充足確率問題と呼ぶ。

基本的な性質

ここで、 LTL^{mp} のいくつかの基本性質について述べる。

ここで LTL^{mp} の利得項について、任意の語 σ と変数割当 α に対して $\llbracket t_1 \rrbracket_{\sigma}^{\alpha} = \llbracket t_2 \rrbracket_{\sigma}^{\alpha}$ であるとき、 t_1 と t_2 は等価であり、 $t_1 \equiv t_2$ と書くこととする。このとき定義 6.2 から、以下の性質が成り立つ。

- 利得項に関する可換環の公理,
- 特性変数に関する以下の補数 (式 (6.9)), 包除 (式 (6.10)), 独立 (式 (6.11)) の原理.

$$\mathbf{1}_{\neg \varphi} \equiv 1 - \mathbf{1}_{\varphi}, \quad (6.9)$$

$$\mathbf{1}_{\varphi_1 \vee \varphi_2} \equiv \mathbf{1}_{\varphi_1} + \mathbf{1}_{\varphi_2} - \mathbf{1}_{\varphi_1 \wedge \varphi_2}, \quad (6.10)$$

$$\mathbf{1}_{\varphi_1 \wedge \varphi_2} \equiv \mathbf{1}_{\varphi_1} \cdot \mathbf{1}_{\varphi_2}. \quad (6.11)$$

ここで、下極限は、その定義から、一般にプレフィックスには依存しない値をもつ。そのため、任意の語 σ と変数割当 α に対して、トレース $\llbracket \text{MP}(t) \rrbracket_{\sigma}^{\alpha}$ は

固定された値からなる列である。よって、任意の平均利得式 θ に関して以下の関係が成り立つ。

$$\theta \Leftrightarrow \neg \theta \Leftrightarrow \neg \theta. \quad (6.12)$$

これにより、任意の語 σ と変数割当 α に対して、平均利得リテラル θ は「常に成り立つ」または「常に成り立たない」。従って、平均利得リテラル θ を部分式として持つ式 φ について、以下の関係が成り立つ [19].

$$\varphi \Leftrightarrow (\varphi[\top/\theta] \wedge \theta) \vee (\varphi[\perp/\theta] \wedge \neg \theta). \quad (6.13)$$

式 $\varphi[\top/\theta]$ と $\varphi[\perp/\theta]$ は平均利得リテラル θ を含まない式となっているため、変形を繰り返すことで φ と等価な式を通常の LTL 式 ψ と平均利得リテラル θ の Boolean 結合で表現することができる。

また、下極限の基本的な性質から、以下の条件が成り立つ。

$$\text{MP}(t) \sim c \Leftrightarrow \text{MP}(t - c) \sim 0. \quad (6.14)$$

従って、式 (6.13) と式 (6.14) より、 LTL^{mp} の構文は表現力を失うことなく以下のように制限できる。

定義 6.3 (平均利得標準形) LTL^{mp} 式が以下のような形をもつとき、**平均利得標準形** (mean-payoff normal form, MPNF) と呼ぶ。

$$\bigvee_i (\psi_i \wedge \bigwedge_j \theta_{ij}).$$

ただし、

- それぞれの ψ_i は通常の LTL 式,
- それぞれの θ_{ij} は閾値 0 の平均利得リテラル $\text{MP}(t_{ij}) \sim 0$,
- それぞれの t_{ij} は $\sum (c_{ijk} \cdot \mathbf{1}_{\varphi_{ijk}} \cdot \prod v_{ijkl})$ の形をもつ利得項である。 ■

定理 6.4 (MPNF 式への変形) n 個の平均利得式をもつ任意の LTL^{mp} 式 φ は、高々 2^n 個の選言肢とそれぞれの選言肢について高々 n 個の平均利得リテラルをもつ等価な MPNF 式に変形できる。 ■

証明. LTL^{mp} の意味論 (定義 6.2) と式 (6.14) より, 任意の平均利得リテラル $MP(t) \sim c$ は閾値 0 の平均利得式 $MP(t - c) \sim 0$, 特性変数 $\mathbf{1}_\top$ は定数 1 と等価である. さらに, 可換環の公理及び独立の原理 (式 (6.11)) が成り立つことから, 利得項 $t - c$ は, $\sum_i (c_i \cdot \mathbf{1}_{\varphi_i} \cdot \prod_j v_{ij})$ という形で表現することができる. そして, 式 (6.13) より, n 個の平均利得式をもつ任意の LTL^{mp} 式 φ について以下の条件が成り立つ.

$$\varphi \Leftrightarrow \bigvee_{\langle b_1, \dots, b_n \rangle \in \{\top, \perp\}^n} \left(\varphi[b_m/\theta_m \text{ for } 1 \leq m \leq n] \wedge \bigwedge_{1 \leq m \leq n} (b_m \leftrightarrow \theta_m) \right). \quad (6.15)$$

ここで, 任意の式 f に関して, 明らかに $(\top \leftrightarrow f) \Leftrightarrow f$ かつ $(\perp \leftrightarrow f) \Leftrightarrow \neg f$ である. 従って, 任意の LTL^{mp} 式は, 高々 2^n 個の選言肢からなり, かつ, それぞれの選言肢に関して高々 n 個の平均利得リテラルをもつ等価な MPNF 式に変換可能である. $\square$

注意 6.5 (平均利得リテラルの極性) 平均利得リテラル $\theta = MP(t) \sim c$ は, その極性によって大きく性質が異なる.

単純平均利得式 $MP(t) \triangleright c$ は, 「利得項 t の平均利得トレースの下極限 $[[MP(t)]]_\sigma^\alpha[0]$ が閾値条件 $\triangleright c$ を満たす」という性質を表す. ここで, 上極限は下極限以上であるため, t の平均利得トレースの上極限も閾値条件 $\triangleright c$ を満たしている. つまり, 単純平均利得式 $MP(t) \triangleright c$ は, 「その任意の近傍を “ t の平均利得トレース” が無限にしばしば通るような点のすべてが閾値条件 $\triangleright c$ を満たす」という全称的な閾値条件を表している.

一方, 単純平均利得式の否定 $MP(t) \triangleleft c$ (ただし $\triangleleft \in \{<, \leq\}$) は, 「 t の平均利得トレースの下極限 $[[MP(t)]]_\sigma^\alpha[0]$ が閾値条件 $\triangleleft c$ を満たす」という性質を表す. このとき, その平均利得トレースの上極限は閾値条件 $\triangleleft c$ を満たすとは限らない. つまり, 単純平均利得式の否定 $MP(t) \triangleleft c$ は, 「その任意の近傍を “ t の平均利得トレース” が無限にしばしば通るような点のすべてが閾値条件 $\triangleleft c$ を満たすとは限らないが, いくつかの点は閾値条件を満たす」という存在的な閾値条件を表している. $\blacksquare$

注意 6.6 (LTL^{lim} [19] との関係) [19] で提案されている LTL^{lim} は, LTL^{mp} の利得項 t を自由変数 V の要素のみに制限したフラグメントである.

$$t ::= v \in V.$$

LTL^{lim} は, 定量的モデル検査のための仕様記述言語としては有効であるが, 充足可能性などの仕様自体を検証には不向きである. なぜならば, 平均利得式は

平均の上極限/下極限に関する閾値条件であり、平均利得式部分については代数的に明らかに矛盾しているか、そうでないときには、それぞれを真にする変数割当が自明に存在するためである。 ■

注意 6.7 (LTL_{MP}[17] との関係) [17] で提案されている LTL_{MP} は、LTL^{mp} を以下のように制限したフラグメントである。

$$\begin{aligned}\varphi &::= \psi \mid \text{MP}(t) \geq c \mid \varphi \wedge \varphi, \\ t &::= \mathbf{1}_p \mid \mathbf{1}_{\neg p} \mid t + t \mid c \cdot t.\end{aligned}$$

ただし、 $p \in AP$, $\psi \in \text{LTL}$.

[17] では、LTL_{MP} で記述された仕様について、任意の環境に対して有限メモリで実現可能であるかを判定する手法を提案している。 ■

6.3 仕様記述例

本節では、平均利得式で記述できる定量的性質をの例をいくつか示す。

頻度

LTL^{mp} では、時間式 φ の充足頻度 (φ が成り立つ時間点の割合) に関する性質は、平均利得式によって簡単に記述できる。例えば、以下の式は「 φ を満たしている頻度の下極限が 0 より大きい」という定量的回帰性を表す。

$$\text{MP}(\mathbf{1}_\varphi) > 0. \quad (6.16)$$

この式は、定性的回帰性「無限にしばしば φ が成り立つ ($\Box\Diamond\varphi$)」とは意味が異なる。MP($\mathbf{1}_\varphi$) > 0 が成り立つならば $\Box\Diamond\varphi$ が成り立つが、一方で、 $\Box\Diamond\varphi$ は「無限にしばしば φ が成り立つがその充足頻度が 0 に収束する」語によっても満たされる。つまり、以下のような含意関係は成り立つが、逆は必ずしも成り立たない。

$$\text{MP}(\mathbf{1}_\varphi) > 0 \rightarrow \Box\Diamond\varphi. \quad (6.17)$$

また、利得項 $(\mathbf{1}_{\varphi_1} - c) \cdot \mathbf{1}_{\varphi_2}$ (ただし、 $0 < c < 1$) を持つ平均利得式によって、「条件 φ_2 が成り立つ時間のうち φ_1 も成り立つ条件付き頻度」を記述することができる。例えば、以下のような式は「リクエスト頻度の下極限が 0 でない、

かつ、リクエストに対して次ステップで応答する条件頻度の下極限が c より大きい」ことを表す.

$$\text{MP}((\mathbf{1}_{\text{Xresponse}} - c) \cdot \mathbf{1}_{\text{request}}) > 0. \quad (6.18)$$

なお、この式は離散時間 PFTL パス式 $Q_{>c}[\text{Xresponse}|\text{request}]$ とよく似た性質を表現しているが、「request が無限にしばしば成り立たない」ときには真となる $Q_{>c}[\text{Xresponse}|\text{request}]$ とは異なり、「request の充足頻度の下極限が 0 である」ときには偽となる.

また、以下のような式で「 φ_1 の充足頻度は φ_2 の充足頻度の c 倍より多い」ことを表すことができる.

$$\text{MP}(\mathbf{1}_{\varphi_1} - c \cdot \mathbf{1}_{\varphi_2}) > 0. \quad (6.19)$$

平均コスト

利得やコストを考慮したモデル化では、命題の充足とイベントやアクションの発生を同一視し、イベントが発生する度（連続時間モデルでは単位時間毎）に一定の利得が得られるものとして形式化することが多い. そして、長期平均利得をしばしば平均利得の下極限と定義し、それをふるまいの質として評価している. LTL^{mp} では、利得項 $t = \sum c_i \cdot \mathbf{1}_{p_i}$ に対する平均利得式によりそれらの性質を記述できる. ただし、 p_i はイベント e_i に対応した命題、 c_i は e_i の利得あるいはコストとする. 例えば、「起動 *booting* に 6、駆動 *operating* に 4、停止 *shutdown* に 5 のコストがかかるとき、長期的な平均コストの上界が 3 より小さい」は以下のように記述できる.

$$\overline{\text{MP}}(6 \cdot \mathbf{1}_{\text{booting}} + 4 \cdot \mathbf{1}_{\text{operating}} + 5 \cdot \mathbf{1}_{\text{shutdown}}) < 3. \quad (6.20)$$

また、イベントの立ち上げ/立ち下げコストも特性変数 $\mathbf{1}_{p_i \wedge \text{X} \neg p_i} / \mathbf{1}_{\neg p_i \wedge \text{X} p_i}$ で表現できる.

陰的イベンチャリティ性

強い until 演算子やその肯定的な省略形である $\diamond$ 演算子は、「いつかある性質が満たされなければならない」という**イベンチャリティ性** (eventuality property) に関する性質を表現する.

LTL リアクティブシステム仕様ではしばしば、強い until 演算子を用いて、以下のような「要求 *request* に対して、いつか応答 *response* を返す」という応答性（例 3.27）に関する制約を記述する。

$$\Box(request \rightarrow \Diamond response). \quad (6.21)$$

LTL^{mp} では、平均利得式を用いることで同様の性質を定性的なイベンチャリティを用いずに記述できる。例えば、以下のような式である。

$$\Box(request \rightarrow (wait \text{ W } response)) \wedge MP(1_{\neg wait}) > 0. \quad (6.22)$$

このとき、平均利得式により $\neg wait$ が無視できない頻度で生起しなければならない。このように弱い until 演算子 W と平均利得式を用いることで、強い until 演算子 U（または $\Diamond$ ）で記述しなければならなかった制約と同じ意図を反映した性質を表現できる。

6.4 LTL^{mp} vs. ω -Kleene 閉包言語

ω -正規言語のクラス $\mathcal{L}(\omega\text{-REG})$ は、Büchi/Rabin 受理条件などをもつ有限状態 ω オートマトンが認識できる言語のクラスであり、最もよく研究されてきた無限長言語のクラスである。LTL の規定できる言語は star-free（あるいは counter-free） ω -正規言語とも呼ばれ、 ω -正規言語のクラスの真部分集合である。LTL の表現力は ω -正規言語より小さいものの、安全性や活性などの様々な定性的性質を直観的に記述できることから仕様記述言語としてしばしば用いられてきた。本節では、LTL の平均利得拡張である LTL^{mp} と ω -正規言語の関係について論じる。

LTL^{mp} は LTL に平均利得式を追加した時間論理であり、表現力の差は平均利得式で規定できる性質を反映した言語である。ここで、平均利得式で記述可能な性質は無限長語全体の文字パターンの漸近的出現頻度に関する性質であり、局所的な文字パターンの出現については全く制限しない。よって、平均利得式のブール結合では、プレフィックスに依存しない性質しか記述することはできない。これは定義 6.2 から明らかである。一方、 ω -正規言語や LTL で記述可能な性質は局所的な文字パターンの繰り返しであり、無限長語全体の文字パターンの出現頻度についてはいっさい言及できない。従って、式 (6.17) のような恒真含意関係から $L(MP(1_\varphi) > 0) \subseteq L(\Box \Diamond \varphi)$ などの包含関係が成り立つものの、平均利得式が LTL と ω -正規言語の差を埋めることはない。

そのため、以下の命題が成り立つ。

定理 6.8 言語クラスの包含関係に関して、以下が成り立つ.

1. $\mathcal{L}(\omega\text{-REG}) \not\subseteq \mathcal{L}(LTL^{\text{mp}})$.
2. $\mathcal{L}(LTL^{\text{mp}}) \not\subseteq \mathcal{L}(\omega\text{-KC})$.

ただし、 $\omega\text{-KC}$ は、任意の有限長言語クラスの ω -Kleene 閉包とする.

証明. (1) : LTL^{mp} 文では記述できない ω -正規言語が存在することを示す. ここで、アルファベット $\Sigma = 2^A$, ある文字集合 $A \subset \Sigma$, $B = \Sigma \setminus A$ とする. そして、 LTL では表現できない ω -正規言語 $l = (\Sigma A)^*(\Sigma B)\Sigma^\omega$, すなわち、「ある偶数ステップに A に含まれない文字 $b \in B$ が出現する語」からなる言語を考える¹. ここで、 LTL^{mp} 文 φ でこの言語を記述できると仮定する (つまり、 $l = L(\varphi)$). 6.2 節で示した通り、この文 φ は MPNF 文 φ' で表現することができ、 $L(\varphi) = L(\varphi')$ である. また、定義 6.2 から、平均利得式の真偽は語のどんなプレフィックスにも依存しない. 一方で言語 l は、十分な有限長以後のサフィックスには依存しない語の集合である. 従って、語 σ に対する充足関係 $\sigma \models \varphi'$ の真偽は φ' の平均利得式部分には依存せずに決定される. よって、言語 l は LTL でも記述できなければならない. これは矛盾している. 以上により ω -正規言語 l は LTL^{mp} 文では一般に記述できない.

(2) : どんな有限長言語クラスの ω -Kleene 閉包でも記述できないような、 LTL^{mp} 文で定義される言語が存在することを示す. ここで、 LTL^{mp} 文 $\psi = \overline{\text{MP}}(\mathbf{1}_p) > 1/2 \wedge \text{MP}(\mathbf{1}_p) < 1/2$ を考える. このとき、この式を満たすすべての語について、 $\llbracket \mathbf{1}_p \rrbracket$ の平均利得トレースは収束しない. 一方、 ω -Kleene 閉包で定義されるどんな言語も $\bigcup_i l_i l_i'$ (ただし、 l_i, l_i' は正規言語) という形をしており、 $\sigma_1(\sigma_2)^\omega$ という形の語を必ず持っている. この語に対しては、 $\llbracket \mathbf{1}_p \rrbracket_{\sigma_1(\sigma_2)^\omega}$ の平均利得トレースは明らかに収束する. よって、 LTL^{mp} 文 ψ で定義される言語 $L(\psi)$ は ω -Kleene 閉包で定義されるどんな言語クラスにも含まれない. $\square$

¹ 「偶数ステップであること」はカウンタがなければ表現できないため、star-free あるいは counter-free である LTL ではこの言語 l は記述できない. この証明は本論文では省略する. ω -正規言語のフラグメントとも呼ばれる LTL は、

第7章 ω -正規オートマトンの平均 利得拡張

時間論理仕様の検証では、仕様が規定する言語を認識するオートマトン、つまり、仕様と等価なオートマトンを構成し、そのオートマトンを利用するアプローチ (automata-based approach) がしばしば用いられる。例えば、LTL 式 ψ から言語 $L(\psi)$ を受理言語とする Büchi オートマトン (BA) をアルゴリズム的に生成することができるため、その BA を利用することで LTL を用いた検証や合成を行うことができる [84, 85].

本研究では、 LTL^{mp} を用いた検証・合成に利用するためのオートマトン、すなわち、 LTL^{mp} 式で規定される言語を認識可能なオートマトンとして、 ω -正規受理条件 (Büchi/Rabin 受理条件) と多閾値平均利得 (MTMP) 受理条件を同時に満たす実行をもつ語を受理する重み付きオートマトン、**多閾値平均利得 ω -正規オートマトン** (MTMP Büchi/Rabin automaton, MTMPBA/MTMPRA, 単に MPBA/MPRA) を提案する [81, 97].

LTL^{mp} を認識できる MPBA/MPRA を構成する基本的なアイデアは以下のようなものである。

- 6.2 節で示したように、 LTL^{mp} 文 φ は等価な MPNF 文 $\bigvee_i (\psi_i \wedge \bigwedge_j \theta_{ij})$ で表現できる。そこで、
 - φ をと等価なオートマトンを、各選言肢 $\psi_i \wedge \bigwedge_j \theta_{ij}$ と等価なオートマトンの和合成として構成する。
 - 各選言肢 $\psi_i \wedge \bigwedge_j \theta_{ij}$ と等価なオートマトンでは、
 - * LTL 部分 ψ_i と等価な ω -正規オートマトンを構成し、
 - * 各平均利得式 $\theta_{ij} = \text{MP}(t_{ij}) \sim_{ij} 0$ の利得項 t_{ij} の組と対応するように、遷移に対する多次元重み付け関数 W を定義し、
 - * 平均利得式部分 $\bigwedge_j \theta_{ij}$ と対応するように、MTMP 受理条件を与える。

[6] で定義されている MTMP 受理条件は、それぞれの次元の平均利得に関する上極限/下極限に関する閾値条件を原子命題とする Boolean 結合で与えられるものであり、その原子命題と平均利得式はほとんどそのまま対応させることができる。そのため、LTL より表現力の高い古典的な ω -正規オートマトンに、MTMP 受理条件を追加することで、LTL^{mp} 文を認識できるような拡張 ω -正規オートマトン、MTMP ω -正規オートマトンを構成できる。

本章では、MTMP ω -正規オートマトンの定義を与え、その性質及び空判定手続きについて論じる。なお、LTL^{mp} から MTMPBA/MTMPRA への具体的な変換法については、8.1 節で述べる。

本章の構成

まず、7.1 節でこの MTMP ω -正規オートマトンの定義とその性質を述べる。次に、7.2 節で MTMP ω -正規オートマトンの認識する言語の閉包性について述べる。そして、7.3 節で MTMP ω -正規オートマトンの空判定アルゴリズムについて述べる、最後に、7.4 節で、LTL^{mp} と MTMP ω -正規オートマトンの表現力の関係について考察する。

7.1 多閾値平均利得 (MTMP) ω -正規オートマトン

MTMP ω -正規オートマトンは形式的には以下のように定義される。

定義 7.1 (MTMP ω -正規オートマトン [81]) MTMP ω -オートマトン $\mathcal{A} = \langle Q, \Sigma, \hookrightarrow, q_{init}, Acc \rangle$ は受理条件 Acc が Büchi 条件や Rabin 条件などの正規受理条件 F と多閾値平均利得受理条件 $\langle W, G \rangle$ の対で与えられるオートマトンである。 ■

MTMP ω -正規オートマトンの実行や実行上の語、パス、トレース、到達可能性、強連結成分などのオートマトン関連の各種概念については ω -オートマトンと同様に定義される。

そして、MTMP ω -正規オートマトン $\mathcal{A} = \langle Q, \Sigma, \hookrightarrow, q_{init}, \langle F, \langle W, G \rangle \rangle \rangle$ は、 ω -正規受理条件 Büchi _{F} (r) あるいは Rabin _{F} (r) と多閾値平均利得受理条件 MP _{$\langle W, G \rangle$} (r) を同時に満たす無限長実行 $r \in \text{Runs}^{\mathcal{A}}$ をもつ語 $\sigma = \text{word}(r)$ を受理する。

ここで、オートマトンの呼称については 3.3 節と同様の表記法を用いる。例えば F が Rabin 条件である決定性多閾値平均利得 ω -オートマトンを DMTMPRA

と表記する. また, オートマトンの認識する言語のクラスについても同様に $\mathcal{L}(\cdot)$ で表す. さらに, 集積点条件 G に関しても, 式 (3.67) で定義される, 実行の平均利得トレースの下極限が閾値条件 ~ 0 を満たすべき重み次元の集合 $H^\sim \subseteq \{1, \dots, d\}$ の四つ組 $\langle H^>, H^\geq, H^<, H^\leq \rangle$ の集合として与える略記を用いる.

注意 7.2 (LTL^{mp} 平均利得式と MTMP 受理条件の関係) LTL^{mp} 式で規定される言語を認識可能なオートマトンとして MTMPBA/MTMPRA を用いるが, LTL^{mp} 式で規定される性質のうち LTL 部分については Büchi/Rabin 受理条件によりその特徴を捉えることができる.

一方, LTL^{mp} 平均利得式で規定される性質は MTMP 受理条件でその特徴を捉える. 直観的には, W は,

- それぞれの m 番目次元成分 $W[m]$ と
- LTL^{mp} 利得項 t_m

を対応させることを意図した重み関数である. それにより, 任意の実行 r に対して, 平均利得トレース $\text{mp}_{W[m]}(r)$ と t_m の解釈を一致させる. G は,

- m が $H^\sim$ の要素であることと
- LTL^{mp} 平均利得リテラル $\text{MP}(t_m) \sim 0$

及び,

- 四つ組 $\langle H^>, H^\geq, H^<, H^\leq \rangle$ (の集合) と
- 平均利得リテラルの連言 (の選言)

を対応させることを意図した重みインデックス集合の四つ組の集合である.

なお, LTL^{mp} から MTMPBA/MTMPRA への具体的な変換法については, 8.1 節で述べる. ■

例 7.3 MTMPBA $\mathcal{A} = \langle \{q_0, q_1, q_2\}, 2^{\{p_0, p_1\}}, \{\delta_1, \dots, \delta_6\}, q_0, \langle \{q_0\}, \langle W, G \rangle \rangle \rangle$ を考える (図 7.1). ここで, 命題の集合 $\emptyset$, $\{p_0\}$, $\{p_1\}$ をそれぞれ A , B , C と略記し, 遷移関係は以下のように与えられるものとする.

$$\begin{aligned} \delta_1 &= \langle q_0, A, q_1 \rangle, & \delta_2 &= \langle q_1, A, q_1 \rangle, \\ \delta_3 &= \langle q_1, B, q_0 \rangle, & \delta_4 &= \langle q_0, A, q_2 \rangle, \\ \delta_5 &= \langle q_2, A, q_2 \rangle, & \delta_6 &= \langle q_2, C, q_0 \rangle. \end{aligned} \tag{7.1}$$

また、 W と G は以下のような二次元重み関数と多閾値平均利得受理条件であるとする.

$$W[1](\delta) = \begin{cases} -1 & \text{if } \delta \in \{\delta_1, \delta_2, \delta_3\}, \\ 1 & \text{otherwise,} \end{cases} \quad (7.2)$$

$$W[2](\delta) = \begin{cases} -1 & \text{if } \delta = \delta_3, \\ 1 & \text{if } \delta = \delta_6, \\ 0 & \text{otherwise,} \end{cases} \quad (7.3)$$

$$G = \{\langle \emptyset, H^{\geq}, \emptyset, H^{\leq} \rangle\}, \quad (7.4)$$

$$H^{\geq} = \{2\}, \quad (7.5)$$

$$H^{\leq} = \{1\}. \quad (7.6)$$

Büchi 受理条件 $\text{Büchi}_{\{q_0\}}$ より, このオートマトンの受理言語は明らかに ω -正規言語 $(A^+(B+C))^\omega$ の部分集合である. また, 重み関数 $W[1]$ は, Büchi 受理条件を考慮すれば「 A until B を満たす時刻では利得 -1 , そうでない時刻では利得 1 を与える」ものと解釈できる. 一方, 重み関数 $W[2]$ は, 「 B に利得 -1 , C に利得 1 を与える」ことを表している. そして, $H^{\geq} = \{2\}$ であることから, このオートマトンに受理される語では「実行時間に対する“ C と B の出現数差”の割合の下極限が 0 以上」である. また, $H^{\geq} = \{1\}$ であることから, このオートマトンに受理される語では「実行時間に対する“ A until B を満たしている時間満たしていない時間の差”の割合の下極限が 0 以下」である.

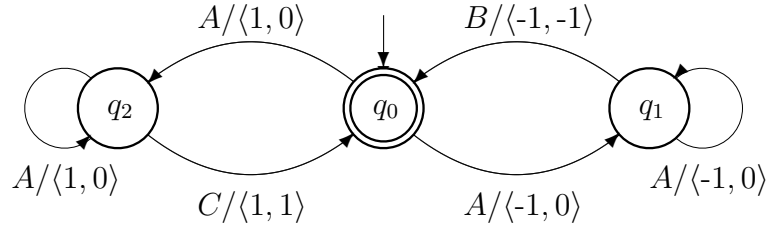
ここで, 以下のような二つの実行を考える.

$$\begin{aligned} r_1 &= (\delta_1 \delta_3)^1 \delta_4 \delta_6 (\delta_1 \delta_3)^2 \delta_4 \delta_6 \dots \\ r_2 &= (\delta_1 \delta_3) (\delta_4 \delta_5^{2^2-2} \delta_6) (\delta_1 \delta_2^{2^3-2} \delta_3) (\delta_4 \delta_5^{2^4-2} \delta_6) \dots \end{aligned}$$

このとき, どちらの実行も Büchi 受理条件 $\{q_0\}$ は満たしている. また, 実行 r_1 上の利得トレース $\text{payoff}_W(r_1)$ は,

$$\text{payoff}_W(r_1) = \left(\begin{pmatrix} -1 \\ 0 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \end{pmatrix} \right)^1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \left(\begin{pmatrix} -1 \\ 0 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \end{pmatrix} \right)^2 \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \dots$$

この r_1 上の平均利得トレース $\text{mp}_W(r_1)$ は明らかに $(W(\delta_1) + W(\delta_3))/2$, すなわち, $\langle -1, -1/2 \rangle$ へ収束する. よって, 実行 r_1 は, 2 番目次元の平均利得トレースの下極限 (と上極限) が閾値条件 ≥ 0 を満たさないため, 受理されな

図 7.1: 例 7.3 の MTMPBA $\mathcal{A}$.

い. 一方で, 実行 r_2 上の利得トレース $\text{payoff}_W(r_2)$ は,

$$\text{payoff}_W(r_2) = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}^{2^2-1} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} -1 \\ 0 \end{pmatrix}^{2^3-1} \begin{pmatrix} -1 \\ -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}^{2^4-1} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \cdots$$

である. そしてこのとき, r_1 は, 1 番目次元の平均利得トレースの下極限が $-1/3$ であるため閾値条件 ≤ 0 を満たし, 1 番目次元の平均利得トレースの下極限 (と上極限) が 0 であるため閾値条件 ≥ 0 を満たすため, 受理される. ■

7.2 閉包性

本節では, MTMP ω -正規オートマトンが認識する言語のクラスの閉包性について述べる.

和/積に関する閉包性を示すには, 二つ MTMP ω -正規オートマトンからこれらの認識する言語の和集合/積集合を認識する MTMP ω -正規オートマトンを合成できればよい. これは, ω -正規オートマトンの和合成/積合成法と同様の構成法を適用すればよい. ただし, MTMP ω -正規オートマトンの受理条件は「 F かつ G 」で与えられるため, 和合成については工夫が必要である. 以下では, MTMPBA の和合成/積合成法の直観的アイディアを示す.

和: BA の和合成と同様の構成法で元の二つのオートマトンを非決定的に模倣するオートマトンを合成し, また, 模倣している方のオートマトンと同様の利得を与えることで利得トレースも模倣するオートマトンを構成する. なお, ここでは元の二つのオートマトンの重み関数の次数が等しいと仮定している. もし次数が異なっても, 受理言語が同じ次数が大きなオートマトンは容易に構成できる (式 (3.65)) ため, 一般性を失わない.

利得トレースを模倣するためには d 次元の重み関数があればよいが、どちらのオートマトンを模倣しているかを認識するためにはさらに 1 次元が必要である。この $d+1$ 番目の次元がない場合、一方の Büchi 受理条件 F かつ他方の平均利得受理条件 G' を満たす語を受理してしまう可能性がある。よって、和合成 MTMPBA の $d+1$ 次元重み関数は、 $1-d$ 番目の次元については模倣している方のオートマトンと同じ利得を、 $(d+1)$ 番目の次元については一方を模倣するときには常に 1、もう一方を模倣するときには常に 0 の利得を与えればよい。そして、和合成 MTMPBA の平均利得受理条件でどちらのオートマトンを模倣しているか（つまり $(d+1)$ 番目の次元の平均利得）を考慮にいたした上で設定すればよい。これにより、二つの MTMPBA の認識する言語の和集合を認識する $(d+1)$ 次元 MTMPBA を得ることができる。

積: BA の積と同様の構成法で同期合成オートマトンを合成し、その上で二つのオートマトンの利得トレースをどちらも模倣するように重み関数を定義し、また、その重みの上で元の平均利得受理条件の連言を意味する平均利得受理条件を与えればよい。二つの MTMPBA の認識する言語の和集合を認識する $(d+1)$ 次元 MTMPBA を得ることができる。これにより、二つの MTMPBA の認識する言語の積集合を認識する MTMPBA を得ることができる。

具体的な構成法は、補題 7.5 と補題 7.6 に示す。これにより、以下の定理が成り立つ。

定理 7.4 (MTMP ω -正規オートマトンの和と積に関する閉包性 [97]) MTMP ω -正規オートマトンが認識する言語のクラスは和集合と積集合に関して閉じている。 ■

証明. 本論文では MTMPBA の場合のみを示す。補題 7.5 と補題 7.6 より、二つの MTMPBA の認識する言語の和集合、積集合を認識する MTMPBA を合成できる。以上により示された。 □

補題 7.5 (MTMPBA の和オートマトンの構成法) Q と Q' を互いに素である状態の集合、 $\mathcal{A} = \langle Q, \Sigma, \hookrightarrow, q_{init}, \langle F, \langle W, G \rangle \rangle \rangle$ と $\mathcal{A}' = \langle Q', \Sigma, \hookrightarrow', q'_{init}, \langle F', \langle W', G' \rangle \rangle \rangle$ を d 次元重み付き MTMPBA とする。

このとき、以下のような $(d+1)$ 次元重み付き MTMPBA $\mathcal{A}'' = \langle Q \cup Q' \cup \{q''_{init}\}, \Sigma, \hookrightarrow \cup \hookrightarrow' \cup \hookrightarrow'', q''_{init}, \langle F \cup F', \langle W'', G'' \rangle \rangle \rangle$ は $L(\mathcal{A}) \cup L(\mathcal{A}')$ を認識す

る。ただし、

$$\begin{aligned}
\hookrightarrow'' &= \{ \langle q''_{init}, a, q \rangle \mid \langle q_{init}, a, q \rangle \in \hookrightarrow \} \\
&\cup \{ \langle q''_{init}, a, q' \rangle \mid \langle q'_{init}, a, q' \rangle \in \hookrightarrow' \}, \\
W''(\langle q, a, q' \rangle) &= \begin{cases} \langle W[1](\langle q, a, q' \rangle), \dots, W[d](\langle q, a, q' \rangle), 1 \rangle & \text{if } q' \in Q, \\ \langle W'[1](\langle q, a, q' \rangle), \dots, W'[d](\langle q, a, q' \rangle), 0 \rangle & \text{otherwise,} \end{cases} \\
G''(\cdot) &= (\min \pi_{d+1}(\cdot) \geq 1 \wedge G(\cdot)) \vee (\max \pi_{d+1}(\cdot) \leq 0 \wedge G'(\cdot)). \quad \blacksquare
\end{aligned}$$

証明. $\mathcal{A}''$ は、 $\mathcal{A}$ と $\mathcal{A}'$ の遷移関係 $\hookrightarrow$ と $\hookrightarrow'$ 、そして、両方の初期状態を模倣する新たな初期状態 q''_{init} からの遷移関係 $\hookrightarrow''$ をもつ。よって $\mathcal{A}''$ の実行は、 $\mathcal{A}$ と $\mathcal{A}'$ のどちらかの実行を模倣する。そして、 $\mathcal{A}$ を模倣するときには $d+1$ 次元目の利得トレースは常に 1、 $\mathcal{A}'$ を模倣するときには $d+1$ 次元目の利得トレースは常に 0 である。

そのため、 $\min \pi_{d+1}(\cdot) \geq 1$ を満たす $\mathcal{A}''$ の実行上には Q' に含まれる状態、つまり、 F' に含まれる状態は出現しない。よって、ある語に対して、 $\mathcal{A}''$ の実行 $r'' = \langle q''_{init}, a_0, q_1 \rangle \langle q_1, a_1, q_2 \rangle \dots$ が Büchi 受理条件 $\text{Büchi}_{F \cup F'}(r'')$ 、MTMP 受理条件 $\text{MP}_{W'', G''}(r'')$ 及び $\min \pi_{d+1}(\cdot) \geq 1$ を満たすならば、かつそのときに限り、 $\mathcal{A}$ には Büchi 受理条件 $\text{Büchi}_F(r)$ かつ MTMP 受理条件 $\text{MP}_{W, G}(r)$ を満たす実行 $r = \langle q_{init}, a_0, q_1 \rangle \langle q_1, a_1, q_2 \rangle \dots$ が存在する。同様の事実が $\mathcal{A}'$ についても成り立つ。

従って、 $\mathcal{A}''$ は $\mathcal{A}$ または $\mathcal{A}'$ に受理される語の集合を認識する。 $\square$

補題 7.6 (MTMPBA の積オートマトンの構成法) Q と Q' を互いに素である状態の集合、 $\mathcal{A} = \langle Q, \Sigma, \hookrightarrow, q_{init}, \langle F, \langle W, G \rangle \rangle \rangle$ と $\mathcal{A}' = \langle Q', \Sigma, \hookrightarrow', q'_{init}, \langle F', \langle W', G' \rangle \rangle \rangle$ をそれぞれ d 次元重み付き MTMPBA と d' 次元重み付き MTMPBA とする。

このとき、以下のような $(d + d')$ 次元重み付き MTMPBA $\mathcal{A}'' = \langle Q \times Q' \times \{0, 1\}, \Sigma, \hookrightarrow'', \langle q_{init}, q'_{init}, 0 \rangle, \langle F'', \langle W'', G'' \rangle \rangle \rangle$ は $L(\mathcal{A}) \cap L(\mathcal{A}')$ を認識する。ただし、

$$\begin{aligned}
\langle \langle q_1, q'_1, i \rangle, a, \langle q_2, q'_2, j \rangle \rangle \in \hookrightarrow'' &\Leftrightarrow (\langle q_1, a, q_2 \rangle \in \hookrightarrow \wedge \\
&\delta' \langle q'_1, a, q'_2 \rangle \in \hookrightarrow' \wedge \\
&(i = 0 \Rightarrow (q_1 \in F \Leftrightarrow j = 1)) \wedge \\
&(i = 1 \Rightarrow (q'_1 \in F' \Leftrightarrow j = 0))), \\
W''(\delta'') &= \langle W[1](\delta), \dots, W[d](\delta), W'[1](\delta'), \dots, W'[d'](\delta') \rangle, \\
F'' &= \{ \langle q, q', 0 \rangle \mid q \in F \}, \\
G''(\cdot) &= G(\cdot) \wedge G'[\pi_1, \dots, \pi_{d'} / \pi_{d+1}, \dots, \pi_{d+d'}](\cdot).
\end{aligned}$$

証明. 遷移関係 $\hookrightarrow''$ の構成法から、任意の語 $\sigma = a_0 a_1 \dots$ に対して、 $\mathcal{A}''$ 上に実行 $r'' = \langle \langle q_{init}, q'_{init}, i_0 \rangle, a_0, \langle a_1, a'_1, i_1 \rangle \rangle \langle \langle q_1, q'_1, i_1 \rangle, a_1, \langle a_2, a'_2, i_2 \rangle \rangle \dots$ が存在するならば、 $\mathcal{A}$ 上には実行 $r = \langle q_{init}, a_0, a_1 \rangle \langle q_1, a_1, a_2 \rangle \dots$ 、 $\mathcal{A}'$ 上には実行 $r' = \langle q'_{init}, a_0, a'_1 \rangle \langle q'_1, a_1, a'_2 \rangle \dots$ が存在している。

$\mathcal{A}''$ の状態 $\langle q, q', i \rangle$ のラベル i は、BA の積合成法と同様に、 F や F' に含まれる状態を訪れた履歴を表すものである。 $\hookrightarrow''$ では、(a) 0 ラベル付きの状態 $\langle q, q', 0 \rangle$ からは、 $q \in F$ のとき 1 ラベル付きの状態へ、そうでないとき 0 ラベル付きの状態への遷移し、(b) 1 ラベル付きの状態 $\langle q, q', 1 \rangle$ からは、 $q' \in F'$ のとき 0 ラベル付きの状態へ、そうでないとき 1 ラベル付きの状態への遷移する。

よって、 r が Büchi 受理条件 $\text{Büchi}_F(r)$ を満たし、かつ r' が Büchi 受理条件 $\text{Büchi}_{F'}(r')$ を満たすならば、かつそのときに限り、 r'' は Büchi 受理条件 $\text{Büchi}_{F''}(r'')$ を満たす。そして、 $\mathcal{A}''$ の重み関数 W'' は、 $1-d$ 次元で $\mathcal{A}$ の重み関数を、 $(d+1)-(d+d')$ 次元で $\mathcal{A}'$ の重み関数をそれぞれ保存しており、また、多閾値平均利得受理条件 G'' はそれに対応した閾値条件に言及するように次元を再設定している。よって、 r と r' が MTMP 受理条件 $\text{MP}_{W,G}(r)$ と $\text{MP}_{W',G'}(r')$ をそれぞれ満たすならば、かつそのときに限り、 r'' は MTMP 受理条件 $\text{MP}_{W'',G''}(r'')$ を満たす。

従って、 $\mathcal{A}''$ は $\mathcal{A}$ と $\mathcal{A}'$ のどちらにも受理される語の集合を認識する。 $\square$

また、決定性の MTMP ω -オートマトンについては、以下の定理が成り立つ。

定理 7.7 (決定性 MTMP ω -正規オートマトンの和と積に関する閉包性)

DMTMPBA/DMTMPRA が認識する言語は積集合に関して閉じている。 $\blacksquare$

証明. 補題 7.6 に示されている積合成法は、元のオートマトンが決定性であれば積合成オートマトンも決定性のものを得ることができる。以上により示された。 $\square$

7.3 空判定

MPBA の空判定手続きは、Boker らによって、 LTL^{mp} のフラグメントである LTL^{lim} のモデル検査手続きの一部として暗に提案されている [19]。つまり、 LTL^{lim} のモデル検査を実質的には MTMPBA の空判定に帰着している。そのアルゴリズムは、MTMP 受理条件のみを受理条件としてもつオートマトンの空判定手続き [6] を基に、Büchi 受理条件も考慮するように改良したものであ

る。しかしながら、そのアルゴリズムはオートマトンのサイズに対して指数時間の計算量を要する非効率的なものであった。

本研究では Boker らの手法とは異なり、MPBA の空判定問題を**線形制約充足問題** (linear constraint satisfaction problem, LCSP) に帰着して解く効率的なアルゴリズムを提案する [81]。

本節では、まず 7.3.1 項で MPBA が Büchi 受理条件を満たす実行をもつ必要充分条件とそれに関連した補題を示す。次に、7.3.2 項で MPBA が平均利得受理条件を満たす実行をもつ必要充分条件について述べ、7.3.3 項でその必要充分条件を線形制約で表現するアイデアについて述べる。最後に 7.3.4 項で空判定手続きとその正当性の証明を与える。

7.3.1 Büchi 受理実行存在の必要充分条件

d 次元 MTMPBA $\mathcal{A} = \langle Q, \Sigma, \hookrightarrow, q_{init}, \langle F, \langle W, G \rangle \rangle \rangle$ の Büchi 受理条件を満たす実行は、明らかに BA $\langle Q, \Sigma, \hookrightarrow, q_{init}, F \rangle$ の受理実行である。この BA の受理実行が存在するための必要充分条件は、初期状態 q_{init} から到達可能かつ非自明な極大強連結成分 (MSCC) $\langle Q', \hookrightarrow' \rangle \in \text{MSCC}^{\mathcal{A}}$ が存在し、以下の条件を満たすことである。

- (i). $F \cap Q' \neq \emptyset$, すなわち、受理状態集合 F と共通部分をもつ。

この条件が成り立つならば、かつそのときに限り、 $\langle Q', \hookrightarrow' \rangle$ 上には、 q_{init} から受理状態 $q \in F \cap Q'$ への実行断片 r_0 と q から q への閉路 $r_1 \in (\hookrightarrow')^+$ が存在する。よって、それらから構成される実行 $r_0(r_1)^\omega$ は明らかに Büchi 受理条件 $\text{Büchi}_F(r_0(r_1)^\omega)$ を満たす。

また、以下の補題が成り立つ。

補題 7.8 MTMPBA $\mathcal{A} = \langle Q, \Sigma, \hookrightarrow, q_{init}, \langle F, \langle W, G \rangle \rangle \rangle$ の初期状態 q_{init} から到達可能かつ非自明な MSCC $\langle Q', \hookrightarrow' \rangle \in \text{MSCC}^{\mathcal{A}}$ に関して以下の条件が成り立つ。

$$\begin{aligned} & \exists r \in \text{Runs}^{\mathcal{A}}. (\text{Inf}(r) \cap Q' \neq \emptyset \wedge F \cap Q' \neq \emptyset \wedge \text{MP}_{W,G}(r)) \\ & \Leftrightarrow \exists r \in \text{Runs}^{\mathcal{A}}. (\text{Inf}(r) \cap Q' \cap F \neq \emptyset \wedge \text{MP}_{W,G}(r)). \end{aligned} \quad (7.7)$$

■

証明. 左辺式 $\Leftarrow$ 右辺式は自明であるため、以下では左辺式 $\Rightarrow$ 右辺式を示す。

左辺式を成立させる実行 $r \in \text{Runs}^{\mathcal{A}}$ を考える。この r は明らかに $r_0 \delta_0 \delta_1 \dots \in (\hookrightarrow)^*(\hookrightarrow')^\omega$ という形をもつ実行である。また、 Q' は F と共通部分をもつ MSCC

であるため、任意の $q \in Q'$ について、 q から q へ受理状態 $q_F \in F$ を経由して到達する長さ $2 \cdot |Q'|$ 以下の閉路 $r_q^F \in (\hookrightarrow')^+$ が存在する。そのため、 $q_i = \text{post}(\delta_i)$ ($i \in \mathbb{N}$) としたとき、 $\delta_0 \delta_1 \dots$ の 2^i 番目の遷移 δ_{2^i} の後に閉路 $r_{q_{2^i}}^F$ を挿入した実行 r' を構成できる。

$$r' = r_0 \delta_0 \delta_1 r_{q_1}^F \delta_2 r_{q_2}^F \delta_3 \dots \delta_{2^i} r_{q_{2^i}}^F \delta_{2^i+1} \dots \quad (7.8)$$

この r' は明らかに $\text{Inf}(r') \cap Q' \cap F \neq \emptyset$ を満たす。ここで、 r' 上の閉路族 $r_{q_{2^i}}^F$ の出現頻度は 0 に収束するため、任意の j について以下の等式が成り立つ。

$$\liminf_{n \rightarrow \infty} \text{mp}_{W[j]}(r')[n] = \liminf_{n \rightarrow \infty} \text{mp}_{W[j]}(r)[n]. \quad (7.9)$$

以上により、 $\text{MP}_{W,G}(r')$ 及び右辺式が成り立つ。 $\square$

7.3.2 平均利得受理実行存在の必要充分条件

補題 7.8 より、 d 次 MTMPBA $\mathcal{A} = \langle Q, \Sigma, \hookrightarrow, q_{\text{init}}, \langle F, \langle W, G \rangle \rangle \rangle$ について、「受理状態集合 F と共通部分をもつ MSCC $\langle Q', \hookrightarrow' \rangle$ を無限にしばしば訪れ、かつ、平均利得受理条件 $\text{MP}_{W,G}(r)$ を満たす実行」が存在するならば、かつそのときに限り、Büchi 受理条件と MTMP 受理条件を同時に満たす実行が存在する。そこで本節では、初期状態 q_{init} から到達可能かつ非自明な MSCC $\langle Q', \hookrightarrow' \rangle$ に含まれる状態を無限にしばしば訪れ、かつ、四つ組 $g = \langle H^>, H^{\geq}, H^<, H^{\leq} \rangle$ のみの単集合に関する MTMP 受理条件 $\text{MP}_{W,\{g\}}(r)$ を満たす実行 r が存在する必要充分条件を考える。

ここでまず、 $\langle Q', \hookrightarrow' \rangle$ 中の状態を無限にしばしば訪れる実行 r の d 次元平均利得トレース $\text{mp}_W(r) \in (\mathbb{R}^d)^\omega$ が無限にしばしばその近傍を通り得るベクトル (標準 Euclidean 空間 $\mathbb{R}^d$ 上の点) について考える。 $q \in \text{Inf}(r) \cap Q'$ を満たす実行 r は、初期状態 q_{init} から q への実行断片 r_0 と、 q から q への無限個の閉路 $r_1, r_2, \dots$ の連結 $r_0 r_1 r_2 \dots$ であり、それぞれの閉路 r_l ($l \in \mathbb{N}_{>0}$) が単純閉路の有限回の連結/挿入で構成されている。よって、 r の平均利得トレース $\text{mp}_W(r)$ が無限にしばしばその任意の近傍を通るベクトルは、 r のそれぞれの時点までの単純閉路総出現数に対する各単純閉路の出現比を表すベクトルが無限にしばしばその任意の近傍に近づく比で特徴付けることができる。

ここで、 $\langle Q', \hookrightarrow' \rangle$ 上の単純閉路の個数を n ($= \mathcal{O}(2^{|\hookrightarrow'|})$) とし、単純閉路集合 $C_{\text{simp}}^A(\langle Q', \hookrightarrow' \rangle) = \{r_1, \dots, r_n\}$ 中の閉路の出現比を表す零ベクトルでない非負ベクトル $\mathbf{y} \in \mathbb{R}_{\geq 0}^n$ を考える。ただし、 $\mathbf{y}$ は単純閉路 $r_m \in C_{\text{simp}}^A(\langle Q', \hookrightarrow' \rangle)$ でインデックス付けされているものとし、 $\mathbf{y}[r_m]$ で r_m に対応する値を表す。この

とき, 任意の $\mathbf{y}$ に対して, 平均利得トレース $\text{mp}_W(r)$ が以下のように定義されるベクトル $\mathbf{c}_y^W \in \mathbb{R}^d$ に収束するような実行 r が存在する.

$$\mathbf{c}_y^W = \frac{\sum_{m=1}^n \left(\mathbf{y}[r_m] \cdot \sum_{i=0}^{|r_m|-1} w(r_m[i]) \right)}{\sum_{m=1}^n (\mathbf{y}[r_m] \cdot |r_m|)}. \quad (7.10)$$

直観的には, $\mathbf{c}_y^W$ は単純閉路 $r_1, \dots, r_n$ のみが比 $\mathbf{y}[r_1], \dots, \mathbf{y}[r_n]$ で現れる際の利得の平均値を表している.

$\langle Q', \hookrightarrow' \rangle$ は SCC であるため, ある実行断片 $r'_1, \dots, r'_n$ が存在し, $r_1 r'_1 \dots r_n r'_n \in (\hookrightarrow')^+$ という大きな閉路が存在する. ここで, $\mathbf{y}$ の要素がすべて自然数である場合を考える. このとき, 閉路族

$$r_{\mathbf{y}}(i) = (r_1)^{\mathbf{y}[r_1] \cdot i} r'_1 \dots (r_n)^{\mathbf{y}[r_n] \cdot i} r'_n$$

($i \in \mathbb{N}_{>0}$) から構成される実行

$$r_{\mathbf{y}} = r_0(r_{\mathbf{y}}(1))(r_{\mathbf{y}}(2)) \dots$$

が存在する. 閉路 $r_{\mathbf{y}}(1) \dots r_{\mathbf{y}}(i)$ の長さ $|r_{\mathbf{y}}(1) \dots r_{\mathbf{y}}(i)|$ は $\Theta(i^2)$ であり, この閉路上の $\mathbf{y}[r_m] > 0$ であるそれぞれの単純閉路 r_m の出現数も $\Theta(i^2)$ である. 一方, この閉路上のそれぞれのパス r'_m の出現数は i である. 従って, 実行 $r_{\mathbf{y}}$ の平均利得トレース $\text{mp}_W(r_{\mathbf{y}})$ は $\mathbf{c}_y^W$ に収束する. $\mathbf{y}$ の要素が自然数でない場合でも, 単純閉路出現比が $\mathbf{y}$ に収束するような実行が構成でき, その平均利得トレースはやはり $\mathbf{c}_y^W$ に収束する. つまり, 任意の実数比 $\mathbf{y}$ について, 「 S 中の状態を無限にしばしば訪れ, かつ, その平均利得トレースが $\mathbf{c}_y^W$ の任意の近傍を無限にしばしば通るような実行」が存在する.

以上の事実から, 以下の補題が成り立つ.

補題 7.9 $g = \langle H^>, H^{\geq}, H^<, H^{\leq} \rangle$ を重みインデックス集合の四つ組とする. このとき, 初期状態 q_{init} から到達可能かつ非自明な MSCC $\langle Q', \hookrightarrow' \rangle$ に含まれる状態を無限にしばしば訪れる実行集合 $\{r \in \text{Runs}^A \mid \text{Inf}(r) \cap Q' \neq \emptyset\}$ の中に, 平均利得受理条件 $\text{MP}_{W, \{g\}}(r)$ を満たす実行 r が存在する必要充分条件は, 条件 (ii-1)-(ii-3) をすべて満たすことである.

(ii-1). $H^< = H^{\leq} = \emptyset$ ならば, $\langle Q', \hookrightarrow' \rangle$ 上の単純閉路集合 $C_{\text{simp}}^A(\langle Q', \hookrightarrow' \rangle)$ の出現比を表す非零かつ非負のベクトル $\mathbf{y} \in \mathbb{R}_{\geq 0}^{|C_{\text{simp}}^A(\langle Q', \hookrightarrow' \rangle)|}$ が存在し, 以下

の条件を満たす.

$$\forall \triangleright \in \{>, \geq\}. h \in H^\triangleright. \mathbf{c}_y^W[h] \triangleright 0. \quad (7.11)$$

(ii-2). $H^< \neq \emptyset$ ならば, それぞれの $h' \in H^<$ について, $\langle Q', \hookrightarrow' \rangle$ 上の単純閉路集合 $C_{\text{simp}}^A(\langle Q', \hookrightarrow' \rangle)$ の出現比を表す非零かつ非負のベクトル $\mathbf{y}_{h'} \in \mathbb{R}_{\geq 0}^{|C_{\text{simp}}^A(\langle Q', \hookrightarrow' \rangle)|}$ が存在し, 以下の条件を満たす.

$$\forall \triangleright \in \{>, \geq\}. h \in H^\triangleright. \mathbf{c}_{y_{h'}}^W[h] \triangleright 0 \wedge \mathbf{c}_{y_{h'}}^W[h'] < 0. \quad (7.12)$$

(ii-3). $H^\leq \neq \emptyset$ ならば, それぞれの $h'' \in H^\leq$ について, $\langle Q', \hookrightarrow' \rangle$ 上の単純閉路集合 $C_{\text{simp}}^A(\langle Q', \hookrightarrow' \rangle)$ の出現比を表す非零かつ非負のベクトル $\mathbf{y}_{h''} \in \mathbb{R}_{\geq 0}^{|C_{\text{simp}}^A(\langle Q', \hookrightarrow' \rangle)|}$ が存在し, 以下の条件を満たす.

$$\forall \triangleright \in \{>, \geq\}. h \in H^\triangleright. \mathbf{c}_{y_{h''}}^W[h] \triangleright 0 \wedge \mathbf{c}_{y_{h''}}^W[h''] \leq 0. \quad (7.13)$$

■

証明. まず, $H^< = H^\leq = \emptyset$ を仮定する. このとき, 明らかに, 条件 (ii-1) を成立させるベクトル $\mathbf{y}$ が存在するならば, かつそのときに限り, 「 $\langle Q', \hookrightarrow' \rangle$ に含まれる状態を無限にしばしば訪れ, かつ, 単純閉路出現比が $\mathbf{y}$ に収束する実行 r 」が存在する. この r の平均利得トレース $\text{mp}_W(r)$ は $\mathbf{c}_y^W$ に収束するため, 条件 (ii-1) を成立させるベクトル $\mathbf{y}$ が存在するならば, かつそのときに限り, 平均利得受理条件 $\text{MP}_{W, \{g\}}(r)$ を満たす実行 r が存在する.

次に, $H^< \neq \emptyset$ または $H^\leq \neq \emptyset$ を仮定する. ここで, 条件 (ii-2) と条件 (ii-3) が成り立つならば, かつそのときに限り, ベクトル $\mathbf{y}_{h'_1}, \dots, \mathbf{y}_{h'_{|H^<|}}, \mathbf{y}_{h''_1}, \dots, \mathbf{y}_{h''_{|H^\leq|}}$ が存在し, 以下の条件が成り立っている.

$$\forall \mathbf{y}' \in \{\mathbf{y}_{h'_1}, \dots, \mathbf{y}_{h'_{|H^<|}}, \mathbf{y}_{h''_1}, \dots, \mathbf{y}_{h''_{|H^\leq|}}\}. \triangleright \in \{>, \geq\}. h \in H^\triangleright. \mathbf{c}_{\mathbf{y}'}^W[h] \triangleright 0. \quad (7.14)$$

このとき, 任意の $h \in H^\triangleright$ について全稱的閾値条件 $\liminf_{n \rightarrow \infty} \text{mp}_{W[h]}(r')[n] \triangleright 0$ を満たし, かつ, 単純閉路出現比が各ベクトル $\mathbf{y}_{h'_1}, \dots, \mathbf{y}_{h'_{|H^<|}}, \mathbf{y}_{h''_1}, \dots, \mathbf{y}_{h''_{|H^\leq|}}$ の任意の近傍を無限にしばしば通るような実行 r' を構成できる. また, r' の構成及び, 式 (7.12) と式 (7.13) より, 任意の $h \in H^\triangleleft$ について存在的閾値条件 $\liminf_{n \rightarrow \infty} \text{mp}_{W[h]}(r')[n] \triangleleft 0$ も成り立っている. 従って, 条件 (ii-2) と条件 (ii-3) が成り立つならば, かつそのときに限り, $\text{MP}_{W, \{g\}}(r')$ を満たす実行 r' が存在する.

以上により示された.

□

7.3.3 線形制約を用いた平均利得受理実行存在の必要充分条件の表現

本節では、前節の補題 7.9 の条件 (ii-1)-(ii-3) を線形制約で表現するためのアイデアについて述べる。

本研究では、そのような線形制約を $|\hookrightarrow'|$ 次元非負変数ベクトル $\mathbf{x}$ 上で与える。ここで、ベクトル $\mathbf{x}$ は遷移 $\delta \in \hookrightarrow'$ でインデックスが付けられており、遷移 δ に対応する $\mathbf{x}$ の要素を $\mathbf{x}[\delta]$ と表記する。このベクトル $\mathbf{x}$ は $\hookrightarrow'$ 上の遷移出現数（出現比）を表現することを意図した変数である。つまり、 S 上の単純閉路集合 $C_{\text{simp}}^A((Q', \hookrightarrow')) = \{r_1, \dots, r_n\}$ に対して、単純閉路出現比 $\mathbf{y}$ は以下のような遷移出現比 $\mathbf{x}$ と対応する。

$$\mathbf{x}[\delta] = \sum_{m=1}^n (\mathbf{y}[r_m] \cdot \text{Occ}_\delta(r_m)). \quad (7.15)$$

ただし、 $\text{Occ}_\delta(r_l)$ は閉路 r_l 中出现する遷移 δ の数とする。

単純閉路出現比を表現するための線形制約

非負ベクトル $\mathbf{x}$ が、SCC S 上の単純閉路出現比 $\mathbf{y}$ を表現するために満たすべき線形制約について述べる。

遷移 δ の出現比（出現数） $\mathbf{x}[\delta]$ の満たすべき条件は以下のような線形制約として表現できる。

$$\mathbf{u}^T \mathbf{x} > 0, \quad (7.16)$$

$$\mathbf{A}_{\text{cyc}}^{A, \langle Q', \hookrightarrow' \rangle} \mathbf{x} = \mathbf{0}. \quad (7.17)$$

ただし、 $\mathbf{u}$ はすべての要素が 1 の $|\hookrightarrow'|$ 次元実係数ベクトル、 $\mathbf{0}$ はすべての要素が 0 の $|Q'|$ 次元実係数ベクトル、 $\mathbf{A}_{\text{cyc}}^{A, \langle Q', \hookrightarrow' \rangle}$ は以下のように定義される $|Q'|$ 行 $|\hookrightarrow'|$ 列実係数行列である。

$$\mathbf{A}_{\text{cyc}}^{A, S}[q, \delta] = \begin{cases} 1 & \text{if } \text{pre}(\delta) = q \neq \text{post}(\delta), \\ -1 & \text{if } \text{post}(\delta) = q \neq \text{pre}(\delta), \\ 0 & \text{otherwise.} \end{cases} \quad (7.18)$$

ここで、 $\mathbf{A}_{\text{cyc}}^{A, \langle Q', \hookrightarrow' \rangle}$ の行は状態 $q \in Q'$ 、列は遷移 $\delta \in \hookrightarrow'$ でインデックスが付けられており、 q 行 δ 列の要素を $\mathbf{A}_{\text{cyc}}^{A, \langle Q', \hookrightarrow' \rangle}[q, \delta]$ と表記する。このとき、式 (7.16)

は遷移出現総数が0でない, すなわち, $\mathbf{y}$ が零ベクトルでないことを表現している. また, 式 (7.17) の左辺ベクトル $\mathbf{A}_{\text{cyc}}^{A, \langle Q', \hookrightarrow' \rangle} \mathbf{x}$ の q 行目は, 「 q へ入る遷移の出現数の総和」と「 q から出る遷移の出現数の総和」の差を表している. 従って, 式 (7.16) と (7.17) を同時に満たす $\mathbf{x}$ は単純閉路出現比 $\mathbf{y}$ を表現している.

平均利得に関する制約

前節の補題 7.9 の条件 (ii-1)-(ii-3) は, S 上の単純閉路出現比が $\mathbf{y}$ に収束する実行の平均利得収束値 $\mathbf{c}_y^W$ の要素に関する閾値条件で与えられている. そこでまず, $\mathbf{x}$ 上での $\mathbf{c}_y^W$ の表現について考える. ここで, 任意の単純閉路 $r_m \in C_{\text{simp}}^A(\langle Q', \hookrightarrow' \rangle)$ に関して以下の性質が成り立つ.

$$|r_m| = \sum_{\delta \in \Delta(S)} \text{Occ}_\delta(r_m), \quad (7.19)$$

$$\sum_{i=0}^{|r_m|-1} W(\rho_m[i]) = \sum_{\delta \in \Delta(S)} \text{Occ}_\delta(r_m) \cdot W(\delta). \quad (7.20)$$

そのため, 式 (7.15) で与えられる $\mathbf{x}$ に関する W の加重平均¹ $\text{WM}_\mathbf{x}(W)$ は $\mathbf{c}_y^W$ と一致する.

$$\text{WM}_\mathbf{x}(w) = \frac{\sum_{\delta \in \hookrightarrow'} \mathbf{x}[\delta] \cdot w(\delta)}{\sum_{\delta \in \hookrightarrow'} \mathbf{x}[\delta]} = \mathbf{c}_y^w. \quad (7.21)$$

よって, $\mathbf{c}_y^W$ の要素に関する閾値条件は, $\mathbf{x}$ 上ではこの加重平均に関する制約として与えられる. 具体的には, 重みインデックスの集合 $H \subseteq \{1, \dots, d\}$ に関する閾値条件 $\forall h \in H. \mathbf{c}_y^W[h] \sim 0$ は以下のような線形制約として表現できる.

$$\mathbf{A}_{\text{mp}}^{A, \langle Q', \hookrightarrow' \rangle}(H) \mathbf{x} \sim \mathbf{0}. \quad (7.22)$$

ただし, 行列 $\mathbf{A}_{\text{mp}}^{A, \langle Q', \hookrightarrow' \rangle}(H)$ は以下のように定義される $|H|$ 行 $|\hookrightarrow'|$ 列実係数行列である.

$$\mathbf{A}_{\text{mp}}^{A, \langle Q', \hookrightarrow' \rangle}(H)[h, \delta] = W[h](\delta). \quad (7.23)$$

このとき, 式 (7.22) の h 行目の制約は, $\mathbf{c}_y^W[h] = \text{WM}_\mathbf{x}(W)[h] \sim 0$ が成り立つことを表している.

¹ここでは $\mathbf{x}$ が重さ, W がデータである.

7.3.4 空判定手続き

7.3.3節より, 重みインデックス集合の四つ組 $g = \langle H^>, H^\geq, H^<, H^\leq \rangle$ と SCC $\langle Q', \hookrightarrow' \rangle$ に対して, 補題 7.9 の条件 (ii-1)-(ii-3) はそれぞれ以下の条件 (ii-1')-(ii-3') と等価である.

(ii-1') $H^< = H^\leq = \emptyset$ ならば, 線形制約 $LC_{\text{univ}}^{\mathcal{A}, \langle Q', \hookrightarrow' \rangle, g}(\mathbf{x})$ を満たす $|\hookrightarrow'|$ 次元非負ベクトル $\mathbf{x}$ が存在する.

(ii-2') $H^< \neq \emptyset$ ならば, それぞれの $h' \in H^<$ に対して線形制約 $LC_{<}^{\mathcal{A}, \langle Q', \hookrightarrow' \rangle, g, h'}(\mathbf{x}_{h'})$ を満たす $|\hookrightarrow'|$ 次元非負ベクトル $\mathbf{x}_{h'}$ が存在する.

(ii-3') $H^\leq \neq \emptyset$ ならば, それぞれの $h'' \in H^\leq$ に対して線形制約 $LC_{\leq}^{\mathcal{A}, \langle Q', \hookrightarrow' \rangle, g, h''}(\mathbf{x}_{h''})$ を満たす $|\hookrightarrow'|$ 次元非負ベクトル $\mathbf{x}_{h''}$ が存在する.

ただし, $LC_{\text{univ}}^{\mathcal{A}, \langle Q', \hookrightarrow' \rangle, g}$, $LC_{<}^{\mathcal{A}, \langle Q', \hookrightarrow' \rangle, g, h'}$, $LC_{\leq}^{\mathcal{A}, \langle Q', \hookrightarrow' \rangle, g, h''}$ は, 以下のように定義される $|\Delta(S)|$ 次元ベクトルに関する線形制約である.

$$LC_{\text{univ}}^{\mathcal{A}, \langle Q', \hookrightarrow' \rangle, g}(\mathbf{x}) \Leftrightarrow \left(\begin{array}{l} \mathbf{u}^T \mathbf{x} > 0 \wedge \mathbf{A}_{\text{cyc}}^{\mathcal{A}, \langle Q', \hookrightarrow' \rangle} \mathbf{x} = \mathbf{0} \wedge \\ \mathbf{A}_{\text{mp}}^{\mathcal{A}, \langle Q', \hookrightarrow' \rangle}(H^>) \mathbf{x} > \mathbf{0} \wedge \mathbf{A}_{\text{mp}}^{\mathcal{A}, \langle Q', \hookrightarrow' \rangle}(H^\geq) \mathbf{x} \geq \mathbf{0} \end{array} \right), \quad (7.24)$$

$$LC_{<}^{\mathcal{A}, \langle Q', \hookrightarrow' \rangle, g, h'}(\mathbf{x}) \Leftrightarrow \left(LC_{\text{univ}}^{\mathcal{A}, \langle Q', \hookrightarrow' \rangle, g}(\mathbf{x}) \wedge \mathbf{A}_{\text{mp}}^{\mathcal{A}, \langle Q', \hookrightarrow' \rangle}(\{h'\}) \mathbf{x} < \mathbf{0} \right), \quad (7.25)$$

$$LC_{\leq}^{\mathcal{A}, \langle Q', \hookrightarrow' \rangle, g, h''}(\mathbf{x}) \Leftrightarrow \left(LC_{\text{univ}}^{\mathcal{A}, \langle Q', \hookrightarrow' \rangle, g}(\mathbf{x}) \wedge \mathbf{A}_{\text{mp}}^{\mathcal{A}, \langle Q', \hookrightarrow' \rangle}(\{h''\}) \mathbf{x} \leq \mathbf{0} \right). \quad (7.26)$$

補題 7.8 と補題 7.9, 条件 (ii-1')-(ii-3') より, MPBA の空判定問題は以下のような手続きにより解くことができる.

手続き 7.10 (MTMPBA の空判定手続き)

入力: d -次元 MTMPBA $\mathcal{A} = \langle Q, \Sigma, \hookrightarrow, q_{\text{init}}, \langle F, \langle W, G \rangle \rangle \rangle$

出力: $L(\mathcal{A}) = \emptyset$ か否か

1. $G' := G$ とする.
2. $\mathcal{A}$ の SCC 分解を行い, MSCC 集合 $\text{MSCC}^{\mathcal{A}}$ を計算する.
3. $G' = \emptyset$ のとき「 $L(\mathcal{A}) = \emptyset$ 」を出力して終了する. そうでないとき, $g := g' \in G'$, $G' := G' \setminus \{g\}$ とする.
4. $M := \{\langle Q', \hookrightarrow' \rangle \mid \langle Q', \hookrightarrow' \rangle \in \text{MSCC}^{\mathcal{A}} \wedge Q' \cap F \neq \emptyset\}$ とする.

5. $M = \emptyset$ のとき, ステップ3へ. そうでないとき $\langle Q', \hookrightarrow' \rangle := \langle Q'', \hookrightarrow'' \rangle \in M$, $M := M \setminus \{\langle Q', \hookrightarrow' \rangle\}$ とする.
6. S と g に関して条件 (ii-1')-(ii-3') のすべてを満たすか判定する. 満たすとき「 $L(A) \neq \emptyset$ 」を出力して終了する. そうでないときステップ5へ.
 - 条件 (ii-1')-(ii-3') を満たす, すなわち, 線形制約式 (7.24)-(7.26) を満たす解が存在するかは, LCSP を解くことで決定する. ■

この手続きに関して, 以下の定理が成り立つ.

定理 7.11 (MTMPBA の空判定手続きの正当性と計算量) MTMPBA の空判定問題は, 手続き 7.10 により多項式時間で決定可能である. ■

証明. 補題 7.9 と条件 (ii-1')-(ii-3') より, 「MSCC $\langle Q', \hookrightarrow' \rangle$ とある重みインデックス集合の四つ組 $g \in G$ が存在し, 条件 (ii-1')-(ii-3') のすべてを満たす」ならば, かつそのときに限り, $\text{Inf}(r) \cap Q' \neq \emptyset$ かつ $\text{MP}_{W,G}(r)$ を満たす実行 r が存在する. また, 補題 7.8 より, その $\langle Q', \hookrightarrow' \rangle$ が $Q' \cap F \neq \emptyset$ ならば, かつそのときに限り, $\text{Inf}(r) \cap Q' \cap F \neq \emptyset$, $\text{Büchi}_F(r')$ かつ $\text{MP}_{W,G}(r')$ を同時に満たす実行 r' が存在する. 従って, 手続き 7.10 は正当な空判定手続きである.

ここで, SCC S と $g \in G$ に関する LCSP の変数の個数は $|\hookrightarrow'|$, 線形制約式の個数は高々 $|Q'| + 2d + 2$ である. また, SCC の極大性より $\sum_{\langle Q', \hookrightarrow' \rangle \in \text{MSCC}^A} |Q'| \leq |Q|$ である. よって, MTMPBA A の空判定問題は $|Q|$, $|G|$, d に関して多項式時間で決定可能である. □

本研究の空判定アプローチは, [6] と [19] で用いられているアプローチより効率的である. [6] と [19] の手法では, 各 MSCC $\langle Q', \hookrightarrow' \rangle$ に対して加重平均領域 $X = \{\text{WM}_{\mathbf{x}}(w) \mid \mathbf{u}^T \mathbf{x} > 0, \mathbf{A}_{\text{cyc}}^{A, \langle Q', \hookrightarrow' \rangle} \mathbf{x} = \mathbf{0}\}$ を明示的に計算した上で, 線形制約式 (7.24)-(7.26) を満たす解を探索している. この X は, 領域 $\{\mathbf{c}_{\mathbf{y}}^W \mid \mathbf{y} \in (\mathbb{R}_{\geq 0})^{|C_{\text{simp}}^A(\langle Q', \hookrightarrow' \rangle)|} \wedge \mathbf{u}^T \mathbf{y} = 1\}$ と一致し, 注目している $\langle Q', \hookrightarrow' \rangle$ 上の単純閉路の平均利得をすべて含む最小の凸包である. オートマトン上の単純閉路の個数 $|C_{\text{simp}}^A(\langle Q', \hookrightarrow' \rangle)|$ は高々 $|\hookrightarrow'|^{|Q|}$ 個であるため, X の明示的計算には $|Q|$ について指数時間が必要となる. 一方, 本研究のアルゴリズムは, X 全体を明示的に計算せずに線形制約を満たす解を直接求める. それにより, $|Q|$ について多項式時間の計算量で空判定できる.

7.4 表現力

LTL 仕様のモデル検査や充足/実現可能性判定では、仕様を満たす言語を認識する非決定性/決定性オートマトンへ変換し、それを利用して検証するというオートマトンベースの手法がしばしば用いられている。

本節では、 LTL^{mp} 文と MTMPRA の表現力に関して、以下が性質が成り立つことを示し、また、それに関連した議論を行う。

1. $\mathcal{L}(\text{DMTMPRA}) \subsetneq \mathcal{L}(\text{MTMPRA})$,
2. $\mathcal{L}(\text{LTL}^{mp}) \subseteq \mathcal{L}(\text{MTMPRA})$,
3. $\mathcal{L}(\text{LTL}^{mp}) \not\subseteq \mathcal{L}(\text{DMTMPRA})$,
4. $\mathcal{L}(\text{LTL}^{mp}) \not\supseteq \mathcal{L}(\text{DMTMPRA})$.

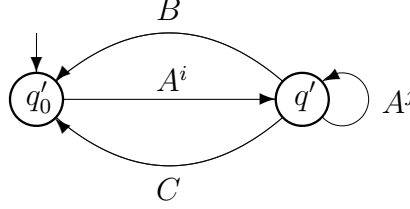
7.4.1 決定性 vs. 非決定性 MTMP ω -正規オートマトン

古典的な ω -オートマトンでは、DBA は BA より表現力が真に小さいが、DRA は RA と同じ表現力を持っている。本項では、古典的 ω -オートマトンとは異なり、DMTMPRA は MTMPRA より表現力が真に小さいことを示す。

ここで、例 7.3 の図 7.1 のオートマトンの受理言語 $L(\mathcal{A})$ を認識する DMTMPRA $\mathcal{A}' = \langle Q', 2^{\{p_1, p_2\}}, \Delta', q'_0, \langle F', \langle W', G' \rangle \rangle \rangle$ (状態数 $|Q'|$ を n とする) を考える。このとき、 $(A^+(B+C))^*L(\mathcal{A}) = L(\mathcal{A})$ であるため、 $\mathcal{A}'$ の初期状態 q'_0 から任意の有限語 $\sigma' \in (A^+(B+C))^+$ で初期状態に戻る DMTMPRA も存在しているはずである。そのため、以下を仮定しても一般性は失われない。

$$\bullet \quad q'_0 \xrightarrow{(A^+(B+C))^+} q'_0.$$

ここで無限長語 $\sigma_{acc} = (A^n B A^n C)^\omega \in \Sigma^\omega$ を考える。この σ_{acc} に対する $\mathcal{A}$ の実行の平均利得トレースは明らかに $\langle 0, 0 \rangle$ に収束する。また、その実行では q_0 を無限にしばしば訪れるため、 $\mathcal{A}$ は σ_{acc} を受理する。 $\mathcal{A}'$ は $L(\mathcal{A})$ を認識するため、 σ_{acc} は $\mathcal{A}'$ にも受理される。そして語 σ_{acc} に対する実行が存在し、その実行は $\text{Rabin}_{F'}$ と $\text{MP}_{G'}^{w'}$ を満たしているはずである。ここで、 $\mathcal{A}'$ の状態数が n であることから、 q'_0 を始点とした有限長語 A^n に対する実行断片上ではある状態 q' を複数回訪れているはずである。すなわち、ある自然数 i, j ($i \geq 0, j > 0, i+j = n$)、状態 $q' \in Q'$ と以下のような実行断片が存在する (図 7.2)。

図 7.2: $L(\mathcal{A})$ を認識する決定性 MPRA の構造的特徴.

- $q'_0 \xrightarrow{A^i} q' \xrightarrow{A^j} q' \xrightarrow{B+C} q'_0$.

$\mathcal{A}'$ は決定的であるため、以下の二つの実行断片の長さ及び累積利得は等しい.

- $q'_0 \xrightarrow{A^i} q' \xrightarrow{A^j} q' \xrightarrow{B'} q'_0 \xrightarrow{A^i} q' \xrightarrow{A^j} q' \xrightarrow{C'} q'_0$
- $q'_0 \xrightarrow{A^i} q' \xrightarrow{B'} q'_0 \xrightarrow{A^i} q' \xrightarrow{A^j} q' \xrightarrow{A^j} q' \xrightarrow{C'} q'_0$

よって、 $\mathcal{A}'$ では σ_{acc} と無限長語 $\sigma'_{acc} = (A^{n-j}BA^{n+j}C)^\omega \in \Sigma^\omega$ の利得トレースの収束値は一致している. また、無限にしばしば訪れる状態も同じであるため、 $\mathcal{A}'$ は σ'_{acc} も受理する. 一方で、 σ'_{acc} に対する $\mathcal{A}$ の実行の第1次元射影平均利得トレースは明らかに0より大きい値に収束するため、 $\mathcal{A}$ は σ'_{acc} を受理しない. これは $\mathcal{A}'$ が $L(\mathcal{A})$ を認識するという前提に反するよって、言語 $L(\mathcal{A})$ は、MTMPBA では認識できるが DMTMPRA には認識できない.

定理 7.12 ([97]) $\mathcal{L}(\text{DMTMPRA}) \subsetneq \mathcal{L}(\text{MTMPRA})$.

証明. $\mathcal{L}(\text{DMTMPRA}) \subseteq \mathcal{L}(\text{MTMPRA})$ は定義から明らか. また、上記のように、MTMPBA では認識できるが DMTMPRA には認識できない言語が存在する. MTMPBA から MTMPRA へは、MTMPBA の受理状態集合 F を Rabin 対集合 $\{\{\emptyset, F\}\}$ に変更することで容易に変換できるため、そのような言語は MTMPRA では認識できる. 以上により示された. $\square$

7.4.2 LTL^{mp} vs. 決定性 MTMP ω -正規オートマトン

LTL で記述できる言語は ω -正規言語、すなわち、DRA などが認識できる言語の真部分集合である. そして LTL リアクティブシステム仕様の実現可能性

判定 [68] では、仕様を記述した LTL 式を決定性オートマトン変換しそれを利用して判定を行うことができる。そして、既存の定量的合成においても、決定性オートマトン仕様を実現し、また、確率的環境のもとで最適な戦略を合成する手法など提案されている [33, 16, 32]。LTL^{mp} 実現可能性についても、仕様を記述した LTL^{mp} 式を認識する決定化オートマトンを構成できるならば既存の手法と同様のアプローチを適用できる²。本項では MTMPRA の認識する言語のクラスと LTL^{mp} 文によって規定される言語のクラスの関係について論じる。

LTL 式と等価な BA/RA を構成できるように、LTL^{mp} 文と等価な MTMPBA/MTMPRA を構成することができるため、以下の定理が成り立つ。

定理 7.13 ([81]) $\mathcal{L}(\text{LTL}^{\text{mp}}) \subseteq \mathcal{L}(\text{MTMPRA})$. ■

証明. 補題 8.1 より明らか. □

しかしながら、LTL と DRA の場合とは異なり、LTL^{mp} 文で記述できるが DMTMPRA では認識できない言語が存在するため、DMTMPRA は LTL^{mp} よりも高い表現力を持っていない。

定理 7.14 ([97]) $\mathcal{L}(\text{LTL}^{\text{mp}}) \not\subseteq \mathcal{L}(\text{DMTMPRA})$.

証明. ここで、以下のような LTL^{mp} 文 $\varphi_1, \dots, \varphi_5$ を考える。

$$\varphi_1 = \neg p_0 \wedge \neg p_1, \quad (7.27)$$

$$\varphi_2 = \Box(\neg p_0 \vee \neg p_1), \quad (7.28)$$

$$\varphi_3 = \Box((p_0 \vee p_1) \rightarrow \mathbf{X}(\neg p_0 \wedge \neg p_1)), \quad (7.29)$$

$$\varphi_4 = \Box\Diamond(p_0 \vee p_1) \quad (7.30)$$

$$\varphi_5 = \text{MP}(1 - 2 \cdot \mathbf{1}_{\psi_1}) \leq 0 \wedge \text{MP}(\mathbf{1}_{p_1} - \mathbf{1}_{p_0}) \geq 0, \quad (7.31)$$

$$\psi = (\neg p_0 \wedge \neg p_1)\mathbf{U}p_1. \quad (7.32)$$

このとき、 $L(\bigwedge_{1 \leq i \leq 4} \varphi_i)$ を認識するオートマトンとして、例えば、例 7.3 の MTMPBA $\mathcal{A}$ 等を得ることができる。

前項で述べたように、この MTMPBA は DMTMPRA では認識できない。以上により示された。 □

²ただし、確率的環境の想定 [33, 32] や有界実現可能性判定 [74, 47] など、LTL^{mp} 仕様の実現可能性判定の問題設定によってはオートマトンを決定化できない仕様の実現可能性判定でも決定可能となりうる。

ただし、DMTMPRA は DRA よりは表現力が高いため、 LTL^{mp} は DMTMPRA よりも高い表現力を持っていない。

定理 7.15 ([81]) $\mathcal{L}(LTL^{\text{mp}}) \not\subseteq \mathcal{L}(\text{DMTMPRA})$. ■

証明. 定理 6.8 より明らか. □

LTL^{mp} 文から等価な決定性オートマトンへの変換を保証するためには、何らかの本質的な構文制限が必要である。

単純な LTL^{mp} フラグメントとしては、未来非依存な利得項のみに制限するフラグメントが考えられる。このような単純なフラグメントであっても、6.3 節に示されている平均コストなどの実用的な定量的要求を反映する仕様を記述することができる。しかし、そのようなフラグメントに対しても、8.1.1 項に示した等価な MTMPBA 構成法 (定理 8.2) で得られるオートマトンが決定的であることを保証できない。なぜならば、補題 7.5 で示した和合成法は合成されたオートマトンを非決定的にしてしまうためである。

一方、8.1.1 項に示した MTMPBA 構成法と定理 7.14 の証明の例のように、利得項の未来依存性はしばしばそれらを含む式を認識するオートマトンを非決定的にするが、いくつかの未来依存な利得項を持つ文については決定性オートマトンによって認識することができることがある。単純な例としては、 $X\varphi$ のような有界な未来に依存する時間式の利得項をもつ平均利得式がある。このような式に対しては、オートマトン上で利得を与えるタイミングを定数ステップ遅延させることで決定化できる。なぜならば、利得を与えるタイミングを定数ステップ遅延させたとしても、利得トレースの平均利得の極限点集合は保存されるからである。そして、非有界な未来に依存する時間式の利得項の場合にも決定化できる場合がある。例として $\varphi = \neg\varphi' \wedge X(\varphi' U \varphi'')$ という式の利得項を考える。この時間式は、

1. ある時間 i で φ' が真のとき、時間 i で φ は偽。
2. ある時間 i で φ' が偽のとき、その後初めて $\neg\varphi' \vee \varphi''$ を満たす時点 j までの未来をみて、
 - 時間 j で φ'' が真のとき、時間 i で φ は真。
 - 時間 i で $\neg\varphi' \wedge \neg\varphi''$ が真のとき、時間 i で φ は偽

である。この式の場合、 i と j の間の時間 k ($i < k < j$) では明らかに φ は偽である。よって、時間 i で与えるべき利得を時間 j に延期する、あるいは、時間 i で暫定的に利得を与え時間 j で必要であれば払い戻すことで平均利得トレース

スの上極限/下極限を保存できる．なぜならば，その利得トレースの累積利得と元の利得トレースの累積利得の差が有界であるためである．つまり， φ の特性変数を含む平均利得式 ψ を含む LTL^{mp} は，時間 i からみて i での φ の真偽が決定できる未来の時間 j の間利得を与えるタイミングが有界個の時間点でしか延期されないならば，決定的オートマトンで認識できる．さらに，単体の平均利得式 ψ を認識するオートマトンが決定化不能であっても，それを含む文を決定性オートマトンで認識できる場合がある．例えば，定理 7.14 の証明中の式 φ_i ($1 \leq i \leq 4$) に，以下の式 φ_5 を加えた文 $\bigcap_{1 \leq j \leq 5} \varphi_j$ を考える．

$$\varphi_5 = \square \bigvee_{1 \leq k \leq 10} \overbrace{X \cdots X}^{k \text{ 個}} p_1 \quad (7.33)$$

この場合， ψ_1 の依存する未来の時間が φ_5 によって有界になるため， $\bigcap_{1 \leq j \leq 5} \varphi_j$ は DMTMPRA でも認識できる．また，追加する式が φ_5 でなくても ($\varphi_1, \dots, \varphi_4$ と合わせて) 結果的に φ_5 が成り立つような式であれば決定化できる．

以上のように， LTL^{mp} はいくつかの条件の下で等価な DMTMPRA に変換できる．

第8章 LTL^{mp}のモデル検査と充足可能性判定

時間論理仕様の検証では、オートマトンベースのアプローチがしばしば用いられている [85]. LTL モデル検査は、仕様 φ の否定と等価な ω -オートマトン $A_{\neg\varphi}$ とモデルの同期合成オートマトンの空判定に帰着して決定できる. なぜならば、その同期合成オートマトンの受理実行は仕様 φ に反するモデルのパスを表しているためである. また、LTL 充足可能性判定は、仕様 φ と等価な ω -オートマトン A_{φ} の非空判定に帰着して決定できる. なぜならば、その ω -オートマトンの受理実行上の語は仕様 φ を満たすためである.

本章では、まず LTL^{mp} から MTMPBA/MTMPRA への具体的な変換法を示し、LTL^{mp} のモデル検査、充足可能性判定、最適化の各手法について論じる [81]. LTL^{mp} のモデル検査と充足可能性判定は 7.3 節で与えた線形制約充足問題 (LCSP) に帰着する MTMPBA 空判定アルゴリズムを用いることで、検証を効率的に行うことができる. さらに、MTMP ω -正規オートマトンの空判定問題が LCSP に帰着して決定できることから、LTL^{mp} 仕様に関する最適化を **線形計画法** (LP 法) を用いて行うことができる. ここで、LTL^{mp} 最適化とは、「モデル上の、LTL^{mp} 仕様に従うふるまいに対する平均利得目的関数の最適化」であり、LTL^{mp} で記述される目的関数に関するモデル性能解析や仕様最適化は、この LTL^{mp} 最適化に帰着可能である.

本章の構成

まず、8.1 節で LTL^{mp} から MTMPBA/MTMPRA への具体的な変換法を示す. そして、8.2 節で LTL^{mp} モデル検査について述べる. また、8.3 節では DTMC に対する確率的 LTL^{mp} モデル検査を与える. 8.4 節では、LTL^{mp} の充足可能性判定手法、さらに、8.5 節では、最適化手法について論じる.

8.1 LTL^{mp} から MTMP ω -正規オートマトンへの変換

線形時間論理を用いた検証では、論理式と等価なオートマトンをしばしば利用している。ここで、利得系モデル $\mathcal{P} = \langle S, \rightarrow, S_{init}, AP, \Lambda, w \rangle$ と AP 上の LTL^{mp} 検査式 φ に対して、「 $\mathcal{P} \models \varphi$ か否か」をモデル検査する場合を考える。ただし、それぞれの自由変数は固定された変数対応関数 β を通して重み関数 w と関連付けられているものとする。ここで、 $\neg\varphi \Leftrightarrow \bigvee \varphi_i$ となる MPNF 式 $\bigvee \varphi_i$ の選言肢 φ_i について、以下のように定義される言語 $L(\mathcal{P}, \varphi_i)$ を考える。

$$L(\mathcal{P}, \varphi_i) = \{\text{word}_\Lambda(\rho) \in (2^{AP})^\omega \mid \rho \in \text{Paths}^\mathcal{P} \text{ and } \text{word}_\Lambda(\rho), \alpha_{w,\beta}^\rho \models \varphi_i\}. \quad (8.1)$$

言語 $L(\mathcal{P}, \varphi_i)$ は、 φ_i を満たす、すなわち、 φ に反するパス $\rho \in \text{Paths}^\mathcal{P}$ 上の語 $\text{word}_\Lambda(\rho)$ からなる言語である。よって、この言語 $L(\mathcal{P}, \varphi_i)$ に関して、以下の条件が成り立つ。

$$\mathcal{P} \models \varphi \Leftrightarrow \forall i. L(\mathcal{P}, \varphi_i) = \emptyset. \quad (8.2)$$

従って、 $\mathcal{P} \models \varphi$ を判定するためには、そのような言語 $L(\mathcal{P}, \varphi_i)$ を受理言語とする MTMPBA $\mathcal{A}_{\varphi_i}^\mathcal{P}$ を構成し、受理言語が空であるかを判定すればよい。さらに、LTL^{mp} 文と等価なオートマトンも同様の MTMPBA を和合成（定理 7.4）することで構成できる。

本節では、これらの MTMPBA の構成法について述べる。

8.1.1 連言的 MPNF 式を満たすモデルのふるまいを受理する MPBA の構成法

本項では、 n 個の平均利得リテラルを含む連言的 MPNF 式 $\varphi = \psi \wedge \bigwedge_{1 \leq m \leq n} \theta_m$ に対して、 $L(\mathcal{P}, \varphi)$ を受理言語とする MTMPBA $\mathcal{A}_\varphi^\mathcal{P}$ の構成法について述べる。

ここでは、 $\mathcal{P} = \langle S, \rightarrow, S_{init}, AP, \Lambda, w \rangle$ は d -次元利得系、 $\theta_m = \text{MP}(t_m) \sim_m 0$ は閾値 0 の平均利得リテラル、 $t_m = \sum_l c_{ml} \cdot \mathbf{1}_{\psi_{ml}} \prod_k v_{mlk}$ とする。

この $\mathcal{A}_\varphi^\mathcal{P}$ は、基本的には以下の手順で構成できる。

1. $L(\psi)$ を受理言語とする BA $\mathcal{A}_\psi$ と $\mathcal{P}$ を同期合成したオートマトンを構成する。

2. その同期オートマトンの遷移関係上の重み関数の m 番目次元がそれぞれの利得項 t_m と対応するように設定する.
3. その同期オートマトンの MTMP 受理条件の集積点条件 G を $\bigwedge_{1 \leq m \leq n} \theta_m$ に対応するように設定する.

ただし, 平均利得リテラル θ_m の利得項 t_m が未来依存であるとき, 各時間で与えられる利得は, 未来のふるまいの性質を考慮して決定する必要がある. そのため, $\mathcal{A}_\psi$ と $\mathcal{P}$ から直接的に $\mathcal{A}_\varphi^P$ を構成することは困難である.

ここで, φ 中に時間演算子を含む式 ψ_i の特性変数 1_{ψ_i} が含まれていると仮定する. このとき, $L(\mathcal{P}, \varphi)$ を受理言語とする MTMPBA を構成するために, そのような ψ_i に対応する新しい補助的命題 $p_{\psi_i} \notin AP$ を導入する. この命題を**予言的命題** (predictive proposition) と呼び, その集合を $PP(\varphi)$ と書くこととする.

$$PP(\varphi) = \{p_\psi \mid 1_\psi \text{ が } \varphi \text{ 中に出現し, かつ, } \psi \text{ は } AP \text{ 上の Boolean 式でない LTL}^{\text{mp}} \text{ 式}\} \quad (8.3)$$

そして以下のような拡張原子命題集合 $AP \cup PP(\varphi)$ 上の LTL^{mp} 式 φ' を考える.

$$\varphi' = \psi' \wedge \bigwedge_{1 \leq m \leq n} \theta'_m. \quad (8.4)$$

ただし,

$$\psi' = \psi \wedge \bigwedge_{p_{\psi_i} \in PP(\varphi)} \Box(p_{\psi_i} \leftrightarrow \psi_i), \quad (8.5)$$

$$\theta'_m = \text{MP}(t'_m) \sim_m 0, \quad (8.6)$$

$$t'_m = \sum_l c_{ml} \cdot 1_{\psi'_{ml}} \prod_k v_{mlk}, \quad (8.7)$$

$$\psi'_{ml} = \begin{cases} p_{\psi_{ml}} & \text{if } p_{\psi_{ml}} \in PP(\varphi), \\ \psi_{ml} & \text{otherwise.} \end{cases} \quad (8.8)$$

ここで新たに追加された LTL 式 $\bigwedge_{p_{\psi_i} \in PP(\varphi)} \Box(p_{\psi_i} \leftrightarrow \psi_i)$ は, 予言的命題 p_{ψ_i} と時間演算子を含む式 ψ_i の真偽が常に一致しなければならないという制約を表し

ている. よって, $\bigwedge_{p_{\psi_i} \in PP(\varphi)} \Box(p_{\psi_i} \leftrightarrow \psi_i)$ を満たす任意の語 $\sigma' \in (2^{AP \cup PP(\varphi)})^\omega$ と任意の変数割当 α に関して, 以下の条件が成り立つ.

$$\sigma', \alpha \models \varphi' \Leftrightarrow (\sigma'[0] \cap AP)(\sigma'[1] \cap AP) \dots, \alpha \models \varphi. \quad (8.9)$$

このとき, 拡張アルファベット $2^{AP \cup PP(\varphi)}$ 上の言語 $L(\psi')$ を受理言語とする BA $\mathcal{A}_{\psi'} = \langle Q, 2^{AP \cup PP(\varphi)}, \hookrightarrow, q_{init}, F \rangle$ を考える. ここで, $a'_1 \neq a''_1$ かつ $a'_1 \cap AP = a''_1 \cap AP$ となる文字 $a'_1, a''_1 \subseteq AP \cup PP(\varphi)$ が存在し, $q'_1 \xrightarrow{a'_1} q'_2$ かつ $q'_1 \xrightarrow{a''_1} q'_2$ であると仮定する. このとき, $q'_1 \xrightarrow{a'_1} q'_2$ から続く任意の遷移列 $q'_1 \xrightarrow{a'_1} q'_2 \xrightarrow{a'_2} q'_3 \dots \in (\hookrightarrow)^\omega$ は, 受理状態 F を無限にしばしば訪れることはない. なぜならば, a'_1 と a''_1 で真偽が異なる予言的命題 p_{ψ_i} が存在しており, 遷移列 $q'_1 \xrightarrow{a'_1} q'_2 \xrightarrow{a'_2} q'_3$ が受理状態 F を無限にしばしば訪れるならば, 遷移列 $q'_1 \xrightarrow{a''_1} q'_2 \xrightarrow{a'_2} q'_3$ も受理状態 F を無限にしばしば訪れるため, 式 (8.9) より語 $(a'_1 \cap AP)(a'_2 \cap AP) \dots (= (a''_1 \cap AP)(a'_2 \cap AP) \dots)$ は ψ_i かつ $\neg \psi_i$ を満たす (矛盾). 従って, $\mathcal{A}_{\psi'}$ に関して以下の条件を仮定しても一般性は失われない.

$$\begin{aligned} & \forall a'_1, a''_1 \subseteq AP \cup PP(\varphi). (\\ & (a'_1 \neq a''_1 \wedge a'_1 \cap AP = a''_1 \cap AP) \Rightarrow \forall q'_1, q'_2 \in Q'. (q'_1 \xrightarrow{a'_1} q'_2 \Rightarrow q'_1 \not\xrightarrow{a''_1} q'_2). \end{aligned} \quad (8.10)$$

このとき, $\mathcal{A}_{\psi'}$ 上では, 予言的命題 $p_{\psi_i} \in PP(\varphi)$ を通して, 未来依存な特性変数 1_{ψ_i} に対応する重みを定義できる.

よって, $\mathcal{P}$ と $\mathcal{A}_{\psi'}$ から, AP 上の n -次元 MPBA $\mathcal{A}_\varphi^P = \langle (S \cup \{s_{init}\}) \times Q, 2^{AP}, \hookrightarrow', \langle s_{init}, q_0 \rangle, \langle (S \cup \{s_{init}\}) \times F, \langle W, G, \rangle \rangle$ を構成できる. ただし,

- s_{init} は S に含まれない新しい状態,
- 遷移関係 $\hookrightarrow'$ は, 以下の二つの遷移関係の和集合,
 - 初期遷移関係 $\{ \langle \langle s_{init}, q_{init} \rangle, \Lambda(s), \langle s, q \rangle \rangle \mid s \in S_{init}, q_{init} \xrightarrow{a'} q \text{ and } \Lambda(s) = a' \cap AP \}$.
 - 非初期遷移関係 $\{ \langle \langle s_1, q_1 \rangle, \Lambda(s_2), \langle s_2, q_2 \rangle \rangle \mid s_1 \rightarrow s_2, q_1 \xrightarrow{a'} q_2 \text{ and } \Lambda(s_2) = a' \cap AP \}$.
- 遷移関係 $\hookrightarrow'$ に対する重み関数 W は, $\Lambda(s_2) = a' \cap AP$ である $\mathcal{A}_{\psi'}$ の遷移 $q_1 \xrightarrow{a'} q_2$ に対して,

$$- W[m](\langle\langle s_1, q_1 \rangle, \Lambda(s_2), \langle s_2, q_2 \rangle \rangle) = \sum_l \left(c_{ml} \cdot (\llbracket \mathbf{1}_{\psi'_{ml}} \rrbracket_{(a')^\omega}[0]) \cdot \prod_k w[\beta(v_{mlk})](s_2) \right),$$

- 集積点条件 G は以下の条件を満たすインデックス集合の四つ組の単集合 $\{\langle H^>, H^\geq, H^<, H^\leq \rangle\}$ である.

$$- H^\sim = \{m \in \{1, \dots, n\} \mid \sim_m \text{ は } \sim\}.$$

ここで、式 (8.10) より、このオートマトン上の任意の遷移について、その重みは一意に定まる。

これにより、以下の補題が成り立つ。

補題 8.1 $L(\mathcal{P}, \varphi) = L(\mathcal{A}_\varphi^P)$ ■

証明. まず、 $L(\mathcal{A}_\varphi^P) \subseteq L(\mathcal{P}, \varphi)$ を示す。遷移関係 $\hookrightarrow'$ の構成から、 $\mathcal{A}_\varphi^P$ の任意の実行 $r \in \text{Runs}^{\mathcal{A}_\varphi^P}$ に対して、 r に対応する $\mathcal{P}$ 上のパス $\rho \in \text{Paths}^P$ と r に対応する $\mathcal{A}_{\psi'}$ 上の実行 $r' \in \text{Runs}^{\mathcal{A}_{\psi'}}$ が存在する。よって、 r が $\mathcal{A}_\varphi^P$ の受理実行であるときに、 r 上の語 $\sigma = \text{word}(r) = \text{word}_\Lambda(\rho)$ について $\sigma, \alpha_{w,\beta}^\rho \models \varphi$ ($\sigma \models \psi$ かつ $\sigma, \alpha_{w,\beta}^\rho \models \bigwedge_{1 \leq m \leq n} \theta_m$) が成り立つことを示せばよい。ここで、 r が Büchi 受理条件 $\text{Büchi}_{(S \cup \{s_{init}\}) \times F}(r)$ を満たすと仮定する。このとき、 $\mathcal{A}_\varphi^P$ の受理状態集合 $(S \cup \{s_{init}\}) \times F$ と $\mathcal{A}_{\psi'}$ の構成より、 r は Büchi 受理条件 $\text{Büchi}_F(r)$ を満たす。従って、 r' 上の語 $\sigma' = \text{word}(r')$ について $\sigma' \models \psi'$ が成り立つ。また、 ψ' ならば $\bigwedge_{p_{\psi_i} \in PP(\varphi)} \Box(p_{\psi_i} \leftrightarrow \psi_i)$ が成り立つため、式 (8.9) より、 $\sigma \models \psi$ も成り立つ。このとき、 $\hookrightarrow'$ と重み関数 W の構成及び $\sigma' \models \psi'$ から、任意の m ($1 \leq m \leq n$) について、 r 上の m 番目次元成分の利得トレース $\text{payoff}_{W[m]}(r)$ と、利得項 t_m と未来非依存利得項 t'_m の解釈に関して、以下の等式が成り立つ。

$$\text{payoff}_{W[m]}(r) = \llbracket t'_m \rrbracket_{\sigma'}^{\alpha_{w,\beta}^\rho} = \llbracket t_m \rrbracket_{\sigma'}^{\alpha_{w,\beta}^\rho} = \llbracket t_m \rrbracket_{\sigma}^{\alpha_{w,\beta}^\rho}. \quad (8.11)$$

従って、 G の構成から、 r が平均利得受理条件 $\text{MP}_{W,G}(r)$ を満たすならば、 $\sigma, \alpha_{w,\beta}^\rho \models \bigwedge_{1 \leq m \leq n} \theta_m$ が成り立つ。よって、 $\text{Büchi}_{(S \cup \{s_{init}\}) \times F}(r)$ かつ $\text{MP}_{W,G}(r)$ 、すなわち、 r 上の語 σ が $\mathcal{A}_\varphi^P$ に受理されるならば、 $\sigma, \alpha_{w,\beta}^\rho \models \varphi$ が成り立つ。

次に、 $L(\mathcal{A}_\varphi^P) \supseteq L(\mathcal{P}, \varphi)$ を示す。 $L(\mathcal{P}, \varphi)$ に含まれる語 $\sigma \in L(\mathcal{P}, \varphi)$ を考える。このとき、 $L(\mathcal{P}, \varphi)$ の定義 (式 (8.1)) より、 $\mathcal{P}$ 上には $\sigma = \text{word}_\Lambda(\rho)$ となるパス ρ が存在し、 $\sigma, \alpha_{w,\beta}^\rho \models \varphi$ が成り立つ。ここで、以下のような $AP \cup PP(\varphi)$ 上の語 σ' を考える。

$$\sigma'[j] = \sigma[j] \cup \{p_{\psi_i} \in PP(\varphi) \mid \sigma, j \models \psi_i\}. \quad (8.12)$$

このような σ' は, σ に対して一意に定まり, $\bigwedge_{p_{\psi_i} \in PP(\varphi)} \Box(p_{\psi_i} \leftrightarrow \psi_i)$ を明らかに満たす. よって, $\sigma' \models \psi'$ が成り立つ. そのため, $\mathcal{A}_{\psi'}$ 上には $\sigma' = \text{word}(r')$ となる実行 r' が存在し, r' は Büchi 受理条件 $\text{Büchi}_F(r')$ を満たしている. このとき, 遷移関係 $\hookrightarrow'$ の構成から, $\mathcal{A}_{\varphi}^P$ 上には ρ と r' を同期させた実行 r が存在している. また, $\mathcal{A}_{\varphi}^P$ の受理状態集合 $(S \cup \{s_{init}\}) \times F$ の構成と $\text{Büchi}_F(r')$ より, r は Büchi 受理条件 $\text{Büchi}_{(S \cup \{s_{init}\}) \times F}(r)$ を満たす. さらに, $\hookrightarrow'$ と重み関数 W の構成及び $\sigma' \models \psi'$ から, 任意の m ($1 \leq m \leq n$) について, 式 (8.11) がここでも成り立つ. 従って, $\sigma, \alpha_{w,\beta}^r \models \bigwedge_{1 \leq m \leq n} \theta_m$ と G の構成から, r は平均利得受理条件 $\text{MP}_{W,G}(r)$ を満たす. よって, $\sigma \in L(\mathcal{P}, \varphi)$ ならば, $\mathcal{A}_{\varphi}^P$ には $\sigma = \text{word}(r)$ となる受理実行 r が存在し, σ は $\mathcal{A}_{\varphi}^P$ に受理される. $\square$

なお, LTL 式 ψ' が規定する言語 $L(\psi')$ を受理言語とする, $\mathcal{O}(2^{|\psi'|})$ の状態数をもつ BA はアルゴリズム的に生成できる [84, 85]. ここで, $\psi' = \mathcal{O}(|\varphi|)$ であるため, $\mathcal{A}_{\varphi}^P$ は $\mathcal{O}(|Q| \cdot 2^{|\varphi|})$ の状態数をもつ MPBA として得ることができる.

8.1.2 LTL^{mp} 文から等価な MTMPBA への変換

前節のオートマトン構成法を利用することで, LTL^{mp} 文が規定する言語を受理する MTMPBA を構成できる.

ここで, 全体言語 $(2^{AP})^\omega$ を生成する状態遷移系を $\mathcal{U}$ とする.

$$\mathcal{U} = \langle S_{AP}, S_{AP}^2, S_{AP}, AP, \Lambda_{AP} \rangle, \quad (8.13)$$

$$S_{AP} = \{s_a \mid a \subseteq AP\}, \quad (8.14)$$

$$\Lambda_{AP}(s_a) = a. \quad (8.15)$$

このとき, 任意の連言的 MPNF 文 φ に対して, 明らかに $L(\varphi) = L(\mathcal{U}, \varphi)$ である.

従って, 以下の定理が成り立つ.

定理 8.2 (LTL^{mp} 文から MTMPBA への変換) 任意の LTL^{mp} 文 φ に対して, $L(\varphi)$ を受理言語とする MTMPBA $\mathcal{A}_{\varphi}$ を構成できる. $\blacksquare$

証明. φ と等価な MPNF 文を $\bigvee \varphi_i$ とする. このとき, 補題 8.1 より, 以下の関係が成り立つ.

$$L(\varphi) = \bigcup_i L(\varphi_i) = \bigcup_i L(\mathcal{U}, \varphi_i) = \bigcup_i L(\mathcal{A}_{\varphi_i}^{\mathcal{U}}). \quad (8.16)$$

よって、定理 7.4 より、 $L(\varphi)$ を認識する MTMPBA $\mathcal{A}_\varphi$ は、それらの $\mathcal{A}_{\varphi_i}^U$ の和合成により構成できる。 $\square$

8.2 モデル検査

本節では、 LTL^{mp} モデル検査手法について述べる。

式 (8.2), 補題 8.1, 定理 7.11 より、 LTL^{mp} モデル検査は以下の手続きを用いて行うことができる。

手続き 8.3 (LTL^{mp} モデル検査手続き)

入力: d -次元利得系 $\mathcal{P}$ と LTL^{mp} 式 φ

- ただし、それぞれの自由変数は固定された変数対応関数 β を通して重み関数 w と関連付けられており、 φ 中出现するすべての自由変数 v について $1 \leq \beta(v) \leq d$ とする。

出力: $\mathcal{P} \models \varphi$ か否か

1. φ の否定 $\neg\varphi$ を等価な MPNF 文 $\bigvee_i \varphi_i$ に変換し、 $D := \bigcup_i \{\varphi_i\}$ とする。
2. $D = \emptyset$ のとき、「 $\mathcal{P} \models \varphi$ 」を出力して終了する。そうでないとき、 $\varphi' := \varphi_i \in D$, $D := D \setminus \{\varphi'\}$ とする。
3. $L(\mathcal{P}, \varphi')$ を受理言語とする MPBA $\mathcal{A}_{\varphi'}^{\mathcal{P}}$ を構成する。
4. $\mathcal{A}_{\varphi'}^{\mathcal{P}}$ の空判定を行う。 $L(\mathcal{A}_{\varphi'}^{\mathcal{P}}) \neq \emptyset$ のとき、「 $\mathcal{P} \not\models \varphi$ 」を出力して終了する。そうでないとき、ステップ 2 へ。

- この空判定は手続き 7.10 で行う。 $\blacksquare$

定理 8.4 (LTL^{mp} モデル検査) 利得系モデルの LTL^{mp} モデル検査は手続き 8.3 を用いて指数時間で決定可能である。 $\blacksquare$

証明. 式 (8.2) が示すように、 $\mathcal{P} \models \varphi$ ならば、かつそのときに限り、 $\neg\varphi$ と等価な MPNF 式 $\bigvee_i \varphi_i$ のすべての選言肢 φ_i について $L(\mathcal{P}, \varphi_i) = \emptyset$ である。ここで、補題 8.1 より、MPBA $\mathcal{A}_{\varphi_i}^{\mathcal{P}}$ は $L(\mathcal{P}, \varphi_i)$ を受理言語としている。よって、手続き 8.3 は正当な LTL^{mp} モデル検査手続きである。

8.1.1 項で示した $\mathcal{A}_{\varphi_i}^{\mathcal{P}}$ の構成法より、 $\mathcal{A}_{\varphi_i}^{\mathcal{P}}$ の状態数は、 $|Q| \cdot 2^{|\varphi_i|}$ である。定理 7.11 より、 $\mathcal{A}_{\varphi_i}^{\mathcal{P}}$ の空判定は、状態数 $|Q| \cdot 2^{|\varphi_i|}$ に関して多項式時間で決定で

きる. ここで, $\neg\varphi$ と等価な MPNF の選言肢の数は高々 2^n 個である. よって, このモデル検査手続きの計算量は $|Q|$ に対して多項式時間かつ $|\varphi|$ と n に対して指数時間である. $\square$

注意 8.5 ([19] の LTL^{lim} モデル検査アルゴリズムの計算量) [19] で提案されている LTL^{lim} (注意 6.6) のモデル検査アルゴリズムは [6] をもとにしており, そのアルゴリズムの時間計算量はモデルのサイズ $|P|$ について指数, 検査式のサイズ $|\varphi|$ について二重指数, 平均利得式の数 n について三重指数であると概算されていた.

一方, 本節で与えた手法では, 解空間を明示的に計算せずに線形制約の解を直接求める. それにより, より表現力が高い LTL^{mp} のモデル検査が, $|P|$ について多項式, $|\varphi|$ と n について指数の時間計算量で決定できる. $\blacksquare$

8.3 確率的モデル検査

本節では, 利得付き DTMC に対する LTL^{mp} モデル検査について述べる. 離散時間 PFLTL モデル検査と同様に, このモデル検査も LTL の確率的モデル検査/充足確率問題に帰着できる.

PFLTL モデル検査においては, 補題 5.3 を利用してそれぞれの BSCC に到達したほとんどすべてのパス上の時間式の充足頻度を捉えることができた. 確率的 LTL^{mp} モデル検査でも同様に, それらのパス上の平均利得を捉えることができる.

補題 8.6 $\mathcal{D} = \langle S, P, P_{init}, AP, \Lambda, w \rangle$ を利得付き DTMC, $t = \sum_i c_i \cdot \mathbf{1}_{\psi_i} \cdot \prod_j v_{ij}$ を LTL^{mp} 利得項とする. このとき, $\mathcal{D}$ の BSCC $\langle B, \hookrightarrow \rangle \in \text{BSCC}^{\mathcal{D}}$ に到達するほとんどすべてのパス ρ 上の語 $\sigma = \text{word}_{\Lambda}(\rho)$ に対して, 利得項 t の平均利得は $\text{MP}_B(t)$ は以下のように収束する.

$$\text{MP}_B(t) = \sum_{s \in B} \left(\text{Freq}_s^{\mathcal{D}}(B) \cdot \left(\sum_i c_i \cdot \text{Prob}_s^{\mathcal{D}}(\psi_i) \cdot \prod w[\beta(v_{ij})] \right) \right) \quad (8.17)$$

証明. 補題 5.3 と同様である. $\square$

ここで, 平均利得式 $\psi = \text{MP}(t) \sim c$ に対して, $\text{MP}_B(t) \sim c$ を満たす $\mathcal{D}$ 上の BSCC $\langle B, \hookrightarrow \rangle$ の和集合を $\text{SatBSCC}_{\text{MP}}^{\mathcal{D}}(\psi)$ と書くこととする.

$$\text{SatBSCC}_{\text{MP}}^{\mathcal{D}}(\psi) = \{B \mid B \text{ は } \mathcal{D} \text{ 上の BSCC} \wedge \text{MP}_B(t) \sim c\}. \quad (8.18)$$

このとき、補題 8.6 から、利得付き DTMC のほとんどすべてのパス上では、「平均利得式 ψ を充足すること」と「 $\text{SatBSCC}_{\text{MP}}^{\mathcal{D}}(\psi)$ へ到達すること」は一致する。よって、補題 5.4 と同様の定理が成り立つ。

補題 8.7 $\mathcal{D} = \langle S, \mathbf{P}, P_{\text{init}}, AP, \Lambda, w \rangle$ を利得付き DTMC, $\psi = \text{MP}(t) \sim c$ を平均利得式とする。このとき、以下が成り立つ。

$$\mathcal{D}' \models \Diamond p_\psi \leftrightarrow \psi. \quad (8.19)$$

ただし、 $\mathcal{D}' = \langle S, \mathbf{P}, P_{\text{init}}, AP \cup \{p_\psi\}, \Lambda', w \rangle$, p_ψ は新しい補助命題、 Λ' は以下のように定義されるラベリング関数とする。

$$\Lambda'(s) = \begin{cases} \Lambda(s) \cup \{p_\psi\} & \text{if } s \in \text{SatBSCC}_{\text{MP}}^{\mathcal{D}}(\psi), \\ \Lambda(s) & \text{otherwise.} \end{cases} \quad (8.20)$$

さらに、 $\mathcal{D}'$ とその任意の状態 s , PFLTL パス式 φ , $\varphi' = \varphi[\psi/\Diamond p_\psi]$ について、以下を満たす。

$$\text{Prob}_s^{\mathcal{D}}(\varphi) = \text{Prob}_s^{\mathcal{D}'}(\varphi') \quad (8.21)$$

■

証明. 補題 5.4 の証明と同様 □

離散時間 PFLTL モデル検査/充足確率問題の場合と同様に、平均利得式 ψ に注目して、補題 8.7 の対 $\langle \mathcal{D}, \varphi \rangle$ から得られる対 $\langle \mathcal{D}', \varphi' \rangle$ を $\langle \mathcal{D}, \varphi \rangle \downarrow_\psi$ とするこの変形を繰り返すことで、DTMC に対する LTL^{mp} モデル検査/充足確率問題を古典的 LTL のそれらの問題に帰着できる。よって、定理 5.5 と同様に以下の定理が成り立つ。

定理 8.8 DTMC に対する確率的 LTL^{mp} モデル検査/充足確率問題は指数時間で決定可能/計算可能である。

証明. 定理 5.5 の証明とほぼ同様である。異なる点は、PFLTL では U や Q に関して時間有界な式を記述できたが、LTL^{mp} 式は時間非有界な U 式のみしか記述できない点である。

以上から、全体では $|\mathcal{D}|$ について多項式時間、 $|\varphi|$ について指数時間、 n について線形時間で確率的モデル検査/充足確率問題を決定/計算できる。 □

8.4 充足可能性判定

本節では、LTL^{mp} 文の充足可能性判定手法について述べる。

LTL 充足可能性判定問題と同様に、LTL^{mp} 充足可能性判定問題「LTL^{mp} 文 φ が充足可能か否か」は LTL^{mp} モデル検査問題「 $\mathcal{U} \models \neg\varphi$ か否か」に帰着できる [81]。ただし、 $\mathcal{U}$ は式 (8.13) で与えられる全体言語 $(2^{AP})^\omega$ を生成する状態遷移系である。

手続き 8.9 (LTL^{mp} 充足可能性判定手続き)

入力：LTL^{mp} 文 φ

出力： φ が充足可能か否か

1. 「 $\mathcal{U} \models \neg\varphi$ か否か」をモデル検査する。「 $\mathcal{U} \models \neg\varphi$ 」であるならば「 φ は充足不能」を出力して終了する。そうでないとき、「 φ は充足可能」を出力して終了する。

- このモデル検査は手続き 8.3 で行う。 ■

定理 8.10 (LTL^{mp} 充足可能性判定問題) LTL^{mp} 充足可能性判定問題は手続き 8.9 により指数時間で決定可能である。 ■

証明. 定理 8.4 で示された手続き 8.3 の正当性及び計算量より明らか。 □

注意 8.11 (LTL_{MP} 充足可能性) [17] では、LTL_{MP} (注意 6.7) については、その仕様の有限メモリでの実現可能性判定が $\text{co-N}(2\text{EXPTIME})$ で決定可能であることが示されている。充足可能性は実現可能性より弱い性質であるため、LTL_{MP} 充足可能性判定にその実現可能性判定手法を利用することもできる。ただし、それは実現可能性判定のための手法であるため、不要な処理に大きな計算量が必要になってしまっている。

一方で本節で与えた手法では、より表現力が大きい LTL^{mp} の充足可能性判定問題を指数時間で決定できる。 ■

8.5 最適化

定量的拡張の最も大きな利点は、定量的解析と最適化が可能となることである。例えば、確率的モデル検査器 PRISM[70] は、様々なモデルと仕様に関して定量的解析ができる。また、最適プログラムを得ることができる定量的合成手法 [33, 16, 32, 86] がいくつも開発されている。

本節では、以下のように定義される LTL^{mp} 最適化問題を解く手法について述べる。

定義 8.12 (LTL^{mp} 最適化問題) d -次元利得系 $\mathcal{P} = \langle S, \rightarrow, S_{init}, AP, \Lambda, w \rangle$ と LTL^{mp} 式 φ に対して、以下の最大化/最小化問題を LTL^{mp} 最適化問題と呼ぶ。

$$\begin{aligned} & \text{maximize/minimize}_{\rho \in \text{Paths}^{\mathcal{P}}} : \llbracket \text{MP}(t) \rrbracket_{\text{word}_{\Lambda}(\rho)}^{\alpha_{w,\beta}^{\rho}}[0], \\ & \text{subject to : } \text{word}_{\Lambda}(\rho), \alpha_{w,\beta}^{\rho} \models \varphi. \end{aligned}$$

ただし、それぞれの自由変数は固定された変数対応関数 β を通して重み関数 w と関連付けられており、 φ 中に出現するすべての自由変数 v について $1 \leq \beta(v) \leq d$ とする。 ■

この問題は、直観的には、利得系モデル $\mathcal{P}$ のパス ρ のうち、その上の語 $\text{word}_{\Lambda}(\rho)$ が仕様 φ を満たすものの中で目的関数 $\text{MP}(t)$ を最適化する問題である。また、LTL^{mp} 平均利得目的関数 $\text{MP}(t)$ に関するモデル性能解析問題（仕様最適化問題）は、 $\varphi = \top$ ($\mathcal{P} = \mathcal{U}$ かつ φ を文) としたときの LTL^{mp} 最適化問題に帰着できる。

- LTL^{mp} モデル性能解析問題
 - 利得系モデル $\mathcal{P}$ に従う LTL^{mp} 平均利得目的関数 $\text{MP}(t)$ に関する最適化問題

$$\text{maximize/minimize}_{\rho \in \text{Paths}^{\mathcal{P}}} : \llbracket \text{MP}(t) \rrbracket_{\text{word}_{\Lambda}(\rho)}^{\alpha_{w,\beta}^{\rho}}[0].$$

- LTL^{mp} 仕様最適化問題
 - LTL^{mp} 文仕様 φ に従う LTL^{mp} 平均利得目的関数 $\text{MP}(t)$ に関する最適化問題

$$\begin{aligned} & \text{maximize/minimize}_{\sigma \in (2^{AP})^{\omega}} : \llbracket \text{MP}(t) \rrbracket_{\sigma}[0], \\ & \text{subject to : } \sigma \models \varphi. \end{aligned}$$

この LTL^{mp} 最適化問題は、以下のような手順で解くことができる。

1. φ と等価な MPNF 式 $\bigvee_i \varphi_i$ のそれぞれの選言肢 φ_i に対して、 $L(\mathcal{P}, \varphi_i)$ を受理言語とする MTMPBA $\mathcal{A}_{\varphi_i}^{\mathcal{P}}$ を構成する。

2. それぞれの $\mathcal{A}_{\varphi_i}^{\mathcal{P}}$ について, $\llbracket \text{MP}(t) \rrbracket_{\text{word}_{\Lambda}(\rho)}^{\alpha_{w,\beta}^{\rho}}[0]$ の最適値を計算する. ただし,
 - $\rho \in \text{Paths}^{\mathcal{P}}$ は $\mathcal{A}_{\varphi_i}^{\mathcal{P}}$ 上の受理実行 r と対応する $\mathcal{P}$ のパスとする.
3. ステップ2で求めたそれぞれの $\mathcal{A}_{\varphi_i}^{\mathcal{P}}$ の最適値の中で最適な値を選ぶ.

このステップ2は, MPBA 空判定手続き中で解いていた LCSP に平均利得目的関数 $\text{MP}(t)$ と対応した目的関数を加えた最適化問題を解いた結果を利用して計算できる. ここで利得項 $t = \sum_l (c_l \cdot \mathbf{1}_{\psi_l} \cdot \prod_k v_{lk})^1$ が未来非依存であるとき, $\mathcal{A}_{\varphi_i}^{\mathcal{P}}$ 上で t に対応する重み関数 $W_{t, \mathcal{A}_{\varphi_i}^{\mathcal{P}}}$ は, $\mathcal{A}_{\varphi_i}^{\mathcal{P}}$ 上の遷移 $\langle \langle s_1, q_1 \rangle, \Lambda(s_2), \langle s_2, q_2 \rangle \rangle$ に対して以下のように定義される1次元重み関数である.

$$W_{t, \mathcal{A}_{\varphi_i}^{\mathcal{P}}}(\langle \langle s_1, q_1 \rangle, \Lambda(s_2), \langle s_2, q_2 \rangle \rangle) = \sum_l \left(c_l \cdot (\llbracket \mathbf{1}_{\psi_l} \rrbracket_{(\Lambda(s_2))^{\omega}}[0]) \cdot \prod_k w[\beta(v_{lk})](s_2) \right). \quad (8.22)$$

ここで, $\mathcal{A}_{\varphi_i}^{\mathcal{P}}$ と重み関数 $W_{t, \mathcal{A}_{\varphi_i}^{\mathcal{P}}}$ の構成から,

- $\mathcal{A}_{\varphi_i}^{\mathcal{P}}$ 上の実行 r の利得トレース $\text{payoff}_{W_{t, \mathcal{A}_{\varphi_i}^{\mathcal{P}}}}(r)$ と
- r に対応する $\mathcal{P}$ のパス ρ 上の t の解釈 $\llbracket t \rrbracket_{\text{word}_{\Lambda}(\rho)}^{\alpha_{w,\beta}^{\rho}}$

は一致している. よって, 7.3.3 節の線形制約式 (7.24)-(7.26) と等価な境界を含まない線形制約 (注意 8.13) に従う加重平均 $\text{WM}_{\mathbf{x}}(w_{t, \mathcal{A}_{\varphi_i}^{\mathcal{P}}})$ を目的関数とした**線形分数計画問題** (linear fractional programming problem, LFPP) を解くことで $\text{MP}(t)$ を最適化できる.

注意 8.13 (境界を含む線形制約への変換) 式 (7.24)-(7.26) より, 非負ベクトル $\mathbf{x}$ が $LC_{\text{univ}}^{\mathcal{A}, \langle Q', \hookrightarrow \rangle, g}$ ($LC_{<}^{\mathcal{A}, \langle Q', \hookrightarrow \rangle, g, h'}$, $LC_{\leq}^{\mathcal{A}, \langle Q', \hookrightarrow \rangle, g, h''}$) を満たす解であるとき, 任意の非負実定数 $c > 1$ に対して, $c\mathbf{x}$ もまたその解である. よって, $LC_{\text{univ}}^{\mathcal{A}, \langle Q', \hookrightarrow \rangle, g}$ ($LC_{<}^{\mathcal{A}, \langle Q', \hookrightarrow \rangle, g, h'}$, $LC_{\leq}^{\mathcal{A}, \langle Q', \hookrightarrow \rangle, g, h''}$) を満たす解 $\mathbf{x}$ が存在するならば, かつそのときに限り, 以下のように定義される線形制約 $BLC_{\text{univ}}^{\mathcal{A}, \langle Q', \hookrightarrow \rangle, g}$ ($BLC_{<}^{\mathcal{A}, \langle Q', \hookrightarrow \rangle, g, h'}$, $BLC_{\leq}^{\mathcal{A}, \langle Q', \hookrightarrow \rangle, g, h''}$) を満たす解 $\mathbf{x}'$ が存在する. さらに, この最適化問題の目的関

¹任意の利得項は $\sum_l (c_l \cdot \mathbf{1}_{\psi_l} \cdot \prod_k v_{lk})$ の形で表現可能である.

数である加重平均 WM に関しては, $WM_{\mathbf{x}}(W_{t, \mathcal{A}_{\varphi_i}^{\mathcal{P}}})$ と $WM_{\mathbf{x}'}(W_{t, \mathcal{A}_{\varphi_i}^{\mathcal{P}}})$ は等しい.

$$BLC_{\text{univ}}^{\mathcal{A}, \langle Q', \hookrightarrow \rangle, g}(\mathbf{x}) \Leftrightarrow \left(\begin{array}{l} \mathbf{u}^T \mathbf{x} \geq 1 \wedge \mathbf{A}_{\text{cyc}}^{\mathcal{A}, \langle Q', \hookrightarrow \rangle} \mathbf{x} = \mathbf{0} \\ \wedge \mathbf{A}_{\text{mp}}^{\mathcal{A}, \langle Q', \hookrightarrow \rangle} (H >) \mathbf{x} \geq \mathbf{u} \wedge \mathbf{A}_{\text{mp}}^{\mathcal{A}, \langle Q', \hookrightarrow \rangle} (H \geq) \mathbf{x} \geq \mathbf{0} \end{array} \right), \quad (8.23)$$

$$BLC_{<}^{\mathcal{A}, \langle Q', \hookrightarrow \rangle, g, h'}(\mathbf{x}) \Leftrightarrow \left(BLC_{\text{univ}}^{\mathcal{A}, \langle Q', \hookrightarrow \rangle, g}(\mathbf{x}) \wedge -\mathbf{A}_{\text{mp}}^{\mathcal{A}, \langle Q', \hookrightarrow \rangle}(\{h'\}) \mathbf{x} \geq \mathbf{u} \right), \quad (8.24)$$

$$BLC_{\leq}^{\mathcal{A}, \langle Q', \hookrightarrow \rangle, g, h''}(\mathbf{x}) \Leftrightarrow \left(BLC_{\text{univ}}^{\mathcal{A}, \langle Q', \hookrightarrow \rangle, g}(\mathbf{x}) \wedge -\mathbf{A}_{\text{mp}}^{\mathcal{A}, \langle Q', \hookrightarrow \rangle}(\{h''\}) \mathbf{x} \geq \mathbf{0} \right). \quad (8.25)$$

■

平均利得目的関数 $MP(t)$ の利得項 t が未来非依存である場合の LTL^{mp} 最適化手続きは以下のように与えられる.

手続き 8.14 (LTL^{mp} 最適化手続き)

入力: 利得系 $\mathcal{P} = \langle S, \neg, S_{\text{init}}, AP, \Lambda, w \rangle$, LTL^{mp} 式 φ , 平均利得項 $MP(t)$, $\text{optimize} \in \{\text{maximize}, \text{minimize}\}$

- ただし, $t = \sum_l (c_l \cdot \mathbf{1}_{\psi_l} \cdot \prod_k v_{lk})$ であり, t は未来非依存であると仮定する.

$$\text{出力: } \begin{cases} \sup_{\rho \in \text{Paths}^{\mathcal{P}}} \{ \llbracket MP(t) \rrbracket_{\text{word}_{\Lambda}(\rho)}^{\alpha_{w, \beta}^{\rho}}[0] \mid \text{word}_{\Lambda}(\rho), \alpha_{w, \beta}^{\rho} \models \varphi \} & \text{if optimize が maximize, and } \mathcal{P} \not\models \neg \varphi, \\ \inf_{\rho \in \text{Paths}^{\mathcal{P}}} \{ \llbracket MP(t) \rrbracket_{\text{word}_{\Lambda}(\rho)}^{\alpha_{w, \beta}^{\rho}}[0] \mid \text{word}_{\Lambda}(\rho), \alpha_{w, \beta}^{\rho} \models \varphi \} & \text{if optimize が minimize, and } \mathcal{P} \not\models \neg \varphi, \\ -\infty & \text{otherwise.} \end{cases}$$

1. $v := -\infty$ とする.
2. φ を等価な MPNF 文 $\bigvee \varphi_i$ に変換し, $D := \bigcup \{\varphi_i\}$ とする.
3. $D = \emptyset$ のとき, v を出力して終了する. そうでないとき, $\varphi' := \varphi_i \in D$, $D := D \setminus \{\varphi'\}$ とする.
4. 言語 $L(\mathcal{P}, \varphi')$ を受理言語とする MPBA $\mathcal{A}_{\varphi'}^{\mathcal{P}} = \langle Q, 2^{AP}, \hookrightarrow, q_{\text{init}}, \langle F, \langle W, \{g\} \rangle \rangle \rangle$ を構成する.

- ただし, $g = \langle H^>, H^\geq, H^<, H^\leq \rangle$ とする.
- 5. $\mathcal{A}_{\varphi'}^P$ の SCC 分解を行い, $M := \{\langle Q', \hookrightarrow' \rangle \in \text{MSCC}^{\mathcal{A}_{\varphi'}^P} \mid F \cap Q' \neq \emptyset\}$ とする.
- 6. $M = \emptyset$ のとき, ステップ 3 へ. そうでないとき, $\langle Q', \hookrightarrow' \rangle := \langle Q'', \hookrightarrow'' \rangle \in M$, $M := M \setminus \{\langle Q', \hookrightarrow' \rangle\}$ とする.
- 7. $\mathcal{A}_{\varphi'}^P$ 上の $\langle Q', \hookrightarrow' \rangle$ と g に関して, 7.3.4 節の条件 (ii-1')-(ii-3') のすべてを満たすか判定する. もし満たさないならば, ステップ 6 へ.
 - 手続き 7.10 のステップ 6 と同様.
- 8. 「 $H^< = H^\leq = \emptyset$ 」または「optimize が minimize」であるときには (a) を, どちらでもないときには (b) を行い, v' を計算する.
 - (a) $|\hookrightarrow'|$ -次元非負ベクトル $\mathbf{x}$ に関する以下の LFPP の最適値を v' とする.

$$\begin{aligned} \text{optimize}_{\mathbf{x}} : & \text{WM}_{\mathbf{x}}(W_{t, \mathcal{A}_{\varphi'}^P}), \\ \text{subject to} : & \text{BLC}_{\text{univ}}^{\mathcal{A}_{\varphi'}^P, \langle Q', \hookrightarrow' \rangle, g}(\mathbf{x}). \end{aligned}$$

- (b) それぞれの $h \in H^\triangleleft$ に対して, $|\hookrightarrow'|$ -次元非負ベクトル $\mathbf{x}_h$ に関する以下の LFPP の最大値を計算し, その中の最小値を v' とする.

$$\begin{aligned} \text{maximize}_{\mathbf{x}_h} : & \text{WM}_{\mathbf{x}_h}(W_{t, \mathcal{A}_{\varphi'}^P}), \\ \text{subject to} : & \text{BLC}_{\triangleleft}^{\mathcal{A}_{\varphi'}^P, \langle Q', \hookrightarrow' \rangle, g, h}(\mathbf{x}_h). \end{aligned}$$

- 9. $v < v'$ ならば $v := v'$ とする.

- 10. ステップ 6 へ. ■

目的関数 $\text{MP}(t)$ の利得項 t が未来依存である場合でも, 未来非依存な利得項 t' をもつ目的関数の LTL^{mp} 最適化問題に帰着できる. 具体的には, 利得系 $\mathcal{P} = \langle S, \rightarrow, S_{\text{init}}, AP, \Lambda, w \rangle$ の実行のうち AP 上の LTL^{mp} 式 φ を満たす実行上の $\text{MP}(t)$ の最適化問題は, 利得系 $\mathcal{P}' = \langle S \times 2^{PP(\text{MP}(t) \sim 0)}, \rightarrow', S_{\text{init}}, 2^{PP(\text{MP}(t) \sim 0)}, AP \cup PP(\text{MP}(t) \sim 0), \Lambda', w' \rangle$ の実行のうち $AP \cup PP(\text{MP}(t) \sim 0)$ 上の LTL^{mp} 式 φ'

を満たす実行上の $\text{MP}(t')$ の最適化問題に帰着できる. ただし, $\rightarrow'$, Λ' , w' は, 以下の条件を満たす遷移関係, ラベリング関数, 重み関数である.

$$\langle s_1, a_1 \rangle \rightarrow' \langle s_2, a_2 \rangle \Leftrightarrow s_1 \rightarrow s_2, \quad (8.26)$$

$$\Lambda'(\langle s_1, a_1 \rangle) = \Lambda(s) \cup a_1, \quad (8.27)$$

$$w'(\langle \langle s_1, a_1 \rangle, \langle s_2, a_2 \rangle \rangle) = w(\langle s_1, s_2 \rangle). \quad (8.28)$$

また, φ' と t' はそれぞれ以下のように定義される.

$$\varphi' = \varphi \wedge \bigwedge_{p_{\psi_i} \in PP(\text{MP}(t) \sim 0)} \Box(p_{\psi_i} \leftrightarrow \psi_i), \quad (8.29)$$

$$t' = t[\mathbf{1}_{p_{\psi_i}} / \mathbf{1}_{\psi_i} \text{ for each } p_{\psi_i} \in PP(\text{MP}(t) \sim 0)]. \quad (8.30)$$

よって, 以下の定理が成り立つ.

定理 8.15 (LTL^{mp} 最適化問題) LTL^{mp} 最適化問題は手続き 8.14 により指数時間で決定可能である. ■

証明. $\mathcal{P} \models \neg\varphi$ であるときには, この手続きは明らかに $-\infty$ を出力する. そこで, そうでないときに, ステップ 8 において求めた値 v' が $\text{Inf}(r) \cap Q' \cap F \neq \emptyset$ かつ $\text{MP}_{W, \{g\}}(r)$ を満たす実行 $r \in \text{Runs}^{A_{\varphi'}}(r)$ の中で, 重み $W_{t, A_{\varphi'}}(r)$ が与える平均利得トレースの下極限 $\liminf_{n \rightarrow \infty} \text{mp}_{W_{t, A_{\varphi'}}}(r)[n]$ に関する最適値であることを示す.

ここで, $\text{Inf}(r) \cap Q' \cap F \neq \emptyset$ かつ $\text{MP}_{W, \{g\}}(r)$ を満たす実行 $r \in \text{Runs}^{A_{\varphi'}}(r)$ の集合を R と書くこととする. 実行 $r \in R$ 上では, 「すべての $h \in H^<$ について, $LC_{\triangleleft}^{A_{\varphi'}, \langle Q', \hookrightarrow' \rangle, h}(\mathbf{x}_h)$ を満たす遷移出現比 $\mathbf{x}_h$ の任意の近傍を無限にしばしば通る」ように遷移出現比が変化しなければならない. ただし, 「 $LC_{\text{univ}}^{A_{\varphi'}, \langle Q', \hookrightarrow' \rangle}(\mathbf{x})$ を満たす任意の遷移出現比 $\mathbf{x}$ 」の任意の近傍を無限にしばしば通るように遷移出現比が変化してもよい. よって, 「 $H^< = H^{\leq} = \emptyset$ 」であるとき, (a) で計算した v' は明らかに目的の最適値である. また, 「optimize が minimize」であるとき, この最適化問題は, 下極限の最小化問題である. このとき, 「 $LC_{\text{univ}}^{A_{\varphi'}, \langle Q', \hookrightarrow' \rangle}(\mathbf{x})$ を満たし, 目的関数を最小化する遷移出現比 $\mathbf{x}$ 」の任意の近傍を無限にしばしば通るように遷移出現比が変化する実行も R には含まれている. よって, (a) で計算した v' は目的の最適値である. 一方, 「 $H^< \cup H^{\leq} \neq \emptyset$ 」かつ 「optimize が maximize」であるとき, この最適化問題は, 下極限の最大化問題である. このとき, $r \in R$ 上では, 「すべての $h \in H^<$ について, $LC_{\triangleleft}^{A_{\varphi'}, \langle Q', \hookrightarrow' \rangle, h}(\mathbf{x}_h)$ を満

たす遷移出現比 x_h の任意の近傍を無限にしばしば通る」ように r 上で遷移出現比が変化しなければならない。そのため、(b) で求めた v' が最適値となる。よって、手続き 8.14 は正当な LTL^{mp} 最適化手続きである。

それぞれの LFPP は、線形制約式 (7.24)-(7.26) の充足判定と同様に、強連結成分の大きさ $|Q'|$ と平均利得リテラルの数 n について多項式時間で計算できる。従って、この LTL^{mp} 最適化手続き全体の計算量は、 $|S|$ に関して多項式時間、 $|\varphi|$ と n について指数時間である。□

注意 8.16 (応用的な最適化) ここでは単一平均利得目的関数を扱ったが、多目的最適化や平均利得比最適化など、他種の LTL^{mp} 最適化問題も LP 関連手法を用いて解くこともできる。■

第9章 LTL^{mp}仕様の平均ケースの実現可能性と最適合成

安全にプログラムを生成するための手法として、オートマトンや時間論理により形式的に規定されたリアクティブシステム仕様からプログラムを自動生成するアプローチがある。古典的な合成 [23, 68] では仕様は LTL や ω -正規オートマトンで与えられるが、これらの表現できる性質は定性的なものであり、イベント生起順序などの「システムが備えるべき機能」しか規定できなかった。

近年、平均利得に注目した定量的合成手法がいくつも提案されている [16, 32, 86, 86, 18]。それらの手法では、単に仕様を満たすか否かに注目するのではなく、仕様を満たし、かつ、与えられた目的関数に関して最適な性能をもつプログラムを自動合成することを目的としている。

定量的合成は大きく分けて二つの手法に分けられる。一つは**最悪ケース定量的合成** (worst-case quantitative synthesis) [16, 86, 18] である。この手法では、どのような環境であつても少なくとも保証される性能が最適になるようなプログラムを合成する。そのため、具体的な使用状況があらかじめ分からない場合に有効な手法である。古典的な実現可能性判定/合成でも同様に、このような最悪な場合を想定する立場をとることが一般的である。そして、実現可能性判定器 Acacia+ (ver.2) [2, 17] では、LTL の平均利得拡張である LTL_{MP} の有限メモリ最悪ケース実現可能性判定/合成を行うことができる。ただし、LTL_{MP} は LTL^{mp} より表現力が低いものであり、また、合成手法も最適化を自動化できていない。

もう一つは**平均ケース定量的合成** (average-case quantitative synthesis) [32, 86] である。こちらの手法では、ある具体的な確率的環境を想定し、その環境のもとでの平均的な性能が最適になるようなプログラムを合成する。ただし、想定とは大きく異なる使用状況では性能が著しく低下する場合もある。しかし、待ち行列処理システムなどの実用的なシステムは、ある特定の使用状況を想定しその中で高性能、高効率、低コストで運用できるように設計されることが多く、この場合には非常に有効である。

本章では、LTL^{mp} 仕様の平均ケースの実現可能性判定と最適合成について論じる [98]. 前章では LTL^{mp} のモデル検査と充足可能性判定が LTL でも有効なオートマトンベースのアプローチで決定でき、さらに充足可能な範囲での最適化が計算できることを示した. しかしながら、LTL^{mp} 実現可能性判定を、古典的な LTL 実現可能性判定と同様のアプローチで行うことはできない. なぜならば、古典的な LTL 実現可能性は仕様と等価な決定性オートマトンを合成しそれを用いて判定を行うが、7.4.2 項で示したように、LTL^{mp} は一般に決定性オートマトンでは認識できないからである. そこで本研究ではまず、実現可能性判定のための LTL^{mp} フラグメント LTL^{mp-} を定義する. LTL^{mp-} は LTL_{MP} より表現力が高く、また、それによって規定される言語は多閾値平均利得 Rabin オートマトン (DMTMPRA) で認識できる. そのため古典的なアプローチと同様に、その仕様 φ の実現可能性判定問題は $L(\varphi)$ を認識する DMTMPRA の構造を反映した**多全称閾値平均利得 Rabin ゲーム** (multi-universal-threshold mean-payoff Rabin Game, MUTMPRG) の勝利戦略問題に帰着することができる. そして、本研究では有限メモリ MDP 環境を想定した平均ケース実現可能性判定/最適合成を、そのゲームと有限メモリ MDP 環境の同期合成ゲーム上の、環境に対応するメモリレス敵戦略に対する対抗戦略/最適対抗戦略問題に帰着して解く手法を与える.

本章の構成

まず、9.1 節で準備として実現可能性判定のための LTL^{mp} フラグメント LTL^{mp-} を与え、LTL^{mp-} 平均ケース実現可能性と最適合成問題を改めて定義する. 次に、9.2 節では、LTL^{mp-} と等価なオートマトンの構造及び受理条件を反映できるように、勝利条件が Rabin 受理条件と MTMP 受理条件で定義されたゲーム MUTMPRG を与える. 9.3 節では、LTL^{mp-} 平均ケース実現可能性判定/最適合成問題が、その MUTMPRG 上の対抗戦略及び最適対抗戦略合成問題へ帰着できることを示す. そして、9.4 節で MUTMPRG 上の最適対抗戦略合成手法について述べる. 最後に、9.5 節で MUTMPRG 最適対抗戦略合成及び LTL^{mp-} 平均ケース最適合成の計算量について述べる.

9.1 LTL^{mp} フラグメント LTL^{mp-}

本節では、以下の点を考慮して、平均ケース実現可能性判定のための LTL^{mp} フラグメント LTL^{mp-} を定義する.

- DMTMPRA で認識可能である.
- 実用的なりアクティブシステム仕様を十分に記述できる.
- 平均ケース実現可能性判定が決定可能である.
- 合成される実現プログラムは有限メモリであることが望ましい.

そこでまず、「LTL 式」と「単純平均利式の連言と選言」の連言に限定する. これは多くの場合, LTL リアクティブシステム仕様 φ は, 環境が満たすべき前提条件 ψ とシステムの必須機能 φ_i の箇条書き, すなわち, 連言 $\bigwedge_i \varphi_i$ の含意関係 $\psi \rightarrow \bigwedge \varphi_i$ によって与えられるためである. この前提条件 ψ は「環境は, ある要求イベント $p \in IAP$ をある時点以後, 常に成り立たせ続けることはない ($= \neg \Diamond \Box p$)」といった極端に非協力的な環境の選択を制限するための制約である. しかしながら, 本研究では具体的な環境を想定するため, 環境が満たすべき前提条件は考慮する必要はない. また, LTL が主にシステムの機能を規定できるのに対して, 平均利得式は主にシステムの性能を規定する. そして時間によって真偽が変化しないため, 時間演算子の部分式にすることは実用的ではないと考えている. ただし, 性能に関する制約には多くの場合トレードオフがあるため, 平均利得式の選言は用いてよいことにする.

次に, 平均利得式は全称平均利得式に限定する. 補題 7.9 から分かるように, 存在平均利得式を充足する語を生成するためですら無限メモリを要求されうる. もし有限メモリで充足する語 σ を生成できるのであれば, その平均利得トレースはすべての閾値条件を常に満たす集積点に収束しているはずである.

最後に, どの利得項も時間演算子を含む特性変数をもたないものとする. until 演算子 U を含む特性変数はしばしばオートマトンの決定化不能性の要因になるため使用を制限する. 一方で next 演算子 X は一般に決定化不能性の原因にはならないが, 利得項部分の構文を制限しても同様の制約は記述できる. 例えば, 二つの next 演算子を含む式 φ の特性変数の平均利得項 $MP(1_\varphi)$ は, $\varphi \leftrightarrow XXp$ である補助命題 p の特性変数の平均利得項 $MP(1_p)$ と一致する. そのため制限しても実用上の記述力は同じである.

以上のような理由から, LTL^{mp-} を以下のように定義する.

定義 9.1 (LTL^{mp-} の構文) AP を原子命題の集合とする. このとき, AP 上

の LTL^{mp}- 式 φ は以下のように帰納的に定義される.

$$\begin{aligned}\varphi &::= \psi \wedge \Theta, \\ \Theta &::= \theta \mid \Theta \vee \Theta \mid \Theta \wedge \Theta, \\ \theta &::= \text{MP}(t) \triangleright 0, \\ t &::= c \mid \mathbf{1}_p \mid t + t \mid -t \mid t \cdot t.\end{aligned}$$

ただし, $p \in AP$, $\psi \in LTL$, $c \in \mathbb{R}$, $\triangleright \in \{>, \geq\}$ とする. ■

以下では, 平均利得式部分 ψ は積和標準形であると仮定する.

ここで, 8.1.1 項の MTMPBA 構成法と同様の手法により, LTL^{mp}- から等価な DMTMPRA を構成できる.

補題 9.2 (LTL^{mp}- から DMTMPRA への変換) LTL^{mp}- 式 $\varphi = \psi \wedge \Theta$ (Θ は平均利得式の積和標準形) に対して, $\mathcal{O}(2^{|\psi|})$ 個の状態, Θ 中の平均利得式の数と同じ次元数の重み関数, Θ の積和結合構造をそのまま保存した平均利得受理条件をもつ, $L(\varphi)$ を認識する DMTMPRA を合成できる. ■

証明. LTL 式 ψ を認識する DRA $\mathcal{A}_\psi$ は Safra の構成法 (定理 3.60[73]) により合成できる. これは最悪で ψ のサイズに対して二重指数オーダーの状態をもつ. ここで, Θ は

- n 個の原子平均利得式 $\theta_m = \text{MP}(t_m) \triangleright_m 0$ の連言節からなり,
- $\Theta = \bigvee_i \bigwedge_{\theta_m \in C_i} \theta_m$ という形をもつ

と仮定する. ただし, 各 C_i について $C_i \subseteq \{\theta_1, \dots, \theta_n\}$ とする.

このとき, $L(\psi)$ を認識するオートマトンは, n 次元 1 状態 DMTMPRA $\mathcal{A}_\Theta = \langle \{q_0\}, 2^{AP}, \{q_0\} \times 2^{AP} \times \{q_0\}, q_0, \langle F, \langle W, G \rangle \rangle \rangle$ として構成できる. ただし,

- $F = \{\emptyset, \{q_0\}\}$,
- 任意の遷移 $\langle q_0, a, q_0 \rangle$ に対して, $W[m](\langle q_0, a, q_0 \rangle) = \llbracket t_m \rrbracket_{a^\omega}$,
- $G = \bigcup_i \{\langle H_i^>, H_i^\geq, \emptyset, \emptyset \rangle\}$.
 - $H_i^> = \{m \mid \theta_m \in C_i \wedge \triangleright_m \text{は } >\}$,
 - $H_i^\geq = \{m \mid \theta_m \in C_i \wedge \triangleright_m \text{は } \geq\}$.

そして, $\mathcal{A}_\psi$ と $\mathcal{A}_\Theta$ がどちらも決定的であるため, 7.2 節で述べた積合成法を用いることで DMTMPRA $\mathcal{A}_\varphi$ を合成できる. その際, $\mathcal{A}_\Theta$ が 1 状態しかもたないため, $\mathcal{A}_\varphi$ の状態数は $\mathcal{A}_\psi$ と同じである. $\square$

注意 9.3 (LTL^{mp-} vs. LTL_{MP}) LTL^{mp-} は平均利得式の選言も用いてよい, [18] (注意 6.7) で定義されている LTL_{MP} より表現力は高いフラグメントである. $\blacksquare$

ここで, LTL^{mp-} 平均ケース実現可能性判定/最適合成問題を改めて定義する.

定義 9.4 (LTL^{mp-} 平均ケース実現可能性判定/最適合成問題) φ を LTL^{mp-} 仕様, $\mathcal{E}$ を MDP 環境, $\text{MP}(t)$ を LTL^{mp-} 目的関数とする.

このとき, $\mathcal{E}$ の下で φ が平均ケース実現可能か否か判定する問題を LTL^{mp-} 平均ケース実現可能性判定問題と呼ぶ.

また, $\mathcal{E}$ の下で φ が

- 実現不能ならば, 「実現不能」,
- 実現可能だが有限メモリ実現不能ならば, 「有限メモリ実現不能」,
- 有限メモリ実現可能ならば, 「目的関数 $\llbracket \text{MP}(t) \rrbracket_{\text{word}(\cdot)}$ に関して (ε) 最適な MDP RS $\mathcal{R} \in \text{Real}_{\text{fin}}^{\mathcal{E}, \varphi}$ 」

を出力する問題を LTL^{mp-} 最適合成問題という. $\blacksquare$

なお, 以下のように定義される, 最適化の対象となる値 $\text{Pfm}_{\mathcal{R}}^{\mathcal{E}}(\text{MP}(t))$ を $\mathcal{E}$ のもとでの $\mathcal{R}$ の $\text{MP}(t)$ に関する**性能値** (performance value) という.

$$\text{Pfm}_{\mathcal{R}}^{\mathcal{E}}(\text{MP}(t)) = \mathbf{E}_{\text{Pr}^{\mathcal{D}, \mathcal{R}, \mathcal{E}}} [\llbracket \text{MP}(t) \rrbracket_{\text{word}(\cdot)}]. \quad (9.1)$$

そして, $\mathcal{E}$ の下での Φ を実現する RS のふるまい上の $\text{MP}(t)$ に関する **(ε -) 最適性能値** $\text{SupPfm}_{(\text{fin})}^{\mathcal{E}, \Phi}(\text{MP}(t))$ は, 以下のように定義できる.

$$\text{SupPfm}_{(\text{fin})}^{\mathcal{E}, \Phi}(\text{MP}(t)) (+\varepsilon) = \sup_{\mathcal{R}' \in \text{Real}_{(\text{fin})}^{\mathcal{E}, \Phi}} \{\text{Pfm}_{\mathcal{R}'}^{\mathcal{E}}(\text{MP}(t))\}. \quad (9.2)$$

9.2 多全称閾値平均利得 Rabin ゲーム

LTL 実現可能性判定では, LTL 式を認識する決定性 ω -オートマトンをもとに, その式の性質を反映した正規ゲームに変換して実現可能性判定に用いてい

た. LTL^{mp}- 実現可能性判定でも, LTL^{mp}- 仕様 φ を認識する DMTMPRA をもとに, φ の性質を反映したゲームに変換して実現可能性判定に用いている.

そのため, 7章の ω -正規オートマトンの平均利得拡張と同様に, 正規ゲームに多次元重み関数と多閾値平均利得勝利条件を導入したゲームを定義する. ただし, ここでは多閾値平均利得条件は全称閾値条件の積和に限定した, **多全称閾値平均利得Rabin ゲーム** (MUTMPRG) を扱う.

本節では, このゲームの定義と関連する概念を与える.

定義

このゲームは形式的には以下のように定義される.

定義 9.5 (MUTMPRG) d 次元 MUTMPRG $\mathcal{G} = \langle Q, \Sigma, \Delta, q_{init}, \text{Win} \rangle$ は勝利条件パラメータ $\text{Win} = \langle F, \langle W, G \rangle \rangle$ が以下のように与えられているゲームである. ただし,

- F は Rabin 対 $\langle f, f' \rangle$ の集合,
- $W : \Delta \rightarrow \mathbb{R}^d$ は遷移に対する d 次元重み関数,
- G はインデックス集合の四つ組 $\langle H^>, H^{\geq}, H^<, H^{\leq} \rangle$ のうち存在閾値条件 $H^<$ と $H^{\leq}$ が空集合 $\emptyset$ で与えられるものの集合からなる MTMP 受理条件であり,
- $\text{Rabin}_F(r) \wedge \text{MP}_{W,G}(r)$ を満たすプレイ $r \in \text{Plays}^{\mathcal{G}}$ ではプレイヤー 0 が勝利する. ■

なお, G の要素である四つ組 $\langle H^>, H^{\geq}, \emptyset, \emptyset \rangle$ は, 以下では単に対 $\langle H^>, H^{\geq} \rangle$ で略記する.

注意 9.6 (既存の拡張ゲーム [34] との関係) [34] で扱っているゲームは, この MUTMPRG の多閾値受理条件 G を連言のみに制限したゲームである. [18] らは LTL^{mp}- のフラグメントについて, [34] のゲームの最適戦略問題に帰着する最悪ケース実現可能性判定/定量合成手法を提案している. また, Acacia+ を拡張しその判定/合成手法を実装している. ただし, 複数の平均利得制約が与えられた場合には有効な最適化手法は提案されていない. ■

目的関数

RS の性能が $\text{LTL}^{\text{mp-}}$ 目的関数 $\text{MP}(t)$ が解釈 $[[\text{MP}(t)]]$ の期待値で評価されることと同様に、対抗戦略の性能は、「ある重み関数 $W_{pfm} : \Delta \rightarrow \mathbb{R}$ に関する平均利得トレースの下極限」の期待値を通して評価することができる。本研究では、 d 次元 MUTMPRG は、最も大きいインデックスの次元の重み $W[d]$ を通してその性能を評価するものとする。そのため、RS の性能と同様に、対抗戦略の性能は平均利得 $\text{mp}_{w[d]}(\cdot)$ の期待値として定義される。つまり、以下のよう定義される値 $\text{Pfm}_{\text{Str}_0}^{\mathcal{G}, \text{Str}_1}$ を、 Str_1 に対する Str_0 の性能値という。

$$\text{Pfm}_{\text{Str}_0}^{\mathcal{G}, \text{Str}_1} = \mathbf{E}_{\text{Pr}_1^{\mathcal{G}, \text{Str}_0, \text{Str}_1}}[\text{mp}_{w[d]}(\cdot)]. \quad (9.3)$$

また、式 (9.2) と同様に、プレイヤー 1 の戦略 Str_1 に対する（有限メモリ/メモリレス）対抗戦略の性能上限、すなわち、 (ε) 最適性能値 $\text{SupPfm}_{(\text{fin/ml})}^{\mathcal{G}, \text{Str}_1}$ は以下のように定義できる。

$$\text{SupPfm}_{(\text{fin/ml})}^{\mathcal{G}, \text{Str}_1} (+\varepsilon) = \sup_{\text{Str}'_0 \in \text{Cnt}_{(\text{fin/ml})}^{\mathcal{G}, \text{Str}_1}} \{\text{Pfm}_{\text{Str}'_0}^{\mathcal{G}, \text{Str}_1}\}. \quad (9.4)$$

9.3 ゲーム上の最適対抗戦略問題への帰着

本節では、 $\text{LTL}^{\text{mp-}}$ 最適合成問題を MUTMPRG 最適対抗戦略合成問題に帰着できることを示す。

$\varphi = \psi \wedge \Theta$ を $\text{LTL}^{\text{mp-}}$ 仕様、 $\text{MP}(t)$ を平均利得目的関数、 $\mathcal{E} = \langle S, 2^{OAP}, 2^{IAP}, P, s_0 \rangle$ を MDP 環境、 $\mathcal{A}_\varphi = \langle Q, 2^{AP}, \hookrightarrow, q_0, \langle F, \langle W, G \rangle \rangle \rangle$ を $\text{LTL}^{\text{mp-}}$ 式 ψ と等価な n 次元全域的 DMTMPRA とする。ただし、

ここで以下のような $n+1$ 次元 MUTMPRG $\mathcal{G}_{\varphi, \text{MP}(t)}^{\mathcal{E}} = \langle Q', 2^{AP}, \Delta, \langle q_0, s_0 \rangle, \langle F', \langle W', G \rangle \rangle \rangle$ を考える。

- $Q'_0 = Q \times S$,
- $Q'_1 = Q \times S \times 2^{OAP}$,
- $\Delta_0(\langle q, s \rangle, a) = \langle q, s, a \rangle$,
- $q \xrightarrow{a \cup b} q'$ と $P(s, a, b) = \langle p, s' \rangle$ に対して、 $\Delta'_1(\langle q, s, a \rangle, b) = \langle q', s' \rangle$,

- $W'[m](\langle q', A, \Delta'(q', A) \rangle) =$

$$\begin{cases} 2 \cdot W[m](\langle q, a \cup b, q \rangle) & \text{if } 1 \leq m \leq n, q' = \langle q, s, a \rangle \in Q_1 \text{ and } A = b \in 2^{IAP}, \\ 2 \cdot \llbracket t \rrbracket_{a \cup b} & \text{if } m = n + 1, q' = \langle q, s, a \rangle \in Q_1 \text{ and } A = b \in 2^{IAP}, \\ 0 & \text{otherwise,} \end{cases}$$
- $F' = \{ \langle f_i \times S, f'_i \times S \rangle \mid \langle f_i, f'_i \rangle \in F \},$

$\mathcal{G}_{\varphi, \text{MP}(t)}^{\mathcal{E}}$ と $\mathcal{A}_{\varphi}$ はどちらも全域的かつ決定的であり、また、 $\mathcal{E}$ も入力 $a \in 2^{OAP}$ に対して出力 $b \in 2^{IAP}$ を応答した際の遷移先の状態は一意に決まっている。そのため、 $\mathcal{G}_{\varphi, \text{MP}(t)}^{\mathcal{E}}$ の任意のプレイ

$$\begin{aligned} r = & \langle \langle q_0, s_0 \rangle, a_0, \langle q_0, s_0, a_0 \rangle \rangle \langle \langle q_0, s_0, a_0 \rangle, b_0, \langle q_1, s_1 \rangle \rangle \\ & \langle \langle q_1, s_1 \rangle, a_1, \langle q_1, s_1, a_1 \rangle \rangle \langle \langle q_1, s_1, a_1 \rangle, b_1, \langle q_2, s_2 \rangle \rangle \dots \end{aligned}$$

は、 $\mathcal{A}_{\varphi}$ の以下のような実行

$$r' = \langle q_0, a_0 \cup b_0, q_1 \rangle \langle q_1, a_1 \cup b_1, q_2 \rangle \dots$$

と一対一対応する。

そしてこのとき、以下の補題が成り立つ。

補題 9.7 MUTMPRG $\mathcal{G}_{\varphi, \text{MP}(t)}^{\mathcal{E}}$ のプレイ $r \in \text{Plays}_{\varphi, \text{MP}(t)}^{\mathcal{G}_{\varphi, \text{MP}(t)}^{\mathcal{E}}}$ と、全域的 DMTM-PRA $\mathcal{A}_{\varphi}$ のそれに対応する実行 $r' \in \text{Runs}^{\mathcal{A}_{\varphi}}$ に関して、以下の性質が成り立つ。

- (a). $\text{Rabin}_{F'}(r) \Leftrightarrow \text{word}(r') \models \psi,$
- (b). $\text{MP}_{W', G}(r) \Leftrightarrow \text{word}(r') \models \Theta,$
- (c). $\text{mp}_{W[n+1]}(r) = \llbracket \text{MP}(t) \rrbracket_{\text{word}(r')}.$ ■

証明. (a): F' の定義から $\text{Rabin}_{F'}(r) \Leftrightarrow \text{Rabin}_F(r')$ である。また、 $\mathcal{A}_{\varphi}$ の構成から明らかに $\text{Rabin}_F(r') \Leftrightarrow \text{word}(r') \models \psi$ 。よって成り立つ。

(b): G の定義から、 $1 \leq m \leq n$ である m について以下を満たすことを示せばよい。

$$\text{mp}_{W[m]}(r) = \llbracket \text{MP}(t_m) \rrbracket_{\text{word}(r')}. \quad (9.5)$$

$\mathcal{G}_{\varphi, \text{MP}(t)}^{\mathcal{E}}$ の定義から、 $\mathcal{G}_{\varphi, \text{MP}(t)}^{\mathcal{E}}$ の2ステップが $\mathcal{A}_{\varphi}$ の1ステップに対応している。そして、 W' の定義から、 $\mathcal{G}_{\varphi, \text{MP}(t)}^{\mathcal{E}}$ の2ステップのうちの前のステップは重み0、

後のステップの重みは $\mathcal{A}_\varphi$ の 1 ステップの利得の倍である。よって、式 (9.5) は明らかに成り立つ。

(c): (b) と同様。 $\square$

そして、 $\mathcal{G}_{\varphi, \text{MP}(t)}^\mathcal{E}$ 上では、環境 $\mathcal{E}$ は以下のようなプレイヤー 1 のメモリレス戦略 $\mu_1^\mathcal{E}$ と見なせる。

$$\mu_1^\mathcal{E}(\langle q, s, a \rangle, b) = p \quad \text{for } P(s, a, b) = \langle p, s' \rangle. \quad (9.6)$$

一方で、 $\mathcal{G}_{\varphi, \text{MP}(t)}^\mathcal{E}$ 上のプレイヤー 0 の戦略 $\text{Str}_0 = \langle M, \mu_0, m_{\text{init}} \rangle$ は次のような RS $\mathcal{R}^{\text{Str}_0}$ と見なせる。

$$\mathcal{R}^{\text{Str}_0} = \langle (Q'_1 \cup \{\langle q_0, s_0 \rangle\}) \times M, 2^{IAP}, 2^{OAP}, P', \langle \langle q_0, s_0 \rangle, m_{\text{init}} \rangle \rangle. \quad (9.7)$$

ただし、遷移確率関数 P' は以下の制約を満たす。

$$P'(\langle q', m \rangle, A, B) = \begin{cases} \langle p, \langle \Delta_0(q', B), m' \rangle \rangle & \text{if } q' = \langle q_0, s_0 \rangle, m = m_{\text{init}} \text{ and } \mu_0(q', m, B) = \langle p, m' \rangle, \\ \langle p, \langle \Delta_0(q'', B), m' \rangle \rangle & \text{if } q' \in Q_1, q'' = \Delta_1(q', A) \text{ and } \mu_0(q'', m, B) = \langle p, m' \rangle. \end{cases} \quad (9.8)$$

ここで、以下の補題が成り立つ。

補題 9.8 (対抗戦略から RS への変換) $\mathcal{E}$ を有限状態 MDP 環境、 φ を $\text{LTL}^{\text{MP-}}$ 仕様、 $\text{MP}(t)$ を $\text{LTL}^{\text{MP-}}$ 目的関数とする。

このとき、 Str_0 が MUTMPRG $\mathcal{G}_{\varphi, \text{MP}(t)}^\mathcal{E}$ 上のプレイヤー 1 のメモリレス戦略 $\mu_1^\mathcal{E}$ に対するプレイヤー 0 の (有限メモリ) 対抗戦略ならば、 $\mathcal{R}^{\text{Str}_0}$ は $\mathcal{E}$ の下で φ を実現する有限メモリ RS であり、 $\text{Pfm}_{\text{Str}_0}^{\mathcal{G}_{\varphi, \text{MP}(t)}^\mathcal{E}, \mu_1^\mathcal{E}} = \text{Pfm}_{\mathcal{R}^{\text{Str}_0}}^\mathcal{E}(\text{MP}(t))$ を満たす。 $\blacksquare$

証明. $\mathcal{G}_{\varphi, \text{MP}(t)}^\mathcal{E}$ 及び $\mathcal{R}^{\text{Str}_0}$ の構成より明らか。 $\square$

これより、以下の定理が成り立つ。

定理 9.9 ($\text{LTL}^{\text{MP-}}$ 最適合成から MUTMPRG 最適対抗戦略合成への帰着)

$\mathcal{E}$ を有限状態 MDP 環境、 φ を $\text{LTL}^{\text{MP-}}$ 仕様、 $\text{MP}(t)$ を $\text{LTL}^{\text{MP-}}$ 目的関数とする。

このとき、 $\mathcal{E}$ の下で φ を実現する (有限状態) MDP RS $\mathcal{R} \in \text{Real}_{(\text{fin})}^{\mathcal{E}, \varphi}$ が存在するならば、かつそのときに限り、MUTMPRG $\mathcal{G}_{\varphi, \text{MP}(t)}^\mathcal{E}$ 上のプレイヤー 1 のメモリレス戦略 $\mu_1^\mathcal{E}$ に対する (メモリレス) 対抗戦略 $\text{Str}_0 \in \text{Cnt}_{(\text{ml})}^{\mathcal{G}_{\varphi, \text{MP}(t)}^\mathcal{E}, \mu_1^\mathcal{E}}$ が存在する

また、その $\mathcal{R}$ が LTL^{mp}- 目的関数 $\text{MP}(t)$ に関して (ε -) 最適であるならば、かつそのときに限り、 μ_0 も (ε -) 最適である。 ■

証明. 補題 9.8 と補題 9.10, 補題 9.11 から成り立つ。 □

補題 9.10 (RS から対抗戦略への変換) $\mathcal{E}$ を有限状態 MDP 環境, φ を LTL^{mp}- 仕様, $\text{MP}(t)$ を LTL^{mp}- 目的関数とする。

このとき、 $\mathcal{E}$ の下で φ を実現する (有限状態) MDP RS $\mathcal{R}$ が存在するならば、MUTMPRG $\mathcal{G}_{\varphi, \text{MP}(t)}^\varepsilon$ 上のプレイヤー 1 のメモリレス戦略 μ_1^ε に対して、 $\text{Pfm}_{\mathcal{R}}^\varepsilon(\text{MP}(t)) = \text{Pfm}_{\text{Str}_0}^{\mathcal{G}_{\varphi, \text{MP}(t)}^\varepsilon, \mu_1^\varepsilon}$ となる (有限メモリ) 対抗戦略 $\text{Str}_0 \in \text{Cnt}_{(\text{fin})}^{\mathcal{G}, \mu_1^\varepsilon}$ が存在する。 ■

証明. 本稿では省略する。 □

補題 9.11 (有限メモリ対抗戦略のメモリレス性) 任意の MPRA $\mathcal{G}$ とその上のプレイヤー 1 のメモリレス戦略 μ_1 に対して有限メモリ対抗戦略 $\text{Str}_0 \in \text{Cnt}_{\text{fin}}^{\mathcal{G}, \mu_1}$ が存在するならば、 $\text{Pfm}_{\text{Str}_0}^{\mathcal{G}, \mu_1} = \text{Pfm}_{\mu_0}^{\mathcal{G}, \mu_1}$ となるメモリレス対抗戦略 $\mu_0 \in \text{Cnt}_{\text{ml}}^{\mathcal{G}, \mu_1}$ が存在する。 ■

証明. 本稿では省略する。 □

定理 9.9 より、LTL^{mp}- 平均ケース実現可能性判定問題及び最適合成問題は、MUTMPRG 上の対抗戦略存在判定及び最適対抗戦略合成問題に帰着できる。

9.4 MUTMPRG 最適対抗戦略の構成法

前節では、LTL^{mp}- 平均ケース実現可能性判定/最適合成問題を MUTMPRG 上の対抗戦略/最適対抗戦略合成問題に帰着できることを示した。ここで、LTL^{mp}- 平均ケース実現可能性判定 (MUTMPRG 上の対抗戦略合成) は LTL^{mp}- 平均ケース最適合成 (MUTMPRG 上の最適対抗戦略合成) の部分問題となるため、本節では、MUTMPRG 最適対抗戦略合成手法についてのみ述べる。

まず、9.4.1 項で MUTMPRG 最適対抗戦略合成に関する基本的なアイデア及び MUTMPRG 最適対抗戦略合成手続きの概略を示す。

そして、9.4.2 項で MPRG 最適対抗戦略合成手続きで用いる部分手続きについて説明し、9.4.3 項で具体的な MPRG 最適対抗戦略合成手続きを与える。

9.4.1 MUTMPRG 最適対抗戦略合成の概略

ゲーム $\mathcal{G} = \langle Q, \Sigma, \Delta, q_0, \langle F, \langle W, G \rangle \rangle \rangle$ においてプレイヤー 1 がメモリレス戦略 μ_1 に従うと仮定した場合、ゲームは MDP と見なすこともできる。このとき、ほとんどすべてのプレイ r について、 μ_1 で規定される確率的選択に関して閉じた、ある強連結な部分グラフ $\langle Q', \Delta' \rangle \subseteq \langle Q, \Delta \rangle$ の状態集合のみを無限にしばしば訪れる、すなわち、 $\text{Inf}(\text{path}(r)) = Q'$ となる。そのような確率的選択に関して閉じた強連結な部分グラフを**終端成分** (end component, EC) と呼ぶ。形式的には、これは以下のように定義される。

定義 9.12 (EC) $\mathcal{G} = \langle Q, \Sigma, \Delta, q_0, \langle F, \langle W, G \rangle \rangle \rangle$ を MUTMPRG, μ_1 を $\mathcal{G}$ 上のプレイヤー 1 のメモリレス戦略とする。

このとき、以下の条件を満たす非空かつ強連結な部分グラフ $\langle Q', \Delta' \rangle$ を、 $\mathcal{G}$ 上の μ_1 に対する EC という。

- $Q'_n \subseteq Q_n$,
- $\Delta'_n : Q'_n \times \Sigma_n \mapsto Q'_n \ (\subseteq \Delta_n)$,
- $Q'_0 = \bigcup_{q \in Q'_1} \{\Delta_1(q, b) \mid \mu_1(q, b) > 0\}$.

また、EC の中で極大なものを極大 EC (maximal EC, MEC) と呼び、すべての MEC の集合を $\text{MEC}^{\mathcal{G}, \mu_1}$ と書く。 ■

ほとんどすべてのプレイ r について、 $\text{Inf}(\text{path}(r))$ がある EC $E = \langle Q', \Delta' \rangle$ の状態集合 Q' と一致しているということは、 μ_1 に対する対抗戦略 Str_0 が

- (a). ある EC E 上で、 $\text{Rabin}_F(r)$, $\text{MP}_{W,G}(r)$ かつ $\text{Inf}(\text{path}(r)) = Q'$ とするための局所的戦略と
- (b). そのような局所的戦略をもつ EC へ誘導する戦略

の組み合わせにより構成されていることを意味している。

そこでまず、(a) の戦略について考える。この戦略は、別の言い方をすれば、初期状態を E の任意の状態 $q'_0 \in Q'_0$ に変更したゲーム $\mathcal{G}_{q'_0}$ 上の μ_1 に対して、以下の条件を満たす対抗戦略 Str_0^E である。

$$\Pr^{\mathcal{G}_{q'_0}, \text{Str}_0^E, \mu_1} \left(\{r \in \text{Plays}^{\mathcal{G}_{q'_0}} \mid \forall i. Q' = \bigcup_{j \in \mathbb{N}} \{r[2(i+j)]\}\} \right) = 1 \quad (9.9)$$

そのような戦略 Str_0^E を E 上の**局所対抗戦略** (local counter strategy) といい、また、 E のことを**対抗 EC** (counter EC, CEC) と呼ぶ。特に、 Str_0^E が有限メモリ戦略であるとき、 Str_0^E をメモリレス局所対抗戦略、 E をメモリレス CEC (memoryless CEC, MLCEC) という。ここで、 E 上の (ML)CEC 上の (メモリレス) 局所対抗戦略の集合を $\text{Cnt}_{(\text{ml})}^{\mathcal{G}, \mu_1, E}$ と書くこととする。また、そのような中で ε -最適戦略 Str_0^E を E 上の (メモリレス) **局所 ε -最適対抗戦略** といい、そのときの性能値 $\text{Pfm}^{\mathcal{G}, \text{Str}_0^E, \text{Str}_1}$ を (メモリレス) **局所 ε -最適性能値** という。

$$\text{Pfm}^{\mathcal{G}, \text{Str}_0^E, \text{Str}_1} + \varepsilon \geq \sup_{\text{Str}'_0 \in \text{Cnt}_{(\text{ml})}^{\mathcal{G}, \mu_1, E}} \{\text{Pfm}^{\mathcal{G}, \text{Str}'_0, \mu_1}\}. \quad (9.10)$$

また、CEC $\langle Q', \Delta' \rangle$ に関しては、その状態集合 Q' は少なくとも以下の条件を満たしている。

$$\exists \langle f_i, f'_i \rangle \in F. (Q' \cap f_i = \emptyset \wedge Q' \cap f'_i \neq \emptyset) \quad (9.11)$$

この条件 (9.11) を満たす EC を **Rabin EC** (REC), その中で極大のものを**極大 REC** (maximal REC, MREC) と呼ぶ。ゲームは状態集合が有限であり、また、EC が REC であるか否かはグラフ構造から判定できるため、ゲーム上の REC はすべて列挙することができる。そして、それぞれの REC についてメモリレス局所対抗戦略の存在判定及び最適化を行うことで、(a) の戦略を合成できる。

次に、(b) の戦略について考える。これは (a) の戦略をもつ MLCEC へ到達させることを目的とした戦略である。また、その中で最適な戦略は、到達させる MLCEC 上の最適性能値の期待値を最大とする戦略である。そのような到達戦略を合成する問題は MDP の最大到達確率問題と類似の問題であり、プレイヤー 1 の戦略がメモリレス戦略ならば容易に構成できる。

MUTMPRG 最適対抗戦略合成手続きの概略

MPRG 上でプレイヤー 1 が μ_1 に対する最適対抗戦略を合成する手続きの概略は以下の通りである。

- i. ゲームのグラフを MEC 分解する。
- ii. それぞれの MEC が CEC を含むか判定する。
- iii. 初期状態 q_0 から CEC を含む MEC への最大到達確率を計算する。

- もしその最大到達確率が1でなければ, 対抗戦略は存在しない.
- iv. それぞれの MEC が MLCEC を含むかを判定する.
 - 同時に, MLCEC の中で最大の局所最適性能値を与える MLCEC を探索し, そのメモリレス最適対抗戦略を計算する.
- v. 「それぞれの MLCEC 上のメモリレス局所最適対抗戦略」が与える性能値を考慮して, それぞれの状態で最大期待性能値を計算する.
 - ただし, MLCEC に含まれる状態への最大到達確率が1でない状態の最大期待性能値は $-\infty$ とする.
 - もし q_0 での最大期待性能値が $-\infty$ である (つまり, MLCEC への最大到達確率が1でない) ならば, メモリレス対抗戦略は存在しない.
- vi. 「それぞれの MLCEC 上のメモリレス局所最適対抗戦略」と「遷移先が最大期待性能値が最も高い状態となるようにアクションを選ぶ戦略」を組み合わせ, ゲーム全体の上のメモリレス最適対抗戦略を構成する.

9.4.2 MUTMPRG 最適対抗戦略合成の部分手続き

本項では, 前項でのべた MUTMPRG 最適対抗戦略合成の各ステップを計算する手続きについて説明する. これは 9.4.3 項で与える MUTMPRG 最適対抗戦略合成手続きで用いる主要な部分手続きであり, 「REC が (ML)CEC であるかを判定する手続き」や「最大到達確率を計算する手続き」などである.

なお, ゲーム上の通常の MEC 分解手続きは, MDP の MEC 分解手続き [31] と同様であり, 多項式時間で計算できる. そのため, 本稿では手続きの詳細は省略するが, 以下の手続きが利用できるものとする.

- $\text{DecompMEC}(\mathcal{G}, \langle Q', \Delta' \rangle, \mu_1)$
 - MUTMPRG $\mathcal{G}$ 上のプレイヤー1のメモリレス戦略 μ_1 に対する, 部分グラフ $\langle Q', \Delta' \rangle$ に含まれる MEC の集合を返す.

また, 本章で与える部分手続きの中では, 7.3 節で述べた MTMPBA の空判定手続きと同様に, しばしば線形制約充足可能性問題 (LCSP) や線形計画問題 (LPP) を用いて解く. そこで, 以下のような手続きも利用できるものとする.

- $\text{SolveLCSP}(\text{Cnst})$

- 線形制約 $\text{Cnst} : (\mathbb{R}_{\geq 0})^n \rightarrow \mathbb{B}$ を満たす n 次元非負実数解ベクトルを返す.
 - * ただし, Cnst が充足不能のとき, 零ベクトル (すべての要素が 0 のベクトル) を返す.
- $\text{SolveLPP}_{\text{opt}}(\text{Obj}, \text{Cnst})$
 - 線形制約 $\text{Cnst} : (\mathbb{R}_{\geq 0})^n \rightarrow \mathbb{B}$ に従う線形目的関数 $\text{Obj} : (\mathbb{R}_{\geq 0})^n \rightarrow \mathbb{R}$ の最適値とそれを与える n 次元非負実数解ベクトルの対を返す.
 - * ただし, Cnst が充足不能のとき, $-\infty$ と零ベクトルの対を返す.
 - * 最適値は, $\text{opt} = \max$ のときには最大値 (最大が定義できない場合は上限値), $\text{opt} = \min$ のときには最小値 (最小が定義できない場合は下限値) である.

なお, 目的関数は Obj は行ベクトルとして与えることとする. また, 本節で与える手続き中で SolveLCSP と $\text{SolveLPP}_{\text{opt}}$ の入力となるすべての線形制約 Cnst は, それらを満たす解ベクトルが零ベクトルではない. そのため, 零ベクトルが返されたならば, かつそのときに限り, 線形制約が充足不能である. そして本節の線形制約の記述には, 状態集合 Q の大きさ $|Q|$ などを行/列のサイズとする行列やベクトルを主に用いる. そのような場合, 行列やベクトルの行/列にはその集合のそれぞれの要素が対応しているものとする. また, その要素で行/列のインデックスを指定する. 例えば, $|Q|$ 次元 (Q は状態集合) のベクトル $\mathbf{x}$ の状態 q に対応する要素は $\mathbf{x}[q]$, $|Q|$ 行 $|J|$ 列 ($J \subseteq \{1, \dots, d\}$ はインデックスの集合) の行列 $\mathbf{A}$ の状態 q に対応する行のインデックス j が対応する列の要素は $\mathbf{A}[q, j]$ と表記する.

MREC 分解手続き DecompMREC

この手続きは, MUTMPRG $\mathcal{G}$ 上でプレイヤ 1 がメモリレス戦略 μ_1 に従うときに, 与えられた MEC に含まれる MREC の集合を計算する手続きである. (ML)CEC は必ず REC であるため, その中で極大なものと計算するために用いる. これは, 通常の MEC 分解を用いる以下のような手続きとして与えられる.

手続き 9.13 (MREC 分解手続き DecompMREC)

- 入力:
 - MUTMPRG $\mathcal{G} = \langle Q, 2^{AP}, \Delta, q_0, \langle F, \langle W, G \rangle \rangle \rangle$

- $\mathcal{G}$ 上のプレイヤー 1 のメモリレス戦略 μ_1
 - $\mathcal{G}$ 上の MEC $\langle Q', \Delta' \rangle \subseteq \langle Q, \Delta \rangle$
 - 出力:
 - $\langle Q', \Delta' \rangle$ 上の MREC の集合
- ```

1: $M := \emptyset;$ // MREC の集合
2: for all $\langle H, I \rangle \in F$ do // それぞれの Rabin 対について
3: $M' := \text{DecompMEC}(\mathcal{G}, \langle Q' \setminus H, \Delta' \rangle, \mu_1);$
 // 高々有限回しか通るべきでない状態を取り除いて MEC 分解を行う
4: for all $E = \langle Q'', \Delta'' \rangle \in M'$ such that
 $(\forall E' \in M. E \not\subseteq E' \text{ and } Q'' \cap I \neq \emptyset)$ then
 // 得られた EC が極大な REC であるとき
5: $M := (M \setminus \{E' \in M \mid E' \subseteq E\}) \cup E;$
 // M から E に含まれる EC を取り除いた後で E を加える
6: end for
7: end for
8: return $M;$

```

**補題 9.14 (手続き 9.13 の正当性)** 手続き 9.13 の結果得られる状態の集合  $M$  は, MUTMPRG  $\mathcal{G}$  の MEC  $\langle Q', \Delta' \rangle$  上の MREC の集合である. ■

**証明.** 通常の MEC 分解手続きの正当性及び RMEC の定義より明らか. □

### 最大到達確率計算手続き SolveReachProb

この手続きは, MUTMPRG  $\mathcal{G}$  上でプレイヤー 1 がメモリレス戦略  $\mu_1$  に従っているときに, それぞれの状態  $q \in Q$  に関して, 状態部分集合  $Q' \subseteq Q$  への**最大到達確率** (プレイヤー 0 が適切に遷移先を選ぶことで得られる到達確率の最大値) を計算する手続きである.  $q$  から  $Q'$  への最大到達確率  $p_q^{\mathcal{G}, \mu_1, Q'}$  は, 厳密には以下のように定義される.

$$p_q^{\mathcal{G}, \mu_1, Q'} = \max_{\text{Str}_0} \{ \Pr^{\mathcal{G}, \text{Str}_0, \mu_1}(\{r \in \text{Plays}^{\mathcal{G}} \mid \exists i. r[2i] \in Q'\}) \}. \quad (9.12)$$

この最大到達確率問題は MDP の最大到達確率問題と同値であり、以下のような LPP に帰着する手続きにより解くことができる。

**手続き 9.15 (最大到達確率計算手続き SolveReachProb)**

- 入力:
  - MUTMPRG  $\mathcal{G} = \langle Q, 2^{AP}, \Delta, q_0, \langle F, \langle W, G \rangle \rangle \rangle$
  - $\mathcal{G}$  上のプレイヤー 1 のメモリレス戦略  $\mu_1$
  - $\mathcal{G}$  の部分状態集合  $Q' \subseteq Q$
- 出力:
  - それぞれの状態から  $Q'$  への最大到達確率  $p_q^{\mathcal{G}, \mu_1, Q'}$  を与える  $|Q|$  次元ベクトル  $\mathbf{p}$

1: **return** SolveLPP<sub>min</sub>( $\mathbf{u}^T, \text{LC}_{\text{reach}}^{Q, \Delta, \mu_1, Q'}$ );

ここで,  $\text{LC}_{\text{reach}}^{Q, \Delta, \mu_1, Q'} : (\mathbb{R}_{\geq 0})^{|Q|} \rightarrow \mathbb{B}$  は以下のように定義される線形制約である。

$$\text{LC}_{\text{reach}}^{Q, \Delta, \mu_1, Q'}(\mathbf{p}) \Leftrightarrow (\mathbf{p} \geq \mathbf{b}_{\text{reach}}^{Q, Q'} \wedge \mathbf{A}_{\text{nc}}^{Q, \Delta_0} \mathbf{p} \geq \mathbf{0} \wedge \mathbf{A}_{\text{pc}}^{Q, \Delta_1, \mu_1} \mathbf{p} \geq \mathbf{0}) \quad (9.13)$$

ただし,  $\mathbf{b}_{\text{reach}}^{Q, Q'}$  は  $|Q|$  次元ベクトル,  $\mathbf{A}_{\text{nc}}^{Q, \Delta_0}$  は  $|\{(q, q') \in Q_0 \times Q_1 \mid \Delta_0(q, a) = q'\}| \times |Q|$  行列,  $\mathbf{A}_{\text{pc}}^{Q, \Delta_1, \mu_1}$  は  $|Q_1| \times |Q|$  行列であり, 以下のように定義される。

$$\mathbf{b}_{\text{reach}}^{Q, Q'}[q] = \begin{cases} 1 & \text{if } q \in Q', \\ 0 & \text{otherwise.} \end{cases} \quad (9.14)$$

$$\mathbf{A}_{\text{nc}}^{Q, \Delta_0}[\langle q, q' \rangle, q''] = \begin{cases} 1 & \text{if } q'' = q \in Q_0, \\ -1 & \text{if } q'' = q' \in Q_1, \\ 0 & \text{otherwise.} \end{cases} \quad (9.15)$$

$$\mathbf{A}_{\text{pc}}^{Q, \Delta_1, \mu_1}[q, q'] = \begin{cases} 1 & \text{if } q' = q \in Q_1, \\ -\sum_{\langle q, b, q' \rangle \in \Delta_1} \mu_1(q, b) & \text{if } q' \in Q_0, \\ 0 & \text{otherwise.} \end{cases} \quad (9.16)$$

■

直観的には、 $\mathbf{A}_{nc}^{Q,\Delta_0} \mathbf{p} \geq \mathbf{0}$  の  $\langle q, q' \rangle$  行に関する線形制約は、 $q \in Q_0$  から 1 回の遷移で到達できる状態  $q' \in Q_1$  について、「 $q$  から  $Q'$  への最大到達確率」が「 $q'$  から  $Q'$  への最大到達確率」以上であることを表している。これは、プレイヤー 0 が遷移先を自由に選択できることに由来している。一方で、 $\mathbf{A}_{pc}^{Q,\Delta_1,\mu_1} \mathbf{p} \geq \mathbf{0}$  の  $q$  行に関する線形制約は、「 $q \in Q_1$  から  $Q'$  への最大到達確率」が「プレイヤー 0 が  $Q'$  へ到達させるために最適の選択を行う場合の到達確率の期待値」であることを表している。これは、プレイヤー 1 が  $\mu_1$  で規定された確率的選択によって遷移先を決定していることに由来している。

**補題 9.16 (手続き 9.15 の正当性)** 手続き 9.15 が返す解ベクトル  $\mathbf{p}$  の各要素  $\mathbf{p}[q]$  は、 $q$  から  $Q'$  への最大到達確率  $p_q^{\mathcal{G},\mu_1,Q'}$  である。 ■

**証明.** MDP における最大到達確率の計算と同様であるため明らか。 □

### CEC 判定手続き SolveCEC

この手続きは、MUTMPRG  $\mathcal{G}$  上でプレイヤー 1 がメモリレス戦略  $\mu_1$  に従うときに、与えられた REC  $E = \langle Q', \Delta' \rangle$  が CEC であるかを判定する手続きである。具体的には、以下のような LCSP に帰着する手続きとして与えられる。

#### 手続き 9.17 (CEC 判定手続き SolveCEC)

- 入力:
  - MUTMPRG  $\mathcal{G} = \langle Q, 2^{AP}, \Delta, q_0, \langle F, \langle W, G \rangle \rangle \rangle$
  - $\mathcal{G}$  上のプレイヤー 1 のメモリレス戦略  $\mu_1$ ,
  - $\mathcal{G}$  上の EC  $E = \langle Q', \Delta' \rangle$ ,
- 出力:
  - $E$  上の局所対抗戦略を表現した  $|\Delta'|$  次元の非負実数ベクトル  $\mathbf{x}$   
 \* ただし、 $E$  が CEC でないときには零ベクトル  $\mathbf{0}$  を返す。

- 1: **for all**  $g = \langle J, K \rangle \in G$  **do**
- 2:    $\mathbf{x} := \text{SolveLCSP}(\text{LC}_{\text{CEC}}^{\mu_1, E, W, g});$
- 3:   **if**  $\mathbf{x} \neq \mathbf{0}$  **then**

```

4: return x;
5: end if
6: end for
7: return 0;

```

ここで,  $\text{LC}_{\text{CEC}}^{\mu_1, E, W, g} : (\mathbb{R}_{\geq 0})^{|\Delta'|} \rightarrow \mathbb{B}$  は, 7.3 節の MTMPBA の空判定手続きで用いた LCSP と同様に以下のように定義される線形制約である.

$$\text{LC}_{\text{CEC}}^{\mu_1, E, W, g}(\mathbf{x}) \Leftrightarrow (\mathbf{u}^T \mathbf{x} > 0 \wedge \mathbf{A}_{\text{cyc}}^E \mathbf{x} = \mathbf{0} \wedge \mathbf{A}_{\text{str}}^{E, \mu_1} \mathbf{x} = \mathbf{0} \wedge \mathbf{A}_{\text{mp}, \geq}^{E, W, J} \mathbf{x} \geq \mathbf{0} \wedge \mathbf{A}_{\text{mp}, >}^{E, W, K} \mathbf{x} > \mathbf{0}). \quad (9.17)$$

ただし,  $\mathbf{A}_{\text{cyc}}^E$  は  $|Q'| \times |\Delta'|$  行列,  $\mathbf{A}_{\text{str}}^{E, \mu_1}$  は  $|\Delta'_1| \times |\Delta'|$  行列,  $\mathbf{A}_{\text{mp}, \geq}^{E, W, J}$  は  $|J| \times |\Delta'|$  行列,  $\mathbf{A}_{\text{mp}, >}^{E, W, K}$  は  $|K| \times |\Delta'|$  行列であり, 以下のように定義される.

$$\mathbf{A}_{\text{cyc}}^E[q, \langle q', A, q'' \rangle] = \begin{cases} 1 & \text{if } q = q', \\ -1 & \text{if } q = q'', \\ 0 & \text{otherwise,} \end{cases} \quad (9.18)$$

$$\mathbf{A}_{\text{str}}^{E, \mu_1}[\langle q, b, q' \rangle, \langle q'', A, q''' \rangle] = \begin{cases} 1 & \text{if } q = q'' \wedge b = A, \\ -\mu_1(q, b) & \text{if } q = q''', \\ 0 & \text{otherwise,} \end{cases} \quad (9.19)$$

$$\mathbf{A}_{\text{mp}, >}^{E, W, J}[j, \langle q, A, q' \rangle] = W[j](q, A), \quad (9.20)$$

$$\mathbf{A}_{\text{mp}, >}^{E, W, K}[k, \langle q, A, q' \rangle] = W[k](q, A). \quad (9.21)$$

■

式 (9.17) の線形制約式は, 直観的には, 7.3 節の MTMPBA の空判定手続きで用いた LCSP に環境のふるまいに関する制約 (式 (9.19) の  $q = q''$  の場合) を加えたものである.  $\mathbf{A}_{\text{str}}^{E, \mu_1} \mathbf{x} = \mathbf{0}$  の  $\langle q, b, q' \rangle$  行に関する線形制約は, 「 $\mu_1$  に従うプレイヤ 1 が状態  $q \in Q'_1$  で  $b$  を選択する確率  $\mu_1(q, b)$ 」と「遷移  $\langle q, b, q' \rangle$  の出現数」と「 $q$  へ入る遷移の出現数」の比が一致していることを表す. これは, プレイヤ 1 が  $\mu_1$  に従うときに,  $\text{Inf}(\text{path}(r)) \subseteq Q'$  となるほとんどすべてのプレイ  $r$  が満たすべき性質である.

従って, 環境側プレイヤが  $\mu_1$  に従っているとき, 線形制約  $\text{LC}_{\text{CEC}}^{\mu_1, E, W, g}$  を満たす解ベクトル  $\mathbf{x}$  に遷移出現比が漸近的に近づくプレイ  $r$  を構成でき,  $r$  は MTMP 受理条件を満たす.

**補題 9.18 (手続き 9.17 の正当性)** 手続き 9.17 が返す解ベクトル  $\mathbf{x}$  が零ベクトルでないならば、かつそのときに限り、 $E$  は  $\mu_1$  に対する CEC である。 ■

**証明.**  $\Rightarrow$ :  $\mathbf{x} \neq \mathbf{0}$  のとき、線形制約  $\mathbf{u}^T \mathbf{x} > 0$ ,  $\mathbf{A}_{\text{cyc}}^E \mathbf{x} = \mathbf{0}$ ,  $\mathbf{A}_{\text{str}}^{E, \mu_1} \mathbf{x} = \mathbf{0}$  を満たすため、 $\mathcal{G}_q$  ( $q \in Q'$ ) 上で、ほとんどすべてのプレイ  $r$  について以下のような特徴を持つプレイや 0 の戦略が構成できる。

- $\rho$  上の遷移出現比が  $\mathbf{x}$  に収束し、かつ、
- $Q'$  に含まれるすべての状態を無限にしばしば訪れる ( $\text{Inf}(\text{path}(r)) = Q'$ )。
  - ただし、ある状態については出現頻度が 0 に収束するかもしれない。

そして、線形制約  $\mathbf{A}_{\text{mp}, \geq}^{E, W, J} \mathbf{x} \geq \mathbf{0}$  と  $\mathbf{A}_{\text{mp}, >}^{E, W, K} \mathbf{x} > \mathbf{0}$  も満たすために、ほとんどすべてのプレイ  $r$  は MTMP 受理条件を満たす。また、 $E$  は REC でもあることから、 $r$  は Rabin 条件も満たしている。よって、 $E$  は EC である。

$\Leftarrow$ :  $\mathbf{x} = \mathbf{0}$  のとき、線形制約  $\text{LC}_{\text{CEC}}^{\mu_1, E, W, g}$  を満たす解ベクトルが存在しない。そのため、プレイヤ 1 が  $\mu_1$  に従うとき、MTMP 受理条件を満たすようなプレイを生成する  $E$  上の局所対抗戦略は存在しない。 □

また、以下のような系も成り立つ。

**系 9.19**  $\mathcal{G}$  を MUTMPRG,  $\mu_1$  をプレイヤ 1 のメモリレス戦略,  $E = \langle Q', \Delta' \rangle$  を  $\mathcal{G}$  上の  $\mu_1$  に対する REC とする。

このとき、 $E' = \langle Q'', \Delta'' \rangle \subseteq E$  が CEC ならば  $E$  も CEC である。 ■

**証明.**  $E'$  が CEC ならば、線形制約  $\text{LC}_{\text{CEC}}^{\mu_1, E', W, g}(\mathbf{x}')$  を満たす  $|\Delta''|$  次元ベクトル  $\mathbf{x}'$  が存在する。ここで、以下のような  $|\Delta'|$  次元ベクトル  $\mathbf{x}$  を考える。

$$\mathbf{x}[\delta] = \begin{cases} \mathbf{x}'[\delta] & \text{if } \delta \in \Delta'', \\ 0 & \text{otherwise.} \end{cases} \quad (9.22)$$

このとき、明らかに  $\mathbf{x}$  は線形制約  $\text{LC}_{\text{CEC}}^{\mu_1, E, W, g}(\mathbf{x})$  を満たす。従って、 $E$  も CEC である。 □

### メモリレス局所最適戦略探索手続き SolveOptLocStr

この手続きは、MUTMPRG  $\mathcal{G}$  上でプレイヤ 1 がメモリレス戦略  $\mu_1$  に従うときに、与えられた REC  $E = \langle Q', \Delta' \rangle$  に含まれる MLCEC の中で最も高い局所性能上限もつ MLCEC を探索し、そのメモリレス局所 ( $\varepsilon$ -) 最適性能値とそ

れを与える対抗戦略の対を計算する再帰的手続きである。ただし、この手続きでは打ち切り性能閾値を設定し、局所性能上限がその値以下であるときには再帰を停止する。この手続きの概略は以下の通りである。

- i.  $E$  が CEC であるかを判定する。
  - 同時に、 $E$  上の局所最適性能値  $v$  とその最適対抗戦略  $y$  も計算する。
  - その局所最適性能値が打ち切り性能閾値未満ならば計算を打ち切る。
    - \* 系 9.19 より、 $E$  の局所最適性能値は  $E$  の部分 CEC  $E' \subseteq E$  の局所最適性能値以上であるためである。
- ii.  $E$  が MLCEC であるかを判定する。
  - 同時に、 $E$  上のメモリレス局所対抗戦略  $x$  も計算する。
- iii. もし  $E$  が MLCEC ならば、局所最適性能値  $v$  を与えるメモリレス局所対抗戦略  $z$  があるかを判定する。
  - メモリレス局所対抗戦略が存在するならば、それを  $E$  上のメモリレス局所最適対抗戦略とする。
  - そうでないならば、 $x$  と  $y$  からメモリレス局所  $\varepsilon$ -最適対抗戦略を構成する。
- iv.  $E$  のそれぞれの子 REC について、メモリレス局所最適対抗戦略探索手続き  $\text{SolveOptLocStr}$  を再帰的に計算する。
- v. 子 REC に対する手続き  $\text{SolveOptLocStr}$  の計算結果を用いて、 $E$  とその部分 REC の中で最も高いメモリレス局所 ( $\varepsilon$ -) 最適性能値を持つ MLCEC を同定する。
  - そして、そのメモリレス局所 ( $\varepsilon$ -) 最適性能値とそれを与えるメモリレス局所 ( $\varepsilon$ -) 最適対抗戦略の対を返す。

ここで、 $E$  のそれぞれの子 REC とは、 $E$  より真に小さい REC の中で極大は REC を指す。その集合  $\text{ChildREC}^{G, \mu_1}(E)$  は、以下のように定義できる。

$$\text{ChildREC}^{G, \mu_1}(E) = \{\text{REC } E' \subset E \mid \forall \text{REC } E'' \subset E. E' \not\subseteq E''\}. \quad (9.23)$$

本節ではまず、メモリレス局所最適対抗戦略探索手続き  $\text{SolveOptLocStr}$  で用いる部分手続きを説明し、その後手続き  $\text{SolveOptLocStr}$  の詳細を示す。

**MLCEC 判定手続き SolveMLCEC** この手続きは, MUTMPRG  $\mathcal{G}$  でプレイヤー 1 がメモリレス戦略  $\mu_1$  に従うときに, 与えられた REC  $E = \langle Q', \Delta' \rangle$  が MLCEC であるかを判定する手続きである. CEC 判定手続き 9.17 と同様に, 以下のような LCSP に帰着する手続きとして与えられる.

**手続き 9.20 (MLCEC 判定手続き SolveMLCEC)**

- 入力:
    - MUTMPRG  $\mathcal{G} = \langle Q, 2^{AP}, \Delta, q_0, \langle F, \langle W, G \rangle \rangle \rangle$
    - $\mathcal{G}$  上のプレイヤー 1 のメモリレス戦略  $\mu_1$ ,
    - $\mathcal{G}$  上の EC  $E = \langle Q', \Delta' \rangle$ ,
  - 出力:
    - メモリレス局所対抗戦略を表現した  $|\Delta'|$  次元の非負実数ベクトル  $\mathbf{x}$   
 \* ただし,  $E$  が MLCEC でないとき零ベクトル  $\mathbf{0}$  を返す.
- 1: **for all**  $g = \langle J, K \rangle \in G$  **do**
  - 2:    $\mathbf{x} := \text{SolveLCSP}(\text{LC}_{\text{MLCEC}}^{\mu_1, E, W, g});$
  - 3:   **if**  $\mathbf{x} \neq \mathbf{0}$  **then**
  - 4:     **return**  $\mathbf{x}$ ;
  - 5:   **end if**
  - 6: **end for**
  - 7: **return**  $\mathbf{0}$ ;

ここで,  $\text{LC}_{\text{MLCEC}}^{\mu_1, E, W, g} : (\mathbb{R}_{\geq 0})^{|\Delta'|} \rightarrow \mathbb{B}$  は以下のように定義される線形制約である.

$$\text{LC}_{\text{MLCEC}}^{\mu_1, E, W, g}(\mathbf{x}) \Leftrightarrow (\mathbf{x} > \mathbf{0} \wedge \mathbf{A}_{\text{cyc}}^E \mathbf{x} = \mathbf{0} \wedge \mathbf{A}_{\text{str}}^{E, \mu_1} \mathbf{x} = \mathbf{0} \wedge \mathbf{A}_{\text{mp}, \geq}^{E, W, J} \mathbf{x} \geq \mathbf{0} \wedge \mathbf{A}_{\text{mp}, >}^{E, W, K} \mathbf{x} > \mathbf{0}). \quad (9.24)$$

■

CEC 判定手続きの線形制約式 (9.17) とこの線形制約式 (9.24) の違いは、式 (9.17) の線形制約  $\mathbf{u}^T \mathbf{x} > 0$  が式 (9.24) では線形制約  $\mathbf{x} > \mathbf{0}$  となっている点である。既に述べたように、 $\mathbf{x}$  は遷移出現比を表すベクトルであり、 $\mathbf{x} > \mathbf{0}$  は「すべての遷移の出現数（出現比）が0より大きい」という制約を表す。両プレイヤーがメモリレス戦略に従うとき、状態や遷移の出現頻度はほとんどすべてのプレイ  $r$  で収束するため、 $\text{Inf}(\text{path}(r)) = Q'$  であるならば、かつそのときに限り、「すべての遷移の出現数（出現比）が0より大きい」。

**補題 9.21 (手続き 9.20 の正当性)** 手続き 9.20 が返す解ベクトル  $\mathbf{x}$  が零ベクトルでないならば、かつそのときに限り、 $E$  は  $\mu_1$  に対する MLCEC であり、 $\mathbf{x}$  からメモリレス局所対抗戦略を構成できる。 ■

**証明.**  $\Rightarrow$ :  $\mathbf{x} \neq \mathbf{0}$  のとき、任意の  $q \in Q'_0$  について、MPRG  $\mathcal{G}_q$  上でプレイヤー 1 が  $\mu_1$  に従い、プレイヤー 0 が以下のように与えられる戦略  $\mu_0^x$  に従う場合を考える。

$$\mu_0^x(q, a) = \frac{\mathbf{x}[\langle q, a, q' \rangle]}{\sum_{a' \in \Sigma_0} \mathbf{x}[\langle q, a', q'' \rangle]} \quad (9.25)$$

このとき、線形制約  $\mathbf{x} > \mathbf{0}$  が満たすため、ほとんどすべてのプレイ上で遷移出現比は  $\mathbf{x}$  に収束する。ここで、 $\mathbf{x}$  のすべての要素が0より大きい、かつ、 $E$  は REC であるため、ほとんどすべてのプレイは Rabin 受理条件を満たす。また、 $\mathbf{x}$  は他の線形制約も満たすため、ほとんどすべてのプレイは平均利得受理条件も満たす。よって、 $\mu_0^x$  はメモリレス局所対抗戦略であり、 $E$  は MLCEC である。

$\Leftarrow$ : 対偶を示す。 $\mathbf{x} = \mathbf{0}$  のとき、線形制約  $\text{LC}_{\text{MLCEC}}^{\mu_1, E, W, g}$  を満たす解ベクトルが存在しない。そのため、プレイヤー 1 が  $\mu_1$  に従うとき、 $E$  上のすべての遷移の出現比が0に収束せずに MTMP 受理条件を満たすようなプレイを生成する  $E$  上のメモリレス局所対抗戦略は存在しない。 □

**局所最適対抗戦略合成手続き SolveOptCEC** この手続きは、MUTMPRG  $\mathcal{A}$  上でプレイヤー 1 がメモリレス戦略  $\mu_1$  に従うときに、与えられた REC  $E = \langle Q', \Delta' \rangle$  上の（無限メモリ）局所最適性能値とそれを与える対抗戦略を計算する手続きである。CEC 判定手続きの線形制約式 (9.17) を満たすベクトル  $\mathbf{x}$  は、局所対抗戦略を表現しており、その  $\mathbf{x}$  は遷移出現比を表現している。そのため、最適化対象である性能値は、 $|\Delta'|$  次元非負実数ベクトル上では  $(\mathbf{c}_{\text{obj}}^{E, W})^T \mathbf{x} / (\mathbf{u}^T \mathbf{x})$

で表現できる. ただし,  $(\mathbf{c}_{\text{obj}}^{E,W})^T : (\mathbb{R}_{\geq 0})^{|\Delta'|} \rightarrow \mathbb{R}$  は以下のように定義された  $|\Delta'|$  次元の行ベクトルとして与えられる線形目的関数である.

$$(\mathbf{c}_{\text{obj}}^{E,W})^T[\langle q, A, q' \rangle] = W[d](q, A) \quad (9.26)$$

従って, 線形制約式 (9.17) に従う  $(\mathbf{c}_{\text{obj}}^{E,W})^T \mathbf{x} / (\mathbf{u}^T \mathbf{x})$  の最適値とそれを与える  $\mathbf{x}$  を求めればよい. これは, 以下のような LPP に帰着する手続きとして与えることができる.

#### 手続き 9.22 (局所最適対抗戦略合成手続き SolveOptCEC)

- 入力:
    - MUTMPRG  $\mathcal{G} = \langle Q, 2^{A^P}, \Delta, q_0, \langle F, \langle W, G \rangle \rangle \rangle$
    - $\mathcal{G}$  上のプレイヤー 1 のメモリレス戦略  $\mu_1$ ,
    - $\mathcal{G}$  上の EC  $E = \langle Q', \Delta' \rangle$ ,
  - 出力:
    - $E$  上の局所最適性能値  $v$  と局所最適対抗戦略を表現した  $|\Delta'|$  次元の非負実数ベクトル  $\mathbf{y}$  の対  $\langle v, \mathbf{y} \rangle$
    - \* ただし,  $E$  が CEC でないとき,  $\langle -\infty, \mathbf{0} \rangle$  を返す.
- 1:  $\langle v, \mathbf{y} \rangle := \langle -\infty, \mathbf{0} \rangle$ ;
  - 2: **for all**  $g = \langle J, K \rangle \in G$  **do**
  - 3:    $\langle v', \mathbf{y}' \rangle := \text{SolveLPP}_{\max}((\mathbf{c}_{\text{obj}}^{E,W})^T, \text{LC}_{\text{OptCEC}}^{\mu_1, E, W, g})$ ;
  - 4:   **if**  $v > v'$  **then**
  - 5:      $\langle v, \mathbf{y} \rangle := \langle v', \mathbf{y}' \rangle$ ;
  - 6:   **end if**
  - 7: **end for**
  - 8: **return**  $\langle v, \mathbf{y} \rangle$ ;

ここで,  $LC_{\text{OptCEC}}^{\mu_1, E, W, g} : (\mathbb{R}_{\geq 0})^{|\Delta'|} \rightarrow \mathbb{B}$  は以下のように定義される線形制約である.

$$LC_{\text{OptCEC}}^{\mu_1, E, W, g}(\mathbf{y}) \Leftrightarrow (\mathbf{u}^T \mathbf{y} = 1 \wedge \mathbf{A}_{\text{cyc}}^E \mathbf{y} = \mathbf{0} \wedge \mathbf{A}_{\text{str}}^{E, \mu_1} \mathbf{y} = \mathbf{0} \wedge \mathbf{A}_{\text{mp}, \geq}^{E, W, J} \mathbf{y} \geq \mathbf{0} \wedge \mathbf{A}_{\text{mp}, >}^{E, W, K} \mathbf{y} \geq \mathbf{0}). \quad (9.27)$$

■

この手続きで求める  $E$  上の局所最適性能値は, それぞれの  $g = \langle J, K \rangle \in G$  に対する以下のような線形分数計画問題 (LFPP) の最適解の中の最大値である.

$$\text{maximize}_{\mathbf{x}} : (\mathbf{c}_{\text{obj}}^{E, W})^T \mathbf{x} / (\mathbf{u}^T \mathbf{x}) \quad (9.28)$$

$$\text{subject to} : \begin{pmatrix} \mathbf{u}^T \mathbf{x} & \geq & 1, \\ \mathbf{A}_{\text{cyc}}^E \mathbf{x} & = & \mathbf{0}, \\ \mathbf{A}_{\text{str}}^{E, \mu_1} \mathbf{x} & = & \mathbf{0}, \\ \mathbf{A}_{\text{mp}, \geq}^{E, W, J} \mathbf{x} & \geq & \mathbf{0}, \\ \mathbf{A}_{\text{mp}, >}^{E, W, K} \mathbf{x} & \geq & \mathbf{u} \end{pmatrix}. \quad (9.29)$$

8.5 節の LTL<sup>mp</sup> 最適化手法の注意 8.13 でも述べたが, この線形制約式 (9.29) と式 (9.17) で与えられる  $LC_{\text{CEC}}^{\mu_1, E, g}$  は実質的には同じ制約である. そしてこの線形分数計画問題から  $LC_{\text{OptCEC}}^{\mu_1, E, g}$  を制約とする線形計画問題が得られる.

**補題 9.23 (手続き 9.22 の正当性)** 手続き 9.22 が返す対  $\langle v, \mathbf{y} \rangle$  が  $\langle -\infty, \mathbf{0} \rangle$  でないとき, かつそのときに限り,  $E$  は CEC である. またそのとき,  $v$  は局所最適性能値であり,  $\mathbf{y}$  から局所最適対抗戦略を構成できる. ■

**証明.** (概略)  $\langle v, \mathbf{y} \rangle \neq \langle -\infty, \mathbf{0} \rangle$  ならば, かつそのときに限り,  $E$  が CEC かつ  $\mathbf{y}$  から対抗戦略が構成できることは, 補題 9.18 と同様に成り立つ. また,  $v$  が局所最適性能値であることは明らか. □

**メモリレス局所対抗戦略合成手続き SolveOptMLCEC** この手続きは, MUTM-PRG  $\mathcal{G}$  上でプレイヤ 1 がメモリレス戦略  $\mu_1$  に従うときに, 与えられた REC  $E = \langle Q', \Delta' \rangle$  上で与えられた性能値  $v$  をもつ有限メモリ対抗戦略を計算する手続きである. これは, 以下のような LCSP に帰着する手続きとして与えられる.

**手続き 9.24 (メモリレス局所対抗戦略合成手続き SolveOptMLCEC)**

- 入力:

- MUTMPRG  $\mathcal{G} = \langle Q, 2^{AP}, \Delta, q_0, \langle F, \langle W, G \rangle \rangle \rangle$
- $\mathcal{G}$  上のプレイヤー 1 のメモリレス戦略  $\mu_1$ ,
- $\mathcal{G}$  上の EC  $E = \langle Q', \Delta' \rangle$ ,
- 目標性能値  $v$

• 出力:

- 性能値  $v$  を与えるメモリレス局所対抗戦略を表す  $|\Delta'|$  次元の非負実数ベクトル  $\mathbf{z}$ 
  - \* ただし, そのような戦略が存在しないとき, 零ベクトル  $\mathbf{0}$  を返す.

```

1: for all $g = \langle J, K \rangle \in G$ do
2: $\mathbf{z} := \text{SolveLCSP}(\text{LC}_{\text{OptMLCEC}}^{\mu_1, E, g, v});$
3: if $\mathbf{z} \neq \mathbf{0}$ then
4: return $\mathbf{z}$;
5: end if
6: end for
7: return $\mathbf{0}$;

```

ここで,  $\text{LC}_{\text{OptMLCEC}}^{\mu_1, E, g, v} : (\mathbb{R}_{\geq 0})^{|\Delta'|} \rightarrow \mathbb{B}$  は以下のように定義される線形制約である.

$$\text{LC}_{\text{OptMLCEC}}^{\mu_1, E, g, v}(\mathbf{z}) \Leftrightarrow (\text{LC}_{\text{MLCEC}}^{\mu_1, E, g}(z) \wedge (\mathbf{c}_{\text{obj}}^{E, W} - v \cdot \mathbf{u})^T \mathbf{z} = 0.). \quad (9.30)$$

■

式 (9.30) で追加されている線形制約  $(\mathbf{c}_{\text{obj}}^{E, W} - v \cdot \mathbf{u})^T \mathbf{z} = 0$  は, 性能値が  $v$  であることを表現している. よって, 線形制約  $\text{LC}_{\text{OptMLCEC}}^{\mu_1, E, g, v}(\mathbf{z})$  を満たす  $\mathbf{z}$  は性能値  $v$  をもつメモリレス局所対抗戦略を表現している.

**補題 9.25 (手続き 9.24 の正当性)** 手続き 9.24 が返す解ベクトル  $\mathbf{z}$  が零ベクトルでないならば, かつそのときに限り,  $E$  は  $\mu_1$  に対する MLCEC であり,  $\mathbf{z}$  から性能値  $v$  をもつメモリレス局所対抗戦略を構成できる. ■

**証明.**  $\Rightarrow$ :  $\mathbf{z} \neq \mathbf{0}$  のとき, 線形制約  $LC_{\text{OptMLCEC}}^{\mu_1, E, g, v}(\mathbf{z})$  を満たすため, 式 (9.25) と同様に構成される戦略  $\mu_0^z$  は, 明らかにメモリレス局所対抗戦略である. また, プレイヤ 0 が  $\mu_0^z$  に従うとき, ほとんどすべてのプレイ上の遷移出現比が  $\mathbf{z}$  に収束するため, その性能値は  $(\mathbf{c}_{\text{obj}}^{E, W} \mathbf{z}) / (\mathbf{u}^T \mathbf{z})$  である. このとき, 線形制約  $(\mathbf{c}_{\text{obj}}^{E, W} - v \cdot \mathbf{u})^T \mathbf{z} = 0$  を満たすため, 明らかにその性能値は  $v$  と一致する.

$\Leftarrow$ : 対偶を示す.  $\mathbf{z} \neq \mathbf{0}$  のとき, 線形制約  $LC_{\text{OptMLCEC}}^{\mu_1, E, g, v}(\mathbf{z})$  と線形制約  $(\mathbf{c}_{\text{obj}}^{E, W} - v \cdot \mathbf{u})^T \mathbf{z} = 0$  のどちらも満たす解ベクトルは存在しない. つまり,  $E$  は MLCEC でない, または, 性能値  $v$  を与えるメモリレス局所対抗戦略は存在しない.  $\square$

**手続き SolveOptLocStr の詳細** これまで示した部分手続きを用いて, メモリレス局所最適戦略探索手続き SolveOptLocStr は具体的には以下のように与えられる.

#### 手続き 9.26 (メモリレス局所最適戦略探索手続き SolveOptLocStr)

• 入力:

- MUTMPRG  $\mathcal{G} = \langle Q, 2^{AP}, \Delta, q_0, \langle F, \langle W, G \rangle \rangle \rangle$
- $\mathcal{G}$  上のプレイヤ 1 のメモリレス戦略  $\mu_1$
- $\mathcal{G}$  上の REC  $E = \langle Q', \Delta' \rangle \subseteq \langle Q, \Delta \rangle$
- 打切性能閾値  $v_{\text{cutoff}}$

• 出力

- MLCEC  $E' \subseteq E$  の中で最も高い局所最適性能値もつ MLCEC 上の, メモリレス局所 ( $\varepsilon$ -) 最適性能値  $v$  とそのメモリレス局所 ( $\varepsilon$ -) 最適対抗戦略を表す  $|\Delta'|$  次元の非負実数ベクトル  $\mathbf{x}$  の対  $\langle v, \mathbf{x} \rangle$

\* ただし, MLCEC  $E' \subseteq E$  が存在しない, または, それらすべての局所最適性能値が  $v_{\text{cutoff}}$  未満のとき, 対  $\langle -\infty, \mathbf{x} \rangle$  を返す.

- 1:  $o = \langle v, \mathbf{x} \rangle := \langle -\infty, \mathbf{0} \rangle;$       // 局所 ( $\varepsilon$ -) 最適値とそれを与える戦略の組
- 2:  $\langle v_{\text{sup}}, \mathbf{y} \rangle := \text{SolveOptCEC}(\mathcal{G}, \mu_1, E);$   
     // REC  $E$  が CEC か判定し, 同時に局所最適性能値も計算する
- 3: **if**  $v_{\text{cutoff}} > v_{\text{sup}}$  **then**      // 局所最適性能値が打切閾値未満のとき
- 4:   **return**  $\langle -\infty, \mathbf{0} \rangle;$

```

5: end if

6: $\mathbf{x} := \text{SolveMLCEC}(\mathcal{G}, \mu_1, E);$ // メモリレス局所対抗戦略を計算

7: if $\mathbf{x} \neq \mathbf{0}$ then // メモリレス局所対抗戦略が存在するとき

8: $\mathbf{z} := \text{SolveOptMLCEC}(\mathcal{G}, \mu_1, E, v_{\text{sup}});$
 // 無限メモリ最適性能をもつメモリレス対抗戦略を計算する

9: if $\mathbf{z} \neq \mathbf{0}$ then // メモリレスのものが存在するとき

10: $o := \langle v_{\text{sup}}, \mathbf{z} \rangle;$ // 最適戦略を更新する

11: else if $\varepsilon > 0$ then
 // メモリレスのものが存在しない, かつ, 許容誤差が0でないとき

12: $\mathbf{x} := \mathbf{x} + (\tau_{\mathbf{x}, \mathbf{y}}^{v, \varepsilon})\mathbf{y};$ // メモリレス局所 ε -最適対抗戦略を合成する

13: $v := (\mathbf{c}_{\text{obj}}^{E, w})^T \mathbf{x} / (\mathbf{u}^T \mathbf{x});$ // 局所最適値を局所 ε -最適値に変更する

14: end if

15: for all $E' = \langle Q'', \Delta'' \rangle \in \text{ChildREC}^{\mathcal{G}, \mu_1}(E)$ do

16: $\langle v', \mathbf{x}' \rangle := \text{SolveOptLocStr}(\mathcal{G}, \mu_1, E', v_{\text{sup}});$
 // 子 REC のメモリレス局所最適 // 対抗戦略を再帰的に計算する

17: if $v' > v$ or $(v' = v \text{ and } \text{num}_{\neq 0}(\mathbf{x}') < \text{num}_{\neq 0}(\mathbf{x}))$ then
 // より性能が高い, または, 同性能でも遷移関係のサイズ
 // がより小さいとき, 局所最適値と局所最適戦略を更新する

18: $v := v';$

19: for all $\delta \in \Delta'$ do

20: if $\delta \in \Delta''$ then

21: $\mathbf{x}[\delta] := \mathbf{x}'[\delta];$

22: else

23: $\mathbf{x}[\delta] := 0;$

24: end if

```

```

25: end for
26: end if
27: end for
28: return α ;

```

ただし,  $\tau_{\mathbf{x}, \mathbf{y}}^{v, \varepsilon}$  は以下を満たす最小の非負実数とする.

$$\tau_{\mathbf{x}, \mathbf{y}}^{v, \varepsilon} \geq \frac{\left( (v - \varepsilon) \mathbf{u}^T - (\mathbf{c}_{\text{obj}}^{E, w})^T \right) \mathbf{x}}{\varepsilon}. \quad (9.31)$$

また,  $\text{num}_{\neq 0}(\mathbf{x})$  はベクトル  $\mathbf{x}$  の非零の要素数を表す. ■

系 9.19 より,  $E$  が CEC のとき, ステップ 2 において計算される  $E$  上の (無限メモリ) 局所最適性能値  $v$  は,  $E$  の部分 CEC の局所最適性能値以上である. そのため,  $v$  が打切閾値  $v_{\text{cutoff}}$  より小さい場合,  $E$  の部分 CEC の局所最適性能値が打切閾値  $v_{\text{cutoff}}$  以上となることはない.

$E$  が MLCEC であるとき,  $v$  と等しい局所性能値をもつメモリレス対抗戦略が存在するならば, その戦略は局所最適なメモリレス対抗戦略であり,  $E$  の部分 MLCEC の中で最大の最適性能値をもつ. ただし, 同性能の, より小さい (遷移数が少ない) MLCEC が存在する可能性がある. また,  $v$  と等しい性能値をもつメモリレス局所対抗戦略が存在しない場合でも, 局所  $\varepsilon$ -最適なメモリレス対抗戦略を構成できる. これは, ステップ 2 で得られる (無限メモリ) 局所最適対抗戦略を表す  $\mathbf{y}$  と, ステップ 6 で得られるメモリレス局所対抗戦略を表す  $\mathbf{x}$  から構成される. なぜならば, 線形制約  $\text{LC}_{\text{MLCEC}}^{\mu_1, E', W, g}(\mathbf{x})$  を満たす  $\mathbf{x}$  と線形制約  $\text{LC}_{\text{OptCEC}}^{\mu_1, E', W, g}(\mathbf{y})$  を満たす  $\mathbf{y}$  に関して, 任意の  $c > 0$  で  $\mathbf{x} + c\mathbf{y}$  が線形制約  $\text{LC}_{\text{MLCEC}}^{\mu_1, E', W, g}(\mathbf{x} + c\mathbf{y})$  を満たすためである. このとき,  $c = \tau_{\mathbf{x}, \mathbf{y}}^{v, \varepsilon}$  とすれば, その性能値は局所  $\varepsilon$ -最適となる.

そして, ステップ 15-27 ですべての部分 MLCEC の中で最大の局所最適性能値をもつ MLCEC を探索している.

よって, 以下の補題が成り立つ.

**補題 9.27 (手続き 9.26 の正当性)** 手続き 9.26 が返す対  $\langle v, \mathbf{x} \rangle$  が  $\langle -\infty, \mathbf{0} \rangle$  でないならば, かつそのときに限り,  $E$  の部分 REC の中に MLCEC であるものが存在し, それらの中で最大の局所最適性能値が  $v$ , かつ,  $\mathbf{x}$  から性能値  $v$  をもつ局所メモリレス対抗戦略を構成できる. ■

**証明.** MLCEC 判定手続き SolveMLCEC の正当性 (補題 9.21), 局所最適対抗戦略合成手続き SolveOptCEC の正当性 (補題 9.23), メモリレス対抗戦略合成手続き SolveOptMLCEC の正当性 (補題 9.25) より成り立つ.  $\square$

**注意 9.28 (手続き 9.26 の計算量)** 手続き 9.26 は最悪の場合, グラフ  $E = \langle Q', \Delta' \rangle$  上のすべて REC を網羅的に列挙する必要がある. グラフ  $E$  の部分グラフの個数は,  $E$  のサイズの指数オーダーであるため, 手続き 9.26 の時間計算量は  $E$  のサイズに関して指数オーダーである.

なお, この再帰的手続きでは, 同じ部分 REC に対する手続き SolveOptLocStr を複数回呼び出し得る. そうした場合に前の計算結果をそのまま返しても, 全体の手続きの正当性は保たれる.  $\blacksquare$

### 最大到達期待値計算手続き SolveReachExp

この手続きは, MUTMPRG  $\mathcal{G}$  上で

- 状態の部分集合  $Q'_i$  と
- プレイ上で  $Q'_i$  へ到達することの優先度  $v_i$

の組  $\langle Q'_i, v_i \rangle$  の集合  $L = \bigcup_i \langle Q'_i, v_i \rangle$  が与えられているときに, プレイヤ 1 のメモリレス戦略  $\mu_1$  に対して, それぞれの状態  $q$  での**最大到達期待値** (プレイヤ 0 が適切に遷移先を選ぶことで得られる優先度期待値の最大値) を計算する手続きである. これは最大到達確率計算手続き 9.15 と似ているが, 到達される状態に優先度が付けられている点が異なる. 直観的には,  $Q'_i$  は MEC の状態集合, 優先度  $v_i$  はその MEC に含まれる MLCEC の中で最大の局所最適性能値であることを意図している. つまり, 優先度期待値が高いプレイの生起させようとする戦略では, 局所最適性能値が高い MLCEC へ到達する期待値が高くなる. ここで, どの  $Q'_i$  にも含まれない状態の優先度は  $-\infty$  であるものとする. つまりプレイ  $r$  の優先度  $\nu^L(r)$  は以下のように定義する.

$$\nu^L(r) = \begin{cases} \max_{\langle Q'_i, v_i \rangle \in L} \{v_i \mid \exists j. \text{path}(r)[j] \in Q'_i\} & \text{if } \exists j. \rho[j] \in \bigcup_{\langle Q'_i, v_i \rangle \in L} Q'_i, \\ -\infty & \text{otherwise.} \end{cases} \quad (9.32)$$

このように定義するのは, プレイ中に MLCEC へ到達した場合でも, より性能の高い別の MLCEC へ到達させられる可能性があるためである. そして, そ

それぞれの状態  $q$  での最大到達期待値  $e_q^{\mathcal{G}, \mu_1, Loc}$  は、厳密には以下のように定義される。

$$e_q^{\mathcal{G}, \mu_1, L} = \max_{\text{Str}_0} \{ \mathbf{E}_{\text{Pr}^{\mathcal{G}, \mu_1, \text{Str}_0}} [\nu^L] \}. \quad (9.33)$$

そしてこれは、以下のような LPP に帰着する手続きとして与えられる。

### 手続き 9.29 (最大到達期待値計算手続き SolveReachExp)

- 入力

- MUTMPRG  $\mathcal{G} = \langle Q, 2^{AP}, \Delta, q_0, \langle F, \langle W, G \rangle \rangle \rangle$
- $L = \bigcup_i \langle Q'_i, v_i \rangle$ 
  - \*  $Q'_i \subseteq Q$  は状態集合ただし、
    - ・それぞれの  $Q'_i$  は互いに共通部分をもたないと仮定する。
  - \*  $v_i \in \mathbb{R}$  は  $Q'_i$  に含まれる状態の優先度

- 出力

- それぞれの状態の最大到達期待値  $e_q^{\mathcal{G}, \mu_1, L}$  を与える  $|Q|$  次元のベクトル  $\mathbf{e}$

```

1: $\mathbf{p} := \text{SolveReachProb}(\mathcal{G}, \mu_1, \bigcup_{\langle Q', v_i \rangle \in L} Q')$;
2: $\mathbf{e} := \text{SolveLPP}_{\min}((\mathbf{c}_{\text{reach}}^{\mathbf{p}})^T, \text{LC}_{\text{exp}}^{\mathcal{G}, \mu_1, Loc})$;
3: for all $q \in Q$ such that $\mathbf{p}[q] < 1$ do
4: $\mathbf{e}[q] := -\infty$;
5: end for
6: return $\mathbf{e}$;

```

ここで、線形目的関数  $(\mathbf{c}_{\text{reach}}^{\mathbf{p}})^T$  は、以下のように定義される  $|Q|$  次元の行ベクトルである。

$$\mathbf{c}_{\text{reach}}^{\mathbf{p}}[q] = \begin{cases} 1 & \text{if } \mathbf{p}[q] = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (9.34)$$

また、線形制約  $LC_{\text{exp}}^{\mathcal{G}, \mu_1, \mathbf{p}, L} : (\mathbb{R}_{\geq 0})^{|Q|} \rightarrow \mathbb{B}$  は以下のように定義される線形制約である。

$$LC_{\text{exp}}^{\mathcal{G}, \mu_1, L}(\mathbf{e}) \Leftrightarrow (\mathbf{A}_{\text{exp}}^{Q, L} \mathbf{e} \geq \mathbf{b}_{\text{exp}}^{Q, L} \wedge \mathbf{A}_{\text{nc}}^{Q, \Delta_0} \mathbf{e} \geq \mathbf{0} \wedge \mathbf{A}_{\text{pc}}^{Q, \Delta_1, \mu_1} \mathbf{e} \geq \mathbf{0}). \quad (9.35)$$

ただし、 $\mathbf{A}_{\text{exp}}^{Q, L}$  は  $|Q| \times |Q|$  行列、 $\mathbf{b}_{\text{exp}}^{Q, L}$  は  $|Q|$  次元の実数ベクトルであり、以下のように定義される。

$$\mathbf{A}_{\text{exp}}^{Q, L}[q, q'] = \begin{cases} 1 & \text{if } \exists \langle Q', v \rangle \in L. q \in Q' \\ & \text{and } q = q', \\ 0 & \text{otherwise.} \end{cases} \quad (9.36)$$

$$\mathbf{b}_{\text{exp}}^{Q, L}[q] = \begin{cases} v & \text{if } \exists \langle Q', v \rangle \in L. q \in Q', \\ 0 & \text{otherwise.} \end{cases} \quad (9.37)$$

■

**補題 9.30 (手続き 9.29 の正当性)** 手続き 9.29 が返す解ベクトル  $\mathbf{e}$  の各要素  $\mathbf{e}[q]$  は、 $q$  での最大到達期待値  $e_q^{\mathcal{G}, \mu_1, L}$  である。 ■

**証明.** 補題 9.16 と同様に MDP の最大期待値問題と同様であるため正当性は明らか。ただし、 $\mathbf{p}[q] < 1$  である  $q$  に関しては、明らかに  $\mathbf{e}[q] = -\infty$  となるが、線形制約  $LC_{\text{exp}}^{\mathcal{G}, \mu_1, \mathbf{p}, L}$  ではそれに関する制約を与えていない。そのため、ステップ 3-5 で代入している。 □

### 9.4.3 MUTMPRG 最適対抗戦略合成手続き

本章では、MUTMPRG においてプレイヤー 1 のメモリレス戦略に対するメモリレス最適対抗戦略を合成する手続きを与える。

これは、9.4.2 項で与えた手続きを部分手続きとして用い、9.4.1 項の概略に沿った以下の手続きである。

#### 手続き 9.31 (MUTMPRG 最適対抗戦略合成手続き)

- 入力:
  - MUTMPRG  $\mathcal{G} = \langle Q, 2^{AP}, \Delta, q_0, \langle F, \langle W, G \rangle \rangle \rangle$
  - $\mathcal{G}$  上のプレイヤー 1 の戦略  $\mu_1$

– 許容誤差  $\varepsilon \geq 0$

• 出力:

$$- \left\{ \begin{array}{l} \text{“対抗戦略は存在しない”} \\ \quad \text{if } |\text{Cnt}^{\mathcal{G}, \mu_1}| = 0, \\ \text{“メモリレス対抗戦略は存在しない”} \\ \quad \text{if } |\text{Cnt}^{\mathcal{G}, \mu_1}| \neq 0 \text{ and } |\text{Cnt}_{\text{ml}}^{\mathcal{G}, \mu_1}| = 0, \\ \text{メモリレス } (\varepsilon\text{-}) \text{ 最適対抗戦略 } \mu_0 \\ \text{otherwise.} \end{array} \right.$$

```

1: $M := \text{DecompMEC}(\mathcal{G}, \mu_1);$ // ゲーム上の MEC の集合を計算する
2: $N := \emptyset;$ // MEC とそこに含まれる極大な CEC の集合の対の集合
3: for all $E_i = \langle Q'_i, \Delta'_i \rangle \in M$ do
4: $M'_i := \text{DecompMREC}(\mathcal{G}, \mu_1, E_i);$ // MEC E 上の MREC の集合を計算する
5: $M''_i := \emptyset;$ // M'_i 中の CEC の集合
6: for all $E'_{ij} \in M'_i$ do
7: $\mathbf{x} := \text{SolveCEC}(\mathcal{G}, \mu_1, E'_{ij});$ // MREC E' 上の対抗戦略を計算する
8: if $\mathbf{x} \neq \mathbf{0}$ then
9: // E' 上に対抗戦略が存在する (E' が CEC である) とき
10: $M''_i := M''_i \cup \{E'_{ij}\};$
11: end if
12: end for
13: if $M''_i \neq \emptyset$ then
14: $N := N \cup \{\langle E_i, M''_i \rangle\};$
15: end if
16: end for

```

```

16: $Q_{\text{reach}} := \bigcup_{\langle Q'_i, \Delta'_i \rangle, M''_i \in N} Q'_i;$
 // CEC を含む MEC に含まれる状態の集合を計算する

17: $\mathbf{p} := \text{SolveReachProb}(\mathcal{G}, \mu_1, Q_{\text{reach}});$
 // 各状態の Q_{reach} への最大到達確率を計算する

18: if $\mathbf{p}[q_0] < 1$ then
 // 初期状態から Q_{reach} への最大到達確率が1でないとき

19: return “対抗戦略は存在しない”;

20: end if

21: $Loc := \emptyset;$ // “MEC” と “その中の最大のメモリレス局所最適性能値と
 // それを与えるメモリレス局所最適対抗戦略の対” の対の集合

22: for all $\langle E_i, M''_i \rangle \in N$ do

23: $o_i = \langle v_i, \mathbf{x}_i \rangle := \langle -\infty, \mathbf{0} \rangle;$ // メモリレス局所最適性能値とそれを与
 // えるメモリレス局所最適対抗戦略の対

24: for all $E'_{ij} \in M''_i$ do

25: $\langle v_{ij}, \mathbf{x}_{ij} \rangle := \text{SolveOptLocStr}(\mathcal{G}, \mu_1, E'_{ij}, v_i, \epsilon);$
 // 極大な CEC E'_{ij} に含まれる MLCEC の中で
 // 最大の局所最適性能を与えるものを探索する

26: if $v_{ij} > v_i$ or $(v_{ij} = v_i \text{ and } \text{num}_{\neq 0}(\mathbf{x}_{ij}) < \text{num}_{\neq 0}(\mathbf{x}_i))$ then
 // より性能が高い, または, 同性能でも
 // 遷移関係のサイズがより小さいとき

27: $o_i := \langle v_{ij}, \mathbf{x}_{ij} \rangle;$ // メモリレス最適局所対抗戦略を更新する

28: end if

29: end for

30: if $v_i > -\infty$ then // E' が MLCEC を含むとき

31: $Loc := Loc \cup \{ \langle E_i, o_i \rangle \};$

32: end if

```

```

33: end for
34: $L := \bigcup_{\langle \langle Q'_i, \Delta'_i \rangle, \langle v_i, \mathbf{x}_i \rangle \rangle \in Loc} \langle Q'_i, v_i \rangle;$
35: $\mathbf{e} := \text{SolveReachExp}(\mathcal{G}, \mu_1, L);$ // 各状態の最大期待性能値を計算する
36: if $\mathbf{e}[q_0] = -\infty$ then // 初期状態で期待される最適性能値が $-\infty$
// (メモリレス戦略では対抗できない) とき
37: return “メモリレス対抗戦略は存在しない”;
38: end if
39: $\mu_0 := \mu_0^{Loc, \mathbf{e}};$ // Loc と $\mathbf{e}$ からメモリレス最適対抗戦略を構成する
40: return $\mu_0;$

```

また、メモリレス戦略  $\mu_0^{Loc, \mathbf{e}} : Q_0 \times \Sigma_0 \rightarrow [0, 1]$  は以下のように与える.

$$\mu_0^{Loc, \mathbf{e}}(q, a) = \begin{cases} \frac{\mathbf{x}_i[\langle q, a, q' \rangle]}{\sum_{a' \in \Sigma_0} \mathbf{x}_i[\langle q, a', q'' \rangle]} & \text{if } \exists \langle E_i, \langle v_i, \mathbf{x}_i \rangle \rangle \in Loc. (\mathbf{e}[q] = v_i \text{ and } \mathbf{x}_i[\langle q, a, q' \rangle] > 0), \\ 0 & \text{if } \mathbf{e}[\Delta(q, a)] < \max_{a' \in \Sigma_0} \{\mathbf{e}[\Delta(q, a')]\}, \\ \frac{1}{|\{\langle q, a' \rangle | \mathbf{e}[\Delta(q, a')] > 0\}|} & \text{otherwise.} \end{cases} \quad (9.38)$$

■

直観的には、 $\mu_0^{Loc, \mathbf{e}}$  は、それぞれの状態  $q \in Q_0$  において、以下のような特徴を持つメモリレス戦略である.

- $q$  がある MEC  $E_i$  上でメモリレス局所最適性能値  $v_i$  を与える MLCEC に含まれており、かつ、 $q$  の最大期待性能値  $\mathbf{e}[q]$  が  $v_i$  と等しいとき、
  - $E_i$  上で性能値  $v_i$  を与える ( $\mathbf{x}_i$  で表現された) メモリレス局所最適対抗戦略に従う.
- そうでないとき、

- 最大期待性能値が最も高い状態を遷移先とするようなアクションを（ただし、複数存在するときにはそれらを等確率で）選択する。

**定理 9.32 (手続き 9.31 の正当性)** 手続き 9.31 が“(メモリレス) 対抗戦略は存在しない”を返すならば、かつそのときに限り、 $\mathcal{G}$  上の  $\mu_1$  に対する（対抗戦略は存在するがメモリレス）対抗戦略は存在しない。

また、手続き 9.31 がプレイヤ 0 のメモリレス戦略  $\mu_0$  を返すならば、かつそのときに限り、 $\mu_0$  は  $\mathcal{G}$  上の  $\mu_1$  に対するメモリレス最適対抗戦略である。 ■

**証明.** MREC 分解手続き DecompMREC の正当性（補題 9.14）と CEC 判定手続き SolveCEC の正当性（補題 9.18），系 9.19 より，ステップ 1-15 の結果として得られる  $N$  は、「CEC を含む MEC  $E_i$ 」と「それに含まれる極大な CEC の集合  $M_i''$ 」の対の集合となっている。そして、最大到達確率計算手続き SolveReachProb の正当性（補題 9.16）より、 $\mathbf{p}$  は、プレイヤ 1 が  $\mu_1$  に従うときの各状態から CEC に含まれる状態への最大到達確率を表している。よって、初期状態  $q_0$  から CEC への最大到達確率が 1 未満であるとき、プレイヤ 0 がどのような戦略に従ったとしても CEC へ到達できない確率は 0 より大きい。つまり、対抗戦略は存在しない。

そうでないとき、メモリレス局所最適対抗戦略探索手続き SolveOptLocStr の正当性（補題 9.27）より，ステップ 1-15 の結果として得られる  $Loc$  は、「MLCEC を含む MEC  $E_i$ 」と「“それに含まれる MLCEC の中で最大のメモリレス局所最適性能値  $v_i$ ”と“それを与えるメモリレス局所最適対抗戦略を表すベクトル  $\mathbf{x}$ ”」の対の集合となっている。そして、最大到達期待値計算手続き SolveReachExp の正当性（補題 9.30）より、 $\mathbf{e}$  はプレイヤ 1 が  $\mu_1$  に従うときの各状態の最大期待性能値を表している。よって、 $q_0$  の最大期待性能値が  $-\infty$ ，すなわち、 $q_0$  から MLCEC への最大到達確率が 1 未満であるとき、プレイヤ 0 がどのような戦略に従ったとしても MLCEC へ到達できない確率は 0 より大きい。つまり、メモリレス対抗戦略は存在しない。

そうでないとき、 $\mathbf{e}$  はそれぞれの状態での  $(\varepsilon-)$  最適な最大性能期待値を表すため、 $\mu_0^{Loc, \mathbf{e}}$  は明らかにメモリレス  $(\varepsilon-)$  最適対抗戦略となっている。以上により示された。 □

## 9.5 計算量

本章では、MUTMPRG 最適対抗戦略合成手続き（手続き 9.31）と  $LTL^{mp}$ -平均ケース最適合成問題の計算量について述べる。

### MUTMPRG 最適対抗戦略合成手続きの計算量

MUTMPRG 最適対抗戦略合成手続きの各ブロックの計算量は以下の通りである。

- ステップ 1-15

まず、MEC 分解はゲームのサイズに関して多項式時間で計算できる。ゲーム上の MREC の総サイズは、その極大性より、ゲームのサイズと Rabin 対の個数に関して線形である。そして、それぞれの MREC に対する CEC 判定は、CEC 判定手続き SolveCEC（手続き SolveCEC）により対象とする MREC のサイズに対して多項式時間で計算できる。そのため、このブロックの計算量はゲームのサイズに関して多項式時間である。

- ステップ 16-21 及び 34-40

最大到達確率計算手続き SolveReachProb（手続き 9.15）と最大到達期待値計算手続き SolveReachExp（手続き 9.29）はどちらもゲームのサイズに対して多項式時間で計算できるため、この二つのブロックの計算量はゲームのサイズに関して多項式時間である。

- ステップ 22-33

メモリレス局所最適戦略探索手続き SolveOptLocStr（手続き 9.26）の計算量は注意 9.28 で述べたように、指数時間オーダーである。そのため、このブロックの計算量はゲームのサイズに関して指数時間である。

よって、MPRG メモリレス最適対抗戦略合成手続きの計算量は指数時間オーダーである。なお、「与えられたメモリレス戦略が対抗戦略であるか否か」は多項式時間で検証可能であるため、最適化を行わない MUTMPRG メモリレス対抗戦略合成問題は NP に属す。

### LTL<sup>mp</sup>- 平均ケース最適合成問題の計算量

補題 9.2 より、LTL<sup>mp</sup>- 仕様  $\varphi = \psi \wedge \Theta$  と等価な MTMPRA は最悪で  $\mathcal{O}(2^{|\psi|})$  の状態数をもつ。そして、9.5 節で述べたように、MUTMPRG 最適対抗戦略合成手続き（手続き 9.31）はゲームのサイズに対して指数時間で計算できる。従って、LTL<sup>mp</sup>- 平均ケース最適合成の計算量は 3 重指数時間である。

LTL<sup>mp</sup>- 平均ケース最適合成問題においては、LTL 合成問題と同様に仕様と等価な決定性オートマトンの構成における計算量がボトルネックとなっている。

ただし,  $LTL^{MP-}$  平均ケース最適合成の DRA (DMTMPRA) の構成に関しては, 平均利得式を上手く用いることで, 計算量を抑制できる可能性がある. 例えば,

$$\Box(\text{req} \rightarrow \text{wait } \mathbf{U} \text{ res}) \quad (9.39)$$

という LTL 式は, 「常に, “リクエスト req が生起したとき, 応答 res するまで待ち wait 続ける”」という性質を表すが, 平均利得式も用いることで同じ意図の性質を以下のように表現できる.

$$\Box(\text{req} \rightarrow \text{wait } \mathbf{W} \text{ res}) \wedge \overline{MP}(\mathbf{1}_{\text{wait}}) < 1 \quad (9.40)$$

このとき, 式 (9.39) は「req が成り立ったならばいつか必ず res も成り立つ」という強いイベンチャリティ性を表している. 一方, 式 (9.40) の LTL 部分では「req が成り立ったとしても, 必ずしもその後 res が成り立たなくてもよい」という弱いイベンチャリティ性を表している. こうした違いは, それらから構成される等価な決定性オートマトンの構造に大きな影響を与える. これらは同じ意図を持つてはいるが通常は同じ意味ではないため, 通常 (つまり最悪ケース) の実現可能性判定/合成においては式 (9.39) を式 (9.40) で置き換えることはできない. しかし, 平均ケースの有限メモリ実現可能性判定/合成においては, RS と環境のふるまいが有限状態 DTMC としてモデル化されるため, 「無限にしばしば  $\psi$  が成り立つ ( $\Box \Diamond \psi$ )」と「 $\psi$  が 0 より大きい頻度で成り立つ ( $MP(\mathbf{1}_\psi) > 0$ )」は真偽が一致する. そのため,  $LTL^{MP-}$  平均ケース最適合成問題においては, 解析に適した表現を選択して仕様を記述することで, 検証を効率化できる可能性がある.



## 第IV部

### 結論



## 第10章 関連研究

本章では、本研究と関連した研究を簡単に紹介し、比較する。10.1 節では、近年になって研究されるようになってきた定量的言語と本研究の関連について述べる。10.2 節では、最も早くから研究されていた定量的拡張時間論理である実時間論理と本研究の関連について述べる。10.3 節では、生命システムなど様々なシステムの解析にも用いられている確率時間論理と本研究の関連について述べる。10.4 節では、既存の平均利得時間論理と本研究で開発した  $LTL^{mp}$  の関連について述べる。そして、10.5 節では、定量的合成の際に用いられている定量的拡張ゲームについての既存研究と本研究の関連について述べる。

### 10.1 定量的言語

定量的な性質を反映させた仕様記述には、二つのアプローチがある。一つは定量的な性質を記述できるように拡張した**ブール言語** (Boolean languages) を用いるアプローチであり、もう一つは**定量的言語** (quantitative languages) を用いるアプローチである。言語は、狭義には語（あるいは木）の集合として定義されるが、広義には語に対して重み空間  $X$  上の値  $x \in X$  を与える全関数  $L$  として定義される。

重み空間  $X$  がブール代数であるとき、その言語をブール言語といい、言語  $L$  は集合  $\{\sigma \mid L(\sigma) \text{ が真}\}$  として解釈できる。つまり、語の集合として定義される狭義の言語は、すべてブール言語である。ブール言語であっても、言語を記述する論理やオートマトンに実時間、確率や利得などを導入することで、定量的制約を満たす語の集合（言語）を規定できる。

一方で、重み空間  $X$  が実ベクトル空間  $\mathbb{R}^k$  などのなんらかの順序関係が定義できる集合として定義されるとき、その言語は定量的言語である。[43] では、有限状態オートマトンと単項二階理論 S1S の等価性 [22] が完備半環重みの上で一般化できることを示した。また、[57] では重み空間として束を持つような定量的 LTL 拡張が提案されている。それらの汎用的な重み付き論理とは異なり、[30] は、最大値、上極限/下極限や平均利得などに関するさまざまな定量

的言語クラスについて研究している。平均利得に関しては、[30, 6, 28, 27]などでより深く研究されている。

Alur らは、ふるまいの定量的な側面を捉えることができ、かつ、扱いやすいブール言語として多閾値平均利得言語 [6] (multi-threshold mean-payoff language) を提案している。これは、多閾値平均利得受理条件を満たす実行を持つ語を受理する全域的決定性オートマトンによって認識される言語であり、その空判定問題は決定可能である [6]。

本研究で開発した PFTL と  $LTL^{\text{mp}}$  はどちらも定量的性質を記述できるように拡張したブール型の言語である。また、 $LTL^{\text{mp}}$  を認識するために開発した多閾値平均利得  $\omega$ -オートマトンは、Alur らのオートマトン [6] と古典的な  $\omega$ -オートマトンを合わせたオートマトンであるが、このオートマトンの認識する言語もブール言語である。

## 10.2 実時間論理

実時間論理 [5][56][94][3][9][4][20] は、実クロック等を導入し実時間的な制約を記述できるように拡張した時間論理である。これまでに、LTL ベースの Timed Propositional Temporal Logic [5] (TPTL)、と Metric Temporal Logic [56] (MTL)、分岐時間論理である計算木論理 [38] (Computation Tree Logic, CTL) をベースとした Timed CTL [3, 4] (TCTL)、区間論理の Duration Calculus [94] (DC) などの時間論理が提案されている。例えば、MTL では「常に、 $\varphi$  が成り立ったときには 2 単位時間以内に  $\psi$  が成り立つ ( $= \Box(\varphi \rightarrow \Diamond_{\leq 2}\psi)$ )」のような実時間制約を記述することができる。また、DC では「 $\varphi$  が成り立っている時間が  $\psi$  が成り立っている時間より短い」などの頻度に言及する性質も記述できる。

しかしながら、CTL の拡張である TCTL のモデル検査を除いて、それらの決定問題は決定不能であった。ただし、MTL フラグメントである Metric Interval Temporal Logic [9] (MITL) や Bounded-MTL [20], CoFlat-MTL [20] のように、構文を制限することでモデル検査/充足可能性判定が決定可能となることも知られている。また、実時間モデル検査器 UPPAAL [83, 15] などが開発されており、TCTL フラグメントで記述された仕様を時間オートマトン [7][8] で与えられたモデルが満たすかを検査できる。そして、[35] では時間有界 MTL 仕様の CTMC に対する確率的モデル検査手法も開発されている。さらに、実現可能性が決定可能な実時間論理も提案されている [42]。ただし、その実現可能性判定及び合成手法は性能を最適化するものではない。

本研究で開発した時間論理 PFTL は連続時間意味論をもった確率時間論理

であり MTL で記述できる実時間的性質を記述できる。そしてさらに頻度演算子が導入されているため、非常に高い表現力を持っている。そのため、厳密なモデル検査は難しい。本研究では統計的手法を用いた時間有界モデル検査手法を与えた。この手法は、厳密な検証が困難、あるいは、決定不能である実時間論理や連続状態をもつモデルに対しても適用可能である。

また、本研究で開発したもう一つの時間論理  $LTL^{\text{mp}}$  は、直接的に実時間制約を記述することはできない。ただし、与えられる利得を時間経過と解釈することで、ある性質が成り立つ時間の割合に関する制約など一部の性質を間接的に記述することはできる。そして、 $LTL^{\text{mp}}$  はモデル検査と充足可能性判定が決定可能だけでなく、最適化や平均ケース実現可能性判定/最適合成などについても解くことができる。

## 10.3 確率時間論理

確率時間論理 [51, 11, 13, 12, 10] は基本的には CTL に確率の概念を導入した拡張時間論理であり、ふるまいの生起確率に関する性質を記述できる。CTL では「すべてのパスで  $\psi$  を満たす ( $A[\psi]$ )」や「あるのパスで  $\psi$  を満たす ( $E[\psi]$ )」のようなパスの分岐に関する定性的な性質を記述できた。その確率的拡張である確率計算木論理 [51] (probabilistic CTL, PCTL) では、「 $\varphi$  を満たすパスの生起確率が 80% 以上である ( $P_{\geq 0.8}[\psi]$ )」のような定量的な性質を記述できる。また、その連続時間拡張である連続確率論理 [11, 12] (continuous stochastic logic, CSL) も提案されている。モデル検査が容易である CTL の拡張であること、そして、確率測度が 0 となるふるまいの集合を無視できるため、実時間モデルや報酬 (利得) 構造を持ったモデル [13, 10] などに対する検査可能である。さらに、充足確率問題を解くことで定量的解析も可能である。すなわち、「ある性質の満たすふるまいの生起確率がある閾値より大きいかな否か」ではなく、「ある性質の満たすふるまいの生起確率」自体を計算できる。そのため、確率時間論理は定量的な性質を反映した仕様を記述するための言語としてしばしば用いられ、PRISM [70, 58] や Ymer [88, 89] などの確率的モデル検査器が開発されている。PRISM は、離散時間 Markov 連鎖や Markov 決定過程などの単純な離散時間確率モデルだけでなく、連続時間 Markov 連鎖や確率時間オートマトン [14] のような実時間モデルを検証することができる。さらにそれに報酬構造 [13, 10] を導入したモデルの累積報酬の期待値などに関する性質も検証できる。また、Ymer は、統計的手法を用いたモデル検査法 [92] を実装したモデル検査器であり、連続時間 Markov 連鎖だけでなく、半 Markov 過程

モデルも検証の対象にできる。

しかしながら、最も単純な確率時間論理である PCTL の充足可能性判定が決定可能であるか否かは未解決問題である。ただし、PCTL の定性的フラグメントの充足可能性判定については決定可能であることが示されている [21]。リアクティブシステム仕様の充足/実現可能性判定では、環境が何らかの前提条件を満たしているという仮定をしばしば設定する。そして性能要求を反映した仕様ではリクエスト生起確率に関する定量的制約などはごく自然な前提条件であるが、その充足可能性（無矛盾性）を検証する手法はない。

一方で、本研究で開発した PFTL では確率だけでなく頻度の概念をとらえることができるため、揺らぎをもった確率的システムのふるまいの性質を直観的に記述できる。また、CTL 型フラグメント PFCTL については既存の確率時間論理と同様のアプローチでモデル検査ができる。また、LTL 型フラグメント PFTL については、離散時間モデル検査では古典的な LTL モデル検査に帰着できた。しかしながら、連続時間モデル検査では厳密な検査は困難であったため、本研究では統計的手法を用いた時間有界モデル検査手法を与えた。PFTL の頻度演算子を利用することでふるまいの揺らぎを記述しやすく、多重検定による精度の劣化も起こらないため、統計的手法に適している。

LTL<sup>mp</sup> では、頻度をさらに一般化した平均利得に関する性質を記述できる。ただし、確率時間論理の報酬拡張とは異なり累積報酬に関する性質には言及できない。しかしながら、DTMC に対する確率的 LTL<sup>mp</sup> モデル検査は決定可能である。また、確率的にふるまう環境のもとでの実現可能性判定/最適合成手法を与えている。確率的環境を確率時間論理によって制約的に記述することは困難ではないが、定量的な確率時間論理の充足可能性/実現可能性判定の決定可能性は未解決であるため、本研究の平均ケース実現可能性判定/合成手法では環境を具体的なシステムとして与えている。

## 10.4 平均利得時間論理

Boker らは、[19] の中で *path-accumulation assertions*（平均利得の閾値に関する制約）を導入した LTL の拡張論理である LTL<sup>lim</sup> とそのモデル検査アルゴリズムを提案した。このモデル検査では、状態に変数  $v$  の値が割り当てられたモデルに対して、「実行上の変数  $v$  に関する平均値の上極限が 4 以下である（ $= \text{LimSupArv}(v) < 4$ ）」という制約式を追加した LTL 仕様の検査を行うことができる。そのアルゴリズムは、仕様の LTL 部分を反映した Büchi 条件を考慮して、[6] で与えられた多閾値平均利得受理条件をもつ全域的決定性オートマト

ンの空判定手続きを改良したものであり、その時間計算量は  $LTL^{\text{lim}}$  仕様のサイズに関して二重指数である。そして、 $LTL^{\text{lim}}$  の平均利得式は、確率モデル検査と同様にモデル上で設定されている利得に対する制約であり、充足可能性判定ではほとんど意味がない。また、 $LTL$  実現可能性判定器 Acacia+[2] を開発している Bohy らは、 $LTL$  に単純な平均利得制約を加えた  $LTL_{\text{MP}}$  [18] を、本研究とは独立に開発している。 $LTL_{\text{MP}}$  は  $LTL^{\text{lim}}$  とは異なり、論理の中でリテラルの充足に対する利得を  $LTL_{\text{MP}}$  の中で設定できる。そして、(多目的) 平均利得 Parity ゲームの有限メモリ勝利戦略問題の解法アルゴリズム [25] ([34]) を利用して、(最悪ケース) 有限メモリ実現可能性判定/合成を行えるように Acacia+ を改良している。ただし、複数の平均利得制約が与えられた場合には有効な最適化手法は提案されていない。

本研究で開発した時間論理  $LTL^{\text{mp}}$  は、 $LTL^{\text{lim}}$  と  $LTL_{\text{MP}}$  より真に表現力が高い時間論理である。また、 $LTL^{\text{mp}}$  文によって定義された言語を認識するための、Büchi 条件と多閾値平均利得受理条件を同時に満たす実行を受理するオートマトンを定義している。このオートマトンの空判定は、[19] の  $LTL^{\text{lim}}$  モデル検査アルゴリズムを利用して決定可能であるが、本研究では線形制約充足問題に帰着して指数時間で解くアルゴリズムを開発した。そして、 $LTL^{\text{mp}}$  文で定義される言語を認識するオートマトンの効率的な空判定アルゴリズムを利用することで、充足可能性判定や最適化問題も指数時間で決定できる。さらに、この最適化には LP 法を用いているため、多目的  $LTL^{\text{mp}}$  最適化には多目的 LP 法を用いることもできる。そして、有限メモリ環境のもとでの (平均ケース) 実現可能性判定/最適合成手法を与えた。この最適合成にも LP 法を用いており、その LP の応用技術はそのまま適用可能である。

## 10.5 定量的ゲーム

古典的な自動プログラム合成 [23] や  $LTL$  実現可能性判定 [68] は、定性的な勝利条件をもつ二人ゲームの勝利戦略問題に帰着して解くことができる。近年になって、Chatterjee らによって、定性的な勝利条件に加えて平均利得などに関する目的関数を設定した定量的拡張ゲーム [25, 29, 26, 34] が研究されている。もともとは平均利得の概念は、平均利得ゲーム [45] など、ゲーム理論において研究がなされていたが、それを定量合成へ応用する研究が行われるようになってきている。そして、それらのゲームの最適戦略問題に帰着する定量合成手法 [16, 86] が開発されている。ここで行われる最適化はどんな環境であつても保証される性能の最適化であり、最悪ケース定量合成と呼ばれる。平均利得

時間論理  $LTL_{MP}$  の最悪ケース合成でも定量的拡張ゲーム [34] の勝利戦略問題に帰着している。

もう一つの定量合成アプローチとして、確率的にふるまう具体的な環境を想定する平均ケース定量合成 [32, 86] がある。これは、特定の環境下での平均的な性能を最適化する手法であり、システムの使用状況が既知である場合には有効な手法である。平均ケース定量的合成問題はしばしば Markov 決定過程の最適戦略問題に帰着でき、LP 法を用いて解くことができる [33, 32, 86]。本研究で与えた平均ケース実現可能性判定/最適合成では、 $LTL_{MP}$  の最悪ケース合成で用いていた定量的拡張ゲームより表現力の高い多全称閾値平均利得 Rabin ゲームの対抗戦略問題に帰着して解いている。また、この対抗戦略は既存の平均ケース定量合成と同様に LP 法を用いて最適化している。

## 第11章 おわりに

本研究では、定量的な性質を記述可能な以下の二つの拡張時間論理を開発した。

- 確率頻度時間論理 PFTL
- 平均頻度線形時相論理 LTL<sup>mp</sup>

そして、これらによって記述された仕様に関してモデル検査、充足可能性判定、最適化、実現可能性判定、最適合成などの解析手法を開発した。

PFTLは確率と頻度の概念が導入された拡張時間論理であり、揺らぎをもったふるまいの性質を記述するのに適している。既存の確率時間論理では、一つの時間系列の性質を通常の定性的時間演算子でしか記述できなかったために、「いつかある性質を満たす」や「常にある性質を満たす」など性質しか捉えることができなかった。一方で、PFTLでは定量的時間演算子によって以下のような性質を記述できるため、揺らぎをもったふるまいの性質をより直観的に記述できる。

- 「ある性質  $\varphi_1$  が成り立つ頻度が80%以上である」
- 「ある性質  $\varphi_2$  が成り立つという条件の下で  $\varphi_1$  が成り立つ条件付き頻度が10%以下である」

PFTLには離散時間上での意味論と連続時間上での意味論が与えられており、実時間的な性質について捉えることができる。

そして本研究では、PFTLで記述された仕様について DTMC/CTMC に対する離散時間/連続時間モデル検査手法を開発した。

CTL型フラグメント PFCTL に関しては、既存の離散時間/連続時間確率的モデル検査と同様に数値計算アプローチによる厳密な検査手法を与えた。一方で、LTL型フラグメント PFLTL に関しては、離散時間モデル検査では、状態と時間の粒度が等しいことを利用して、確率的 LTL モデル検査に帰着する厳

密な検査手法を与えた。しかしながら、連続時間モデル検査では、状態と時間の粒度が異なるために、多くの実時間論理と同様に厳密に検査することは困難である。そこで、時間有界な PFLTL に関して、統計的アプローチを用いた近似的検査手法を与えた。統計的アプローチによる近似的モデル検査は既存の確率時間論理についても以前から提案されている。しかし、これらの時間論理はモデル検査が容易な CTL 型時間論理であり、その仕様についての厳密な検査は計算可能である。また、それらの時間論理仕様の統計的モデル検査では多重検定が必要となるため、精度が保証することが難しい。そのため、本来は統計的モデル検査に適した仕様記述言語でない。一方で、PFLTL の統計的モデル検査手法は、本来検証困難あるいは不能である仕様についての近似的なモデル検査であり、また、多重検定を必要としないため高い精度での検証も困難ではない。そして、PFLTL では、分岐の可能性に関する性質の直接的に記述は制限されているものの、頻度式によって揺らぎをもったふるまいの定量的な性質を充分捉えることができる。さらに、この手法は確率的シミュレーション可能であればどのようなモデルに対しても適用可能である。そのため、無限状態 CTMC や確率微分方程式など、通常はモデル検査では扱うことができないモデルであっても検査することができる。そして、常微分方程式モデルの安定性解析などにも用いることができる。

LTL<sup>mp</sup> は、平均利得の概念が導入された拡張 LTL である。これまでに開発されていた、利得の概念を導入した時間論理 [19] では、利得付きモデル上で定義された利得に関する性質しか記述できなかった。そのため、モデル検査のための仕様記述言語としては適してはいたが、利得に関する性質の充足可能性について言及するには適していなかった。一方で、LTL<sup>mp</sup> では構文中で利得の与え方を設定でき、その利得の与え方のもとの平均利得に関する性質を記述できる。それによって、以下のような頻度や平均コストに関する性質を記述できる。

- 「ある性質を満たす頻度が 80% 以上」
- 「ある資源について、あるイベントの生起に 3、別のあるイベントの生起に 1 消費するとき、長期的な平均資源消費量が 2 以下である」

そして、モデルとは独立に定義された利得の与え方が定義できるため、その充足可能性について検証することもできる。

線形時間論理の検証では一般に、仕様  $\varphi$  と等価なオートマトンを構成し、それを検証に用いることが多い。LTL<sup>mp</sup> についても、平均利得受理条件 [6] を導入した拡張  $\omega$ -正規オートマトンに変換することで検証することができる。既

存の研究 [19] により, このオートマトンの空判定問題と  $LTL^{mp}$  フラグメントのモデル検査が決定可能であることは示されていたが, その判定アルゴリズムの計算量はそれぞれ指数時間と二重指数時間であった. 本研究では, その空判定問題が線形制約充足問題に帰着することができ, 多項式時間で決定できることを示した. そして,  $LTL^{mp}$  のモデル検査と充足可能性判定問題は, このオートマトンの空判定問題に帰着することで指数時間で決定可能であることを示した. また,  $LTL^{mp}$  充足可能性判定問題が最終的に線形制約充足問題に帰着できることを利用して,  $LTL^{mp}$  で記述された仕様を線形計画法を用いて最適化できることを示した.

さらに, 有限メモリ環境のもとで  $LTL^{mp}$  フラグメント仕様の実現可能性判定/最適合成手法を与えた. 時間論理仕様からのプログラム合成問題では, 仕様と等価な決定性オートマトンの構造的特徴を持ったゲームの勝利戦略問題に帰着するアプローチが一般的である. 一般に  $LTL^{mp}$  は決定性の平均利得拡張  $\omega$ -正規オートマトンでは認識できないため, 本研究ではまず, 実現可能性判定のための  $LTL^{mp}$  フラグメント  $LTL^{mp-}$  を開発した. そして,  $LTL^{mp-}$  で記述された仕様が与えられた有限メモリ環境のもとで実現可能であるか判定し, 可能であるときには与えられた平均利得目的関数を最適化する実現プログラムを合成する手法を開発した. この手法は,  $LTL^{mp}$  仕様の充足可能性判定と最適化手法を拡張したものであり, 線形計画法を用いて最適な実現プログラムを計算する. 既存の平均利得拡張時間論理仕様からのプログラム合成手法 [17] では,  $LTL^{mp}$  より表現力が低い平均利得拡張  $LTL$  仕様に対して, もっとも悪い状況でも仕様を実現するプログラムを合成していた. 一方で, 本手法で合成する実現プログラムは, 想定されている使用状況において最適の性能を持つプログラムである. 本手法の大きな利点としては, 線形計画法を最適化に利用しているため, 多目的線形計画法などの応用的な線形計画法を, 多目的  $LTL^{mp-}$  最適合成に適用可能な点が挙げられる.



## 謝辞

本研究を行うにあたり，ご指導を頂きました米崎直樹教授に心から感謝を申し上げます。また，数々のご助言を頂きました萩原茂樹助教，並びに米崎研究室関係者の皆様に深く感謝致します。



## 参考文献



- [1] Martín Abadi, Leslie Lamport, and Pierre Wolper. Realizable and unrealizable specifications of reactive systems. In Giorgio Ausiello, Mariangiola Dezani-Ciancaglini, and Simonetta Della Rocca, editors, *Automata, Languages and Programming*, volume 372 of *Lecture Notes in Computer Science*, pages 1–17. Springer Berlin / Heidelberg, 1989.
- [2] Acacia+. <http://lit2.ulb.ac.be/acaciaplus/>.
- [3] R. Alur, C. Courcoubetis, and D. Dill. Model-checking for real-time systems. In *Logic in Computer Science, 1990. LICS '90, Proceedings., Fifth Annual IEEE Symposium on*, pages 414–425, jun 1990.
- [4] R. Alur, C. Courcoubetis, and D. Dill. Model-checking in dense real-time. *Information and Computation*, 104(1):2 – 34, 1993.
- [5] R. Alur and T.A. Henzinger. A really temporal logic. In *Foundations of Computer Science, 1989., 30th Annual Symposium on*, pages 164–169, oct-1 nov 1989.
- [6] Rajeev Alur, Aldric Degorre, Oded Maler, and Gera Weiss. On omega-languages defined by mean-payoff conditions. In Luca de Alfaro, editor, *Foundations of Software Science and Computational Structures*, volume 5504 of *Lecture Notes in Computer Science*, pages 333–347. Springer Berlin / Heidelberg, 2009.
- [7] Rajeev Alur and David Dill. Automata for modeling real-time systems. In MichaelS. Paterson, editor, *Automata, Languages and Programming*, volume 443 of *Lecture Notes in Computer Science*, pages 322–335. Springer Berlin Heidelberg, 1990.
- [8] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183 – 235, 1994.
- [9] Rajeev Alur, Tomás Feder, and Thomas A. Henzinger. The benefits of relaxing punctuality. In *Proceedings of the tenth annual ACM symposium on Principles of distributed computing*, PODC '91, pages 139–152, New York, NY, USA, 1991. ACM.
- [10] Suzana Andova, Holger Hermanns, and Joost-Pieter Katoen. Discrete-time rewards model-checked. In Kim Larsen and Peter Niebert, editors,

- Formal Modeling and Analysis of Timed Systems*, volume 2791 of *Lecture Notes in Computer Science*, pages 88–104. Springer Berlin / Heidelberg, 2004.
- [11] Adnan Aziz, Kumud Sanwal, Vigyan Singhal, and Robert Brayton. Verifying continuous time markov chains. In Rajeev Alur and Thomas Henzinger, editors, *Computer Aided Verification*, volume 1102 of *Lecture Notes in Computer Science*, pages 269–276. Springer Berlin / Heidelberg, 1996.
  - [12] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time markov chains. *Software Engineering, IEEE Transactions on*, 29(6):524 – 541, june 2003.
  - [13] Christel Baier, Boudewijn Haverkort, Holger Hermanns, and Joost-Pieter Katoen. On the logical characterisation of performability properties. In Ugo Montanari, José Rolim, and Emo Welzl, editors, *Automata, Languages and Programming*, volume 1853 of *Lecture Notes in Computer Science*, pages 780–792. Springer Berlin / Heidelberg, 2000.
  - [14] Danièle Beauquier. On probabilistic timed automata. *Theoretical Computer Science*, 292(1):65–84, jan 2003.
  - [15] Gerd Behrmann, Alexandre David, and Kim Larsen. A tutorial on uppaal. In Marco Bernardo and Flavio Corradini, editors, *Formal Methods for the Design of Real-Time Systems*, volume 3185 of *Lecture Notes in Computer Science*, pages 33–35. Springer Berlin / Heidelberg, 2004. The latest version is available on [83]).
  - [16] Roderick Bloem, Krishnendu Chatterjee, Thomas Henzinger, and Barbara Jobstmann. Better quality in synthesis through quantitative objectives. In Ahmed Bouajjani and Oded Maler, editors, *Computer Aided Verification*, volume 5643 of *Lecture Notes in Computer Science*, pages 140–156. Springer Berlin / Heidelberg, 2009.
  - [17] Aaron Bohy, Véronique Bruyère, Emmanuel Filiot, Naiyong Jin, and Jean-François Raskin. Acacia+, a tool for ltl synthesis. In P. Madhusudan and Sanjit Seshia, editors, *Computer Aided Verification*, volume 7358

- of *Lecture Notes in Computer Science*, pages 652–657. Springer Berlin / Heidelberg, 2012.
- [18] Aaron Bohy, Véronique Bruyère, Emmanuel Filiot, and Jean-François Raskin. Synthesis from ltl specifications with mean-payoff objectives. In Nir Piterman and Scott A. Smolka, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, volume 7795 of *Lecture Notes in Computer Science*, pages 169–184. Springer Berlin Heidelberg, 2013.
- [19] U. Boker, K. Chatterjee, T.A. Henzinger, and O. Kupferman. Temporal specifications with accumulative values. In *Logic in Computer Science (LICS), 2011 26th Annual IEEE Symposium on*, pages 43–52, june 2011.
- [20] Patricia Bouyer, Nicolas Markey, Joël Ouaknine, and James Worrell. On expressiveness and complexity in real-time model checking. In Luca Aceto, Ivan Damgård, Leslie Goldberg, Magnús Halldórsson, Anna Ingólfssdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming*, volume 5126 of *Lecture Notes in Computer Science*, pages 124–135. Springer Berlin / Heidelberg, 2008.
- [21] Tomáš Brázdil, Vojtěch Forejt, Jan Křetínský, and Antonín Kučera. The satisfiability problem for probabilistic ctl. In *Proceedings of the 2008 23rd Annual IEEE Symposium on Logic in Computer Science, LICS '08*, pages 391–402, Washington, DC, USA, 2008. IEEE Computer Society.
- [22] J. Richard Büchi. Symposium on decision problems: On a decision method in restricted second order arithmetic. In Patrick Suppes Ernest Nagel and Alfred Tarski, editors, *Logic, Methodology and Philosophy of Science Proceeding of the 1960 International Congress*, volume 44 of *Studies in Logic and the Foundations of Mathematics*, pages 1 – 11. Elsevier, 1966.
- [23] J. Richard Büchi and Lawrence H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the American Mathematical Society*, 138:pp. 295–311, 1969.
- [24] Doron Bustan, Sasha Rubin, and Moshe Y. Vardi. Verifying omega-regular properties of markov chains. In Rajeev Alur and Doron A. Peled, editors, *Computer Aided Verification*, volume 3114 of *Lecture Notes in Computer Science*, pages 189–201. Springer Berlin Heidelberg, 2004.

- [25] K. Chatterjee, T.A. Henzinger, and M. Jurdzinski. Mean-payoff parity games. In *Logic in Computer Science, 2005. LICS 2005. Proceedings. 20th Annual IEEE Symposium on*, pages 178 – 187, june 2005.
- [26] K. Chatterjee and Y. Velner. Mean-payoff pushdown games. In *Logic in Computer Science (LICS), 2012 27th Annual IEEE Symposium on*, pages 195 –204, june 2012.
- [27] Krishnendu Chatterjee, Laurent Doyen, Herbert Edelsbrunner, Thomas Henzinger, and Philippe Rannou. Mean-payoff automaton expressions. In Paul Gastin and François Laroussinie, editors, *CONCUR 2010 - Concurrency Theory*, volume 6269 of *Lecture Notes in Computer Science*, pages 269–283. Springer Berlin / Heidelberg, 2010.
- [28] Krishnendu Chatterjee, Laurent Doyen, and Thomas Henzinger. Alternating weighted automata. In Mirosław Kutylowski, Witold Charatonik, and Maciej Gebala, editors, *Fundamentals of Computation Theory*, volume 5699 of *Lecture Notes in Computer Science*, pages 3–13. Springer Berlin / Heidelberg, 2009.
- [29] Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger, and Jean-François Raskin. Generalized Mean-payoff and Energy Games. In Kamal Lodaya and Meena Mahajan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2010)*, volume 8 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 505–516, Dagstuhl, Germany, 2010. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [30] Krishnendu Chatterjee, Laurent Doyen, and ThomasA. Henzinger. Quantitative languages. In Michael Kaminski and Simone Martini, editors, *Computer Science Logic*, volume 5213 of *Lecture Notes in Computer Science*, pages 385–400. Springer Berlin Heidelberg, 2008.
- [31] Krishnendu Chatterjee and Monika Henzinger. Faster and dynamic algorithms for maximal end-component decomposition and related graph problems in probabilistic verification. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 1318–1336. SIAM, 2011.

- [32] Krishnendu Chatterjee, Thomas Henzinger, Barbara Jobstmann, and Rohit Singh. Measuring and synthesizing systems in probabilistic environments. In Tayssir Touili, Byron Cook, and Paul Jackson, editors, *Computer Aided Verification*, volume 6174 of *Lecture Notes in Computer Science*, pages 380–395. Springer Berlin / Heidelberg, 2010.
- [33] Krishnendu Chatterjee, Rupak Majumdar, and Thomas Henzinger. Markov decision processes with multiple objectives. In Bruno Durand and Wolfgang Thomas, editors, *STACS 2006*, volume 3884 of *Lecture Notes in Computer Science*, pages 325–336. Springer Berlin / Heidelberg, 2006.
- [34] Krishnendu Chatterjee, Mickael Randour, and Jean-François Raskin. Strategy synthesis for multi-dimensional quantitative objectives. In Maciej Koutny and Irek Ulidowski, editors, *CONCUR 2012 - Concurrency Theory*, volume 7454 of *Lecture Notes in Computer Science*, pages 115–131. Springer Berlin Heidelberg, 2012.
- [35] Taolue Chen, Marco Diciolla, Marta Kwiatkowska, and Alexandru Mereacre. Time-bounded verification of ctmc's against real-time specifications. In Uli Fahrenberg and Stavros Tripakis, editors, *Formal Modeling and Analysis of Timed Systems*, volume 6919 of *Lecture Notes in Computer Science*, pages 26–42. Springer Berlin Heidelberg, 2011.
- [36] Alessandro Cimatti, Edmund Clarke, Enrico Giunchiglia, Fausto Giunchiglia, Marco Pistore, Marco Roveri, Roberto Sebastiani, and Armando Tacchella. Nusmv 2: An opensource tool for symbolic model checking. In Ed Brinksma and Kim Larsen, editors, *Computer Aided Verification*, volume 2404 of *Lecture Notes in Computer Science*, pages 241–268. Springer Berlin / Heidelberg, 2002.
- [37] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Trans. Program. Lang. Syst.*, 8(2):244–263, April 1986.
- [38] Edmund Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In Dexter Kozen, editor, *Logics of Programs*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer Berlin / Heidelberg, 1982.

- [39] EdmundM. Clarke and Paolo Zuliani. Statistical model checking for cyber-physical systems. In Tevfik Bultan and Pao-Ann Hsiung, editors, *Automated Technology for Verification and Analysis*, volume 6996 of *Lecture Notes in Computer Science*, pages 1–12. Springer Berlin Heidelberg, 2011.
- [40] Costas Courcoubetis and Mihalis Yannakakis. The complexity of probabilistic verification. *J. ACM*, 42(4):857–907, July 1995.
- [41] Jean-Michel Couvreur, Nasser Saheb, and Grégoire Sutre. An optimal automata approach to ltl model checking of probabilistic systems. In MosheY. Vardi and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning*, volume 2850 of *Lecture Notes in Computer Science*, pages 361–375. Springer Berlin Heidelberg, 2003.
- [42] Laurent Doyen, Gilles Geeraerts, Jean-Francois Raskin, and Julien Reicherter. Realizability of real-time logics. In Joël Ouaknine and Frits Vaandrager, editors, *Formal Modeling and Analysis of Timed Systems*, volume 5813 of *Lecture Notes in Computer Science*, pages 133–148. Springer Berlin / Heidelberg, 2009.
- [43] Manfred Droste and Paul Gastin. Weighted automata and weighted logics. In Luís Caires, Giuseppe Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *Automata, Languages and Programming*, volume 3580 of *Lecture Notes in Computer Science*, pages 104–104. Springer Berlin / Heidelberg, 2005.
- [44] Rüdiger Ehlers. Symbolic bounded synthesis. In Tayssir Touili, Byron Cook, and Paul Jackson, editors, *Computer Aided Verification*, volume 6174 of *Lecture Notes in Computer Science*, pages 365–379. Springer Berlin / Heidelberg, 2010.
- [45] A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8:109–113, 1979.
- [46] E. Allen Emerson and Joseph Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *Journal of Computer and System Sciences*, 30(1):1 – 24, 1985.

- [47] Emmanuel Filiot, Naiyong Jin, and Jean-François Raskin. An antichain algorithm for ltl realizability. In Ahmed Bouajjani and Oded Maler, editors, *Computer Aided Verification*, volume 5643 of *Lecture Notes in Computer Science*, pages 263–277. Springer Berlin / Heidelberg, 2009.
- [48] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.
- [49] GNU Linear Programming Kit. <http://www.gnu.org/software/glpk/>.
- [50] E. Moritz Hahn, Holger Hermanns, Björn Wachter, and Lijun Zhang. Time-bounded model checking of infinite-state continuous-time markov chains. *Fundamenta Informaticae*, 95(1):129–155, 2009.
- [51] Hans Hansson and Bengt Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6:512–535, 1994.
- [52] G.J. Holzmann. The model checker spin. *Software Engineering, IEEE Transactions on*, 23(5):279–295, may 1997.
- [53] IBM ILOG CPLEX Optimization Studio. <http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/>.
- [54] Barbara Jobstmann and Roderick Bloem. Optimizations for ltl synthesis. In *Formal Methods in Computer Aided Design, 2006. FMCAD '06*, pages 117–124, nov. 2006.
- [55] Joachim Klein and Christel Baier. Experiments with deterministic - automata for formulas of linear temporal logic. *Theoretical Computer Science*, 363(2):182–195, 2006.
- [56] Ron Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2:255–299, 1990.
- [57] Orna Kupferman and Yoad Lustig. Lattice automata. In Byron Cook and Andreas Podelski, editors, *Verification, Model Checking, and Abstract Interpretation*, volume 4349 of *Lecture Notes in Computer Science*, pages 199–213. Springer Berlin / Heidelberg, 2007.

- [58] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In G. Gopalakrishnan and S. Qadeer, editors, *Proceedings of the 23rd International Conference on Computer Aided Verification (CAV'11)*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011.
- [59] Marta Kwiatkowska, Gethin Norman, and David Parker. Stochastic model checking. In Marco Bernardo and Jane Hillston, editors, *Formal Methods for Performance Evaluation*, volume 4486 of *Lecture Notes in Computer Science*, pages 220–270. Springer Berlin Heidelberg, 2007.
- [60] Orna Lichtenstein and Amir Pnueli. Checking that finite state concurrent programs satisfy their linear specification. In *Proceedings of the 12th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, POPL '85, pages 97–107, New York, NY, USA, 1985. ACM.
- [61] Lily. [http://www.iaik.tugraz.at/content/research/design\\\_verification/lily/](http://www.iaik.tugraz.at/content/research/design\_verification/lily/).
- [62] LTL2BA. <http://www.lsv.ens-cachan.fr/~gastin/ltl2ba/>.
- [63] ltl2dstar. <http://www.ltl2dstar.de/>.
- [64] mathSAT 5. <http://mathsat.fbk.eu/>.
- [65] MATLAB. <http://www.mathworks.com/products/matlab/index.html>.
- [66] J. Neyman and E. S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231:289–337, 1933.
- [67] NuSMV. <http://nusmv.fbk.eu/>.
- [68] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proceedings of the 16th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '89, pages 179–190, New York, NY, USA, 1989. ACM.

- [69] Amir Pnueli. The temporal logic of programs. In *Foundations of Computer Science, 1977., 18th Annual Symposium on*, pages 46–57, 31 1977-nov. 2 1977.
- [70] PRISM. <http://www.prismmodelchecker.org/>.
- [71] Anne Remke, Boudewijn R. Haverkort, and Lucia Cloth. Model checking infinite-state markov chains. In Nicolas Halbwachs and LenoreD. Zuck, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, volume 3440 of *Lecture Notes in Computer Science*, pages 237–252. Springer Berlin Heidelberg, 2005.
- [72] R. Rosner. *Modular synthesis of reactive systems*. PhD thesis, Weizmann Institute of Science, 1992.
- [73] S. Safra. On the complexity of omega -automata. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science, SFCs '88*, pages 319–327, Washington, DC, USA, 1988. IEEE Computer Society.
- [74] Sven Schewe and Bernd Finkbeiner. Bounded synthesis. In KedarS. Namjoshi, Tomohiro Yoneda, Teruo Higashino, and Yoshio Okamura, editors, *Automated Technology for Verification and Analysis*, volume 4762 of *Lecture Notes in Computer Science*, pages 474–488. Springer Berlin Heidelberg, 2007.
- [75] Scilab. <http://www.scilab.org/>.
- [76] A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logics. *J. ACM*, 32(3):733–749, July 1985.
- [77] SPIN. <http://spinroot.com/spin/>.
- [78] SPOT. <http://spot.lip6.fr/wiki/>.
- [79] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.
- [80] Takashi Tomita, Shigeki Hagihara, and Naoki Yonezaki. A probabilistic temporal logic with frequency operators and its model checking. In Fang Yu and Chao Wang, editors, *Proceedings 13th International Workshop*

- on *Verification of Infinite-State Systems*, Taipei, Taiwan, 10th October 2011, volume 73 of *Electronic Proceedings in Theoretical Computer Science*, pages 79–93. Open Publishing Association, 2011.
- [81] Takashi Tomita, Shin Hiura, Shigeki Hagihara, and Naoki Yonezaki. A temporal logic with mean-payoff constraints. In *14th International Conference on Formal Engineering Methods (ICFEM 2012)*, volume 7635 of *Lecture Notes in Computer Science*, pages 249–265. Springer Berlin / Heidelberg, 2012.
  - [82] Unbeast. <http://www.react.uni-saarland.de/tools/unbeast/>.
  - [83] UPPAAL. <http://www.uppaal.org/>.
  - [84] Moshe Y. Vardi and Pierre Wolper. Reasoning about infinite computations. *Information and Computation*, 115:1–37, 1994.
  - [85] MosheY. Vardi. An automata-theoretic approach to linear temporal logic. In Faron Moller and Graham Birtwistle, editors, *Logics for Concurrency*, volume 1043 of *Lecture Notes in Computer Science*, pages 238–266. Springer Berlin Heidelberg, 1996.
  - [86] Pavol Černý, Krishnendu Chatterjee, Thomas Henzinger, Arjun Radhakrishna, and Rohit Singh. Quantitative synthesis for concurrent programs. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *Computer Aided Verification*, volume 6806 of *Lecture Notes in Computer Science*, pages 243–259. Springer Berlin / Heidelberg, 2011.
  - [87] A. Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, 1945.
  - [88] Ymer. <http://www.tempastic.org/ymer/>.
  - [89] Håkan L.S. Younes. Ymer: A statistical model checker. In Kousha Etesami and SriramK. Rajamani, editors, *Computer Aided Verification*, volume 3576 of *Lecture Notes in Computer Science*, pages 429–433. Springer Berlin Heidelberg, 2005.
  - [90] Håkan L.S. Younes, Edmund M. Clarke, and Paolo Zuliani. Statistical verification of probabilistic properties with unbounded until. In Jim Davies,

- Leila Silva, and Adenilso Simao, editors, *Formal Methods: Foundations and Applications*, volume 6527 of *Lecture Notes in Computer Science*, pages 144–160. Springer Berlin Heidelberg, 2011.
- [91] Håkan L.S. Younes, Marta Kwiatkowska, Gethin Norman, and David Parker. Numerical vs. statistical probabilistic model checking. *International Journal on Software Tools for Technology Transfer*, 8:216–228, 2006.
- [92] Håkan L.S. Younes and Reid G. Simmons. Probabilistic verification of discrete event systems using acceptance sampling. In Ed Brinksma and Kim Guldstrand Larsen, editors, *Computer Aided Verification*, volume 2404 of *Lecture Notes in Computer Science*, pages 223–235. Springer Berlin Heidelberg, 2002.
- [93] Z3. <http://research.microsoft.com/en-us/um/redmond/projects/z3/>.
- [94] Chaochen. Zhou, C.A.R. Hoare, and Anders P. Ravn. A calculus of durations. *Information Processing Letters*, 40(5):269 – 276, 1991.
- [95] Paolo Zuliani, André Platzer, and Edmund M. Clarke. Bayesian statistical model checking with application to simulink/stateflow verification. In *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, HSCC '10, pages 243–252, New York, NY, USA, 2010. ACM.
- [96] 富田堯, 萩原茂樹, 伊藤宗平, 樋浦信, and 米崎直樹. 頻度オペレータを導入した線形時間論理の統計的手法を用いた確率的モデル検査. In **日本ソフトウェア科学会第 28 回大会論文集**, 2011.
- [97] 富田堯, 萩原茂樹, and 米崎直樹. Mean-payoff 制約を記述可能な線形時間論理. In **日本ソフトウェア科学会第 29 回大会論文集**, 2012.
- [98] 富田 堯, 山崎徹郎, 萩原茂樹, and 米崎直樹. 確率的環境下において平均利得時間論理仕様を実現する最適マルコフ決定過程の自動合成法. In **日本ソフトウェア科学会第 30 回大会論文集**, 2013.



## 付録



## 付 録 A 正規言語

**正規言語** (regular language) は、最も広く用いられている形式言語の一つである。プログラミング言語など計算機が扱う言語の多くは正規言語であり、正規言語を表現するために開発された正規表現 (regular expression) はパターンマッチングなどでしばしば用いられている。

形式的には、以下のように定義される有限長言語である。

**定義 A.1 (正規言語)**  $\Sigma$  を有限のアルファベットとする。このとき、正規言語族 REG は以下を満たす集合  $X$  の中で最小の集合である。

- $A \subseteq \Sigma\{\epsilon\}$  ならば  $A \in X$ ,
- $E_1, E_2 \in X$  ならば  $E_1 \cup E_2, E_1 \cdot E_2, E_1^* \in X$ .

ここで、 $\cdot$  は連結 (単位元は  $\epsilon$ )、 $*$  は Kleene 閉包である。

- $E_1 \cdot E_2 = \{\sigma_1\sigma_2 \mid \sigma_1 \in E_1 \wedge \sigma_2 \in E_2\}$ ,
- $E_1^* = \bigcup_{k \in \mathbb{N}} E_1^k$ ,
  - $E_1^0 = \{\epsilon\}$ ,
  - $E_1^{n+1} = E_1^n \cdot E_1$ ,

ただし、 $\epsilon$  は空文字である。

また、言語  $L \in \text{REG}$  を正規言語という。 ■

この言語のクラスは、Boolean 演算に関して閉じていることが知られている。

**定理 A.2 (正規言語の閉包性)** 正規言語のクラスは、和集合、積集合、補集合に関して閉じている。 ■

そして、正規言語は、正規文法 (regular grammar, Type-3 grammar) によって生成でき、また、有限語上の有限状態オートマトン (finite state automata) が認識できる。



## 付 録 B 確率と統計

本章では、確率論で用いられる基本的な概念の定義といくつかの統計的手法について簡単に紹介する。

### B.1 完全加法族と確率空間

試行の結果の集合、つまり、標本空間が有限である場合、それぞれの結果が生起する確率は単純に定義できる。しかし、状態遷移系のふるまいの集合など標本空間が非可算無限である場合、それぞれの結果が生起する確率は一般に直接的に定義できない。そのため、非可算無限個の要素をもつ集合上で確率を定義するためには、確率を定義できる部分集合の族を考える必要がある。そのような確率を定義できる、すなわち、可測な部分集合の族を**完全加法族** (completely additive class) あるいは  $\sigma$ -代数 ( $\sigma$ -algebra) という。

**定義 B.1 (完全加法族)**  $\Omega$  をある集合とする。このとき、以下を満たす集合  $E \in 2^\Omega$  を  $\Omega$  上の  $\sigma$ -代数という。

- $\Omega \in E$ ,
  - $E$  は全体集合  $\Omega$  を含む。
- $A \in E \Rightarrow (\Omega \setminus A) \in E$ ,
  - $E$  は補集合に関して閉じている。
- $A_1, A_2, \dots \in E \Rightarrow \bigcup A_i \in E$ .
  - $E$  は可算和集合に関して閉じている。 ■

このとき、集合  $\Omega$  とその上の完全加法族  $E$  の対  $\langle \Omega, E \rangle$  を**可測空間** (measurable space) という。その可測空間に対して定義できる測度の中で確率の特性を満たす測度を**確率測度** (probability measure) と呼ぶ。ランダム性をもった

試行は、可測空間とその上の確率測度の対、**確率空間** (probability space) によってモデル化される。

**定義 B.2 (確率空間)** 確率空間 (probability space) は三つ組  $\langle \Omega, E, \Pr \rangle$  である。ただし、

- $\Omega$  は**標本空間** (sample space) ,
- $E$  は**事象** (event) の完全加法族,
- $\Pr : E \rightarrow [0, 1]$  は、可測空間  $\langle \Omega, E \rangle$  上の確率測度である。つまり、以下を満たす測度である。
  - $\Pr(\Omega) = 1$ ,
  - $\Pr(\emptyset) = 0$ ,
  - 任意の事象  $e \in E$  について、 $\Pr(e) \geq 0$ ,
  - 任意の互いに交わらない事象  $e_1, e_2, \dots \in E$  について、 $\Pr(\bigcup_{i \in \mathbb{N}} e_i) = \sum_{i \in \mathbb{N}} \Pr(e_i)$ . ■

直観的には、確率測度  $\Pr(e)$  は事象  $e$  に含まれる結果のうちの 하나가生起する確率である。

**例 B.3**  $1/2$  の確率で「表」、 $1/2$  の確率で「裏」が出るコインを一回投げる試行を考える。このとき、標本空間は  $\{\text{表}, \text{裏}\}$ 、事象の集合は  $2^{\{\text{表}, \text{裏}\}}$ 、確率測度  $\Pr$  は  $\Pr(\emptyset) = 0$   $\Pr(\{\text{表}\}) = \Pr(\{\text{裏}\}) = 1/2$   $\Pr(\{\text{表}, \text{裏}\}) = 1$  である。 ■

ここで、可積分な確率変数  $X : \Omega \rightarrow \mathbb{R}$  に対して、その**期待値** (expectation または expected value) は以下のように与えることができる。

**定義 B.4 (期待値)** 可積分な確率変数  $X$  に対して、 $\text{Expect}[X]$  を  $X$  の期待値という。

$$\text{Expect}[X] = \int_{\omega \in \Omega} X(\omega) d\Pr(\omega) \quad (\text{B.1})$$

ただし、確率変数  $X$  が可積分であるとは、以下を満たすことである。

$$\int_{\omega \in \Omega} |X(\omega)| d\Pr(\omega) < \infty. \quad (\text{B.2})$$

■

## B.2 Bernoulli 試行

例 B.3 のように，結果が成功か失敗かの二つしかない試行を **Bernoulli 試行** と呼ぶ．成功確率  $p$  の Bernoulli 試行を独立に  $n$  回行った際に，成功回数  $m$  が従う離散確率分布を **二項分布** (binomial distribution) という．

**定義 B.5 (二項分布)** パラメータ  $p \in (0, 1)$  と  $n \in \mathbb{N}_{>0}$  に対して，二項分布  $\mathfrak{Binom}_{p,n} : [0, n] \rightarrow [0, 1]$  は以下のように定義される．

$$\mathfrak{Binom}_{p,n}(m) = \frac{n!}{m!(n-m)!} \cdot p^m \cdot (1-p)^{n-m}. \quad (\text{B.3})$$

ここで， $\mathfrak{Binom}_{p,n}$  の平均は  $n \cdot p$ ，分散は  $n \cdot p \cdot (1-p)$  である． ■

パラメータ  $n$  が十分大きく，平均  $n \cdot p$  と分散  $n \cdot p \cdot (1-p)$  も小さくない場合，二項分布  $\mathfrak{Binom}_{p,n}$  は，平均  $n \cdot p$ ，分散  $n \cdot p \cdot (1-p)$  の **正規分布** (normal distribution) と呼ばれる連続確率分布で近似できる．中心極限定理より，独立で同一の確率分布に従う確率変数の和は，多くの場合正規分布に従う．そのため，多くの分野で，様々な分布や誤差が正規分布に従うという仮定されることが多い．

**定義 B.6 (正規分布)** パラメータ  $\mu \in \mathbb{R}$  と  $\sigma^2 \in \mathbb{R}_{>0}$  に対して，正規分布  $\mathfrak{Norm}_{\mu,\sigma^2} : \mathbb{R} \rightarrow [0, 1]$  は以下のように定義される．

$$\mathfrak{Norm}_{\mu,\sigma^2}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \quad (\text{B.4})$$

ここで， $\mathfrak{Norm}_{\mu,\sigma^2}$  の平均は  $\mu$ ，分散は  $\sigma^2$  である． ■

確率変数  $x$  が平均  $\mu$ ，分散  $\sigma^2$  の正規分布に従うとき， $z = \frac{x-\mu}{\sigma}$  は **標準正規分布**  $\mathfrak{Norm}_{0,1}$  (standard normal distribution) に従うことが知られている．統計的手法では，しばしばこれを利用して解析が行われる．

## B.3 Markov 性

系の未来の状態の確率分布が現在の状態のみに依存する性質を **Markov 性** (Markov property) という．これは，待ち行列などの実時間上であるイベントが繰り返しランダムに生起する確率過程のモデル化ではしばしば仮定される性

質であり、Markov 性をもつ確率過程を **Markov 過程** (Markov processes) と呼ぶ。

イベントの生起時間間隔に注目したとき、平均して  $1/\lambda$  時間後に生起するイベントが生起するまでの時間は、**指数分布** (exponential distribution) と呼ばれる連続確率分布に従う。

**定義 B.7 (指数分布)** パラメータ  $\lambda \in \mathbb{R}_{>0}$  に対して、指数分布  $\mathfrak{Exp}_\lambda : \mathbb{R}_{\geq 0} \rightarrow [0, 1]$  は以下のように定義される。

$$\mathfrak{Exp}_\lambda(x) = \lambda \cdot \exp(-\lambda \cdot x). \quad (\text{B.5})$$

ここで、 $\mathfrak{Exp}_\lambda$  の平均は  $1/\lambda$ 、分散は  $1/\lambda^2$  である。 ■

ここで、指数分布の累積分布関数は、

$$\int_0^x \mathfrak{Exp}_\lambda(y) dy = 1 - \exp(-\lambda \cdot x). \quad (\text{B.6})$$

また、イベントの生起回数に注目したとき、平均して単位時間あたり  $\lambda$  回起こるイベントが単位時間中に生起する回数の分布は、**Poisson 分布** (Poisson distribution) と呼ばれる離散確率分布に従う。

**定義 B.8 (Poisson 分布)** パラメータ  $\lambda \in \mathbb{R}_{>0}$  に対して、指数分布  $\mathfrak{Pois}_\lambda : \mathbb{N}_{\geq 0} \rightarrow [0, 1]$  は以下のように定義される。

$$\mathfrak{Pois}_\lambda(x) = \frac{\lambda^x \cdot \exp(-\lambda)}{x!}. \quad (\text{B.7})$$

ここで、 $\mathfrak{Pois}_\lambda$  の平均は  $\lambda$ 、分散は  $\lambda$  である。 ■

そして、イベントが生起した時間に注目したとき、平均して単位時間あたり  $\lambda$  回起こるイベントがちょうど  $k$  回目の生起時間の分布は、**Erlang 分布** (Erlang distribution) と呼ばれる連続確率分布に従う。

**定義 B.9 (Erlang 分布)** パラメータ  $\lambda \in \mathbb{R}_{>0}$  に対して、指数分布  $\mathfrak{Erl}_\lambda : \mathbb{N}_{\geq 0} \rightarrow [0, 1]$  は以下のように定義される。

$$\mathfrak{Erl}_{\lambda,k}(x) = \frac{\lambda^k \cdot x^{k-1} \cdot \exp(-\lambda \cdot x)}{(k-1)!}. \quad (\text{B.8})$$

ここで、 $\mathfrak{Erl}_\lambda$  の平均は  $k/\lambda$ 、分散は  $k/\lambda^2$  である。 ■

なお、二項分布と Poisson 分布の累積分布関数は、正規化不完全 beta 関数や不完全  $\Gamma$  関数などで表現でき、工学的に十分な精度の値を多項式時間で計算できる。

## B.4 統計的手法

統計的手法とは、解析の対象となる不均一な要素の集まりがもつ傾向や性質を数学的に検証する手法である。解析の対象となる要素の集まりを**母集団** (population) といい、その部分を**標本** (sample) という。実際的な問題では以下のような点が当てはまることが多く、直接的に母集団を解析することは困難である。

- 母集団が未知である。
- 母集団が既知であっても数学的解析が困難である。
- すべての標本を抽出して調べることは（実質的に）不可能である。
  - － 母集団のサイズが無限であるため、
  - － 標本の抽出に高いコストを要するため、など。

そのため、一部の標本のみを抽出し、その性質の傾向から母集団の性質を推定しなければならない。統計的手法では、**区間推定** (interval estimation) や **仮説検定** (hypothesis testing) などの確率論を基礎とした手法により、一部の標本集合から母集団の性質を推定する。

本節では、Bernoulli 試行の真の成功確率に関する区間推定と仮説検定の手法を簡単に紹介する。

### B.4.1 区間推定

区間推定法は、抽出する標本数  $n$  と **信頼水準**<sup>1</sup> (confidence level)  $y$  をあらかじめ決めた上で、母集団の統計量が含まれる確率が  $1 - y$  であるような区間を推定する手法である。

ここで、真の成功確率が  $p$  である Bernoulli 試行を考える。  $n$  個の標本を無作為に抽出した場合、成功標本数は二項分布  $\text{Binom}_{p,n}$  に従う。そのため、標本数  $n$  が大きく、  $np$  と  $np(1-p)$  が小さくない場合、この二項分布は正規分布  $\text{Norm}_{\mu,\sigma^2}$  で近似できる。そして、抽出した  $n$  個の標本のうち  $m$  個が成功していたと仮定し、  $x = m/n$  とする。このとき、  $z = \frac{x-\mu}{\sigma}$  が  $\text{Norm}_{0,1}$  に従う。よって、二項分布  $\text{Binom}_{p,n}$  が  $\text{Norm}(\mu, \sigma^2)$  で近似できる場合、

$$\hat{p} - k_{\frac{y}{2}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \leq p \leq \hat{p} + k_{\frac{y}{2}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \quad (\text{B.9})$$

<sup>1</sup>信頼係数 (confidence coefficient) ともいう。

が成り立つ確率は  $1 - y$  である。ただし、 $k_{\frac{y}{2}}$  は以下のような実数である。

$$\int_{k_{\frac{y}{2}}}^{\infty} \text{Norm}_{(0,1)}(x) dx = \frac{y}{2} \quad (\text{B.10})$$

例えば、 $y = 0.05$  のとき、 $k_{\frac{y}{2}}$  はおよそ 1.96 である。

## B.4.2 逐次確率比検定

**逐次確率比検定** (sequential probability ratio test, SPRT)[87] は Neyman-Pearson の補題 [66] を基に提案された逐次仮説検定である。逐次仮説検定では、検定に用いる標本集団の大きさはあらかじめ決めず、検定前に設定した水準で仮説を採択/棄却するに足る有意性を示すまで標本を無作為に逐次抽出 (生成) する。SPRT の場合、**第一種過誤** (タイプ I 誤り, 偽陽性, false positive) の発生率  $\alpha \in (0, 1)$ , **第二種過誤** (タイプ II 誤り, 偽陰性, false negative) の発生率  $\beta \in (0, 1)$  と **識別不能幅**  $2\delta > 0$  をあらかじめ設定する。SPRT の特徴として、この設定した水準のもとで最も検出力が高いことが Neyman-Pearson の補題で保証されている。

ここで、真の成功確率が  $p$  である Bernoulli 試行の真の確率が閾値  $p'$  より大きいかな否かを SPRT で推定する場合を考える。SPRT では帰無仮説  $H_0: p > p' + \delta$  と対立仮説  $H_1: p < p' - \delta$  を立て、どちらかを支持するに足る証拠が得られるまで標本を無作為に逐次抽出していく。もし仮説  $p = \theta$  が真であるならば、 $n$  個の標本を無作為抽出したときの成功標本数は二項分布  $\text{Binom}_{\theta,n}$  に従うはずであり、逆に言えば、抽出した  $n$  個の標本のうち  $m$  個の標本が成功しているとき、二項分布確率  $\text{Binom}_{\theta,n}(m)$  は仮説  $\hat{p} = \theta$  の**尤度** (likelihood) である。従って、標本集団  $\{x_1, \dots, x_n\}$  の下での  $H_0$  の  $H_1$  に対する尤度比  $R$  は、

$$R(\{x_1, \dots, x_n\}) = \frac{(p' + \delta)^m (1 - (p' + \delta))^{n-m}}{(p' - \delta)^m (1 - (p' - \delta))^{n-m}} \quad (\text{B.11})$$

ただし、 $m = |\{x_i \mid x_i \text{ は成功}\}|$ 。

もしこの尤度比が 1 より大きければその標本集団のもとで  $H_0$  のほうが  $H_1$  より尤もらしく、もし尤度比が 1 より小さければその標本集団のもとで  $H_1$  のほうが  $H_0$  より尤もらしい。SPRT の手続きは以下のように与えられる。

1.  $i := 1$ ,  $X := \emptyset$  とする。
2. 標本  $x_i$  を無作為抽出し、 $X := X \cup \{x_i\}$  とする。

3. 尤度比  $R(X)$  を計算し、以下の条件に従って次のアクションを決める.

- $(1 - \beta)/\alpha < R(X)$  のとき,  $H_0$  を採択して検定を終了.
- $R(X) < \beta/(1 - \alpha)$  のとき,  $H_1$  を採択して検定を終了.
- $\beta/(1 - \alpha) \leq R(X) \leq (1 - \beta)/\alpha$  のとき,  $i := i + 1$  として 2 へ.

結果として、この検定から得られる結果は以下のような性質を持つことを保証できる.

- もし仮説  $H_0$  が正しいならば、仮説  $H_0$  が採択される確率は少なくとも  $1 - \alpha$ ,
- もし仮説  $H_1$  が正しいならば、仮説  $H_0$  が採択される確率は大きくても  $\beta$

ただし、もし  $|p - p'| < \delta$  であるならば、すなわち、真の確率  $p$  が識別不能幅に含まれるならば、二つの仮説のどちらが有意であるかを過誤率  $\alpha, \beta$  の下での識別できることは保証されていない.

また、検定に必要と期待される標本集団のサイズの近似値  $N$  は、検査する確率制約値  $p$  と真の確率  $\hat{p}_\psi$  の差が識別不能幅に含まれているとき要求される標本集団サイズは急激に大きくなることが知られている [87, 91]. [91] では、検定前に設定した第一種過誤率  $\alpha$ , 第二種過誤率  $\beta$ , 識別不能幅  $2\delta$ , 確率閾値  $p$  に対して、最悪 ( $p$  と  $\hat{p}_\psi$  の差がほぼ 0 のとき) の場合で以下のような近似値を与えている.

$$N = \frac{-\log(\beta/(1 - \alpha)) \log((1 - \beta)/\alpha)}{\log((p + \delta)/(p - \delta)) \log((1 - p + \delta)/(1 - p - \delta))}. \quad (\text{B.12})$$



## 付 録 C      線形計画法

いくつかの線形不等式で与えられた制約の中で与えられた線形目的関数を最大化あるいは最小化する問題を**線形計画問題** (linear programming problem, LPP), また, それを解く手法を**線形計画法** (LP 法) と呼ぶ. 実際的な効率化問題の多くは LPP として記述できるため, LP 法は最もよく研究され, また, 広く用いられている最適化手法の一つである. 形式的には, LPP は以下のよう  
に定義される問題である.

**定義 C.1 (LPP)**  $\mathbf{x}$  を  $n$  列実変数ベクトル,  $\mathbf{b}$  と  $\mathbf{c}$  を  $m$  列実係数ベクトル,  $\mathbf{A}$  を  $m$  行  $n$  列実係数行列とする. LPP は, **目的関数** (objective function)  $\mathbf{c}^T \mathbf{x}$  を, 線形制約  $\mathbf{Ax} \preceq \mathbf{b}$  を満たす範囲で最大化/最小化する問題である. そして, 以下のように記述される

$$\begin{aligned} \text{maximize/minimize}_{\mathbf{x}} \quad &: \mathbf{c}^T \mathbf{x}, \\ \text{subject to} \quad &: \mathbf{Ax} \preceq \mathbf{b}. \end{aligned}$$

ただし,  $\mathbf{c}^T$  は  $\mathbf{c}$  の転置  $n$  行実係数ベクトル,  $\preceq \in \{\geq, \leq\}$  とする. ■

この問題を解く手法としては単体法や内点法などがあり, 多項式時間で計算可能である. そして, GLPK[49] や CPLEX[53] などの LP ソルバだけでなく, MATLAB[65] や Scilab[75] などの汎用数値計算ソフトウェアを用いても解くことができる.

### 線形制約充足問題

線形制約を満たす領域  $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{Ax} \preceq \mathbf{b}\}$  は**実行可能領域** (feasible region) と呼ばれる. そして, この領域が空であるか, すわなち, 線形制約が充足可能であるかを判定する問題を**線形制約充足問題** (linear constraint satisfaction problem, LCSP) と呼ぶ. LCSP についても LP ソルバで解くことができるが, Z3[93] や mathSAT 5[64] などの SMT (satisfiability modulo theories) ソルバを用いても解くことができる.

### 線形分数計画問題

目的関数を  $(\mathbf{c}^T \mathbf{x} + e)/(\mathbf{d}^T \mathbf{x} + f)$  ( $\mathbf{d}$  を  $n$  列実係数ベクトル,  $e$  を  $f$  実定数) とした問題を, **線形分数計画問題** (linear fractional programming problem, LFPP) と呼ぶ. ただし, 任意の実行可能解  $\mathbf{x}$  について  $\mathbf{d}^T \mathbf{x} + f > 0$  とする. LFPP は, 新たな  $n$  列実変数ベクトル  $\mathbf{y}$  と補助変数  $t$  を加え, 以下のような LPP に帰着して解くことができる.

$$\begin{aligned} \text{maximize/minimize}_{\mathbf{y}, t} & : \mathbf{c}^T \mathbf{y} + et, \\ \text{subject to} & : \mathbf{A}\mathbf{y} - \mathbf{b}t \simeq 0, \\ & \mathbf{d}^T \mathbf{y} + ft = 1, \\ & t \geq 0. \end{aligned}$$

ここで, 新たに導入した変数は,  $t = 1/(\mathbf{d}^T \mathbf{x} + f)$ ,  $\mathbf{y} = t\mathbf{x}$  を意図した変数である. 上記の LPP の解において  $t > 0$  であるとき, 最適値として  $\mathbf{c}^T \mathbf{y} + et$  ( $= \theta$ ) を得ることができる. ただし,  $t = 0$  である場合には,  $\mathbf{d}^T \mathbf{x} + f$  が  $\infty$  に発散する漸近最適解を表す. これは,  $\mathbf{c}^T \mathbf{y}$  に任意に近い値に近づけることを意味している. 漸近最適値  $\theta$  が得られているときには, 非漸近最適解は以下のような LPP に帰着して解くことができる.

$$\begin{aligned} \text{maximize}_{\mathbf{y}, t} & : t, \\ \text{subject to} & : \mathbf{A}\mathbf{y} - \mathbf{b}t \simeq 0, \\ & \mathbf{d}^T \mathbf{y} + ft = 1, \\ & \mathbf{c}^T \mathbf{y} + et = \theta, \\ & t \geq 0. \end{aligned}$$

そして, この LPP の  $t$  の最大値が 0 より大きいならば, その最適解はもとの LFPP の最適解である. そうでないならば, 非漸近最適解は存在しないため, 最大値/最小値は定義できない.