B-011

グラフ書換え言語 LMNtal への確率の導入
Introducing Probability into LMNtal, a Graph Rewriting Language

橋本 悠汰[1)]　上田 和紀[1)]
Yuta Hashimoto　Kazunori Ueda

1　はじめに

本研究は，グラフ書換えに基づくモデリング言語 LMNtal [1] を拡張し，確率的振る舞いを記述・検証可能にすることを目的とする．グラフ書換えは表現力が高く，対称性による状態数削減などの利点があり，モデリングに有用である．また，現実のシステムにはメッセージ損失などの確率的要素が含まれることがあり，そのようなモデルの検証には確率モデル検査が有用となる．既存の LMNtal とそのモデル検査器 SLIM [2] は非決定的な振る舞いに対応しているが，確率には未対応である．本研究では，LMNtal を拡張して確率的な挙動を記述し，離散時間マルコフ連鎖形式での出力を可能にする．さらに，確率モデル検査ツール PRISM [3] との連携により，確率的システムの容易な検証を目指す．

2　LMNtal

LMNtal は，階層グラフの書換えに基づくモデリング言語であり，システムの非決定的な振る舞いをグラフとその書換えによって表現できる．また，LMNtal で記述したモデルに対して，モデル検査を行うことでシステムの検証を行うことができる．LMNtal の構文や意味論の詳細は，文献 [1] を参照されたい．

2.1　LMNtal の構成要素

LMNtal グラフの主な構成要素はアトム，リンク，膜の 3 つである．アトムは LMNtal グラフにおけるノードで，リンクはアトム同士を 1 対 1 で接続するエッジである．また，膜は LMNtal グラフとルールの集まりを区別したり，ルールの適用範囲を制限するために使用される．アトムは識別子と，順序のついた 0 個以上の引数を持つ．この引数としてリンクを指定することにより，アトム間の接続関係が表現される．下記の LMNtal モデルは，アトム a と b，および b と c がリンクで接続されており，アトム c は膜に包まれていることを表す．

```
a(X), b(X, Y), {c(Y)}.
```

2.2　LMNtal の書換え規則

LMNtal における書換え規則はルールと呼ばれる．ルールによってグラフの書換えが行われることにより，計算が進行する．ルールは左辺と右辺からなり，LMNtal グラフにルールの左辺と同型な部分グラフ（すなわち，アトムやリンクの接続関係が一致する部分）が存在する場合に，その部分を右辺で置き換える．

グラフとルールの集まりはプロセスと呼ばれる．下記の LMNtal モデルはプロセスの例を表している．初期状態ではアトム a と b がリンクで接続されており，アトム b は膜に包まれている．@@ はルール名を示す記号であり，下の例ではルールに b2c という名前が付いている．ルール b2c の左辺は初期状態の部分グラフと同型であるため，初期状態のアトム b はアトム c に書き換えられる．

```
a(X), {b(X)}.
b2c@@ {b(L)} :- {c(L)}.
```

2.3　LMNtal 処理系 SLIM

コンパイラによって中間命令列に変換された LMNtal モデルは，LMNtal の抽象機械である SLIM (Slim LMNtal IMprementation) によって実行される．SLIM には通常実行とモデル検査の 2 つのモードがある．通常実行モードでは，計算が停止するまでルールを繰り返し適用する．モデル検査モードでは初期状態からルールの適用によって到達可能な状態を探索することによって，状態空間を構築する．モデル検査モードは並列実行が可能で，効率的な状態空間構築が行える．これらの結果は LMNtal 統合開発環境 LaViT [4] の可視化ツールを用いてグラフィカルに表示することができる．現在の SLIM は LMNtal モデルに対して LTL (Linear Temporal Logic) で記述した性質をモデル検査することができる．一方で，確率モデル検査はサポートされていない．

3　確率モデル検査

確率モデル検査は，確率的な振る舞いを含むシステムの検証手法の一つである．確率的なシステムは，離散時間マルコフ過程 (DTMC) やマルコフ決定過程 (MDP) などのモデルを用いて表現される．本節の内容は Baier と Katoen の著書 [5] を参考にしている．

3.1　離散時間マルコフ過程 (DTMC)

DTMC の定義を定義 1 に示す．DTMC は確率的な振る舞いを含む離散状態のシステムのモデル化に適しており，パケットが一定の確率で損失される信頼性が絶対的ではない通信プロトコルや，乱択アルゴリズムの表現に利用される．

定義 1

DTMC は三つ組 (S, s_0, P) から構成される．

- S：状態の有限集合
- $s_0 \in S$：初期状態
- $P : S \times S \to [0,1]$：各状態 s において，次の状態 s' への遷移確率を与える関数．

3.2　マルコフ決定過程 (MDP)

MDP の定義を定義 2 に示す．MDP は，DTMC を拡張したモデルであり，非決定的な選択と確率的な振る舞いの両方を扱うことができる．

MDP では，状態 s において，どのアクション a を選

1）早稲田大学 Waseda University.

択するかは非決定的であり，選択されたアクションに基づいて次の状態への遷移が確率的に行われる．MDP では,「どのアクションを選択すべきか」という問いに対して，最適な戦略を見つけることが可能である．例えば，ある目的状態へ到達する確率が最大となる戦略の設計・評価を行うことができる．

定義 2

MDP は四つ組 (S, s_0, A, P) から構成される．

- S：状態の有限集合
- $s_0 \in S$：初期状態
- A：アクションの有限集合
- P：$S \times A \times S \to [0,1]$：各状態 s において，アクション a を選択した場合に次の状態 s' へ遷移する確率を与える関数．

3.3　確率モデル検査

確率モデル検査は，確率的な動作を含むシステムに対して，PCTL (Probabilistic Computation Tree Logic) 式などで記述された形式的な仕様が満たされるかどうかを検証する手法である．この分野ではいくつか高機能なツールが開発されており，その中でも特に広く利用されているのが PRISM である．PRISM は DTMC，MDP, 連続時間マルコフ過程（CTMC）などのモデルをサポートしている．モデルの記述には独自の PRISM 言語のほか，遷移行列を直接入力として与えることも可能である．仕様記述言語として時相論理に基づく PRISM 特性仕様言語 (PRISM property specification language) を使用することができる．また，報酬に基づく性質の分析もサポートしており，モデルの各状態に報酬を設定することで，期待値などの計算が可能である．

4　LMNtal への確率の導入

本節では，グラフ書換え言語 LMNtal において，確率的な振る舞いを取り扱うための拡張について述べる．特に，ルールへの重みの導入，およびそれにより確率的な動作を含むシステムの記述を可能とする方法について説明する．

4.1　ルールへの重みの付与と DTMC の表現

LMNtal の書換えは通常，適用可能なルールのうち，いずれかが非決定的に選択されることで状態が遷移する．この遷移に確率的な要素を与えるために，本研究では各ルールに対して「重み」を付与する手法を提案する．この手法については Bournez と Hoyrup の手法 [6] を参考にしている．重みは，適用可能なルールの中から，どのルールが選択されるかの確率を決定するために用いられ，次のように定義される．ここで，LMNtal グラフにおいて部分グラフがルールの左辺と同型であることを「マッチする」と呼ぶ．また，ルールがマッチする部分グラフの個数を「マッチ数」と呼ぶ．

- 各ルール r に対して，重み w_r を定義する．
- 状態 s から遷移可能な状態の集合を S とする．
- 状態 s から状態 $s_i \in S$ への遷移に対応するルール，およびそのルールのマッチ数の組 (r, c) の集合を T_i とする．
- 状態 s から状態 s_i への遷移確率を表す関数 P を次のように定義する：

$$P(s, s_i) = \frac{\sum_{(r,c)\in T_i} w_r c}{\sum_{s_j \in S} \sum_{(r,c)\in T_j} w_r c}$$

LMNtal モデルにおいて各遷移に厳密な確率を直接指定することは難しいが，上記の定義により LMNtal のルールに付与された重みが正規化されることにより，確率的な状態遷移を表現できるようになる．ルールへ重みを付加した LMNtal モデルは次の構成を持つ：

- 状態集合 S：LMNtal モデルの状態空間に含まれる全ての状態の集合.
- 初期状態 s_0：LMNtal モデルの初期状態.
- 遷移確率関数 P：各状態 s について，次の状態 s' への遷移確率を与える関数.

この構成により，ルールに重みを付加した LMNtal モデルは DTMC として解釈される．

4.2　MDP の表現

前節では，LMNtal のルールに重みを付与することで，確率的な状態遷移を表現できることを述べた．本節では，こうした重み付きルールを用いることで，確率と非決定性の両方を含む MDP を表現する方法について述べる．MDP を表現するための LMNtal モデルにおいて，任意のルールは以下の 2 つのいずれかに分類される：

- アクションルール：非決定性を表現するためのルールであり，ある状態に対して 1 つ以上の可能なアクションの中から 1 つを非決定的に選択する.
- 確率ルール：アクションの選択後，そのアクションに対応する遷移確率分布に基づいて状態を遷移させるルールである.

確率ルールは DTMC と同様に重みを付与し，正規化を行うことによって，状態遷移の確率を定義する．

アクションルールと確率ルールを区別して適用させるために，状態内にアクションの状態を保持する `action` アトムを用いた手法を提案する．まず，アクションルールでは `action(none)` という構造があるときに，`none` アトムをアクションに対応するアトム（ここでは `act` とする）に置き換えることで，アクションの選択を表現する．確率ルールでは，左辺に `action(act)` を含めることで，アクションの選択後に対応する確率的な遷移を表現する．以上の方法により，アクションルールと確率ルールを組で適用されるように制御することができ，MDP と対応する状態遷移系を構築できる．具体例については，6 節の例題を参照されたい．

5　LMNtal 拡張の実装

確率的なモデルの記述とその検証のため，LMNtal モデルの SLIM 実行結果とルールの重みをもとに，PRISM が扱える DTMC または MDP の形式へ変換するトランスレータを実装した．これにより，LMNtal によって記述されたシステムの確率的な挙動を，形式手法を用いて

解析・検証することが可能となる．

本トランスレータは，SLIM による LMNtal モデルの実行結果と，ルールに付与された重み情報を入力として受け取り，PRISM に対応した Explicit Model 形式のテキストファイルを生成する．SLIM の出力には，状態空間および遷移に関する詳細な情報が含まれており，特に以下の項目がトランスレータの入力として使用される：

- 各状態において遷移可能な状態の集合
- ある状態から遷移可能な各状態へ遷移するために適用されるルールとマッチする部分グラフの個数
- 各ルールに付与された重み

これらの情報に基づき，トランスレータは各状態における遷移確率の正規化を行い，PRISM の DTMC/MDP モデル記述形式に変換する．変換後のモデルに加え，各状態へのラベル付け情報および検証対象となる PCTL 式等を PRISM に与えることで，確率的な振る舞いを含むシステムの性質の検証を行うことができる．

LMNtal を用いた確率モデル検査の全体的な流れを図 1 に示す．これは，LMNtal モデルの記述から SLIM による状態空間探索，トランスレータによるモデル変換，そして PRISM による検証までの一連のパイプラインを表している．

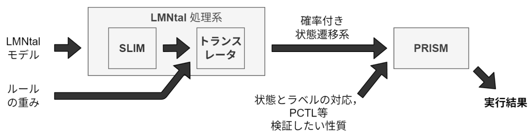

図 1　LMNtal を用いた確率モデル検査の全体像

6　例題

6.1　コイントスによるサイコロの再現

コイントスでサイコロを再現するモデル [7] を題材に，LMNtal による DTMC の表現例を示す．このモデルでは表と裏が出る確率が等しいコインを用いて，1〜6 までの値が等しい確率で出るサイコロを再現する．

ソースコード 1 に LMNtal モデルを示す．このモデルでは，初期状態として空のリスト `coin` と，コイントスの回数を表す `toss` を持つ．`head` と `tail` の 2 つのルールがそれぞれコインの表と裏を表し，コイントスの結果をリストの先頭に追加する．下の 2 つのルールは 3 回連続で同じ結果が出た場合に，2 回目のコイントスからやり直すためのルールである．また，各ルールの重みはそれぞれ 1 を与える．

ソースコード 1　emulate-one-die.lmn

```
coin=[], toss, toss, toss.   // initial state
head@@ coin=H, toss :- coin=[h|H].
tail@@ coin=H, toss :- coin=[t|H].
head2@@ coin=[h,h,h] :- coin=[h], toss, toss.
tail2@@ coin=[t,t,t] :- coin=[t], toss, toss.
```

このモデルによって構築される DTMC を図 2 に示す．各遷移のラベルはルール名を表し，実数値はその遷移の確率を表す．また，最も右側の各状態はサイコロの 1 から 6 の値に対応する．

変換後の DTMC，検証したい性質，各状態とラベルの対応等を PRISM に与えることで，本モデルの確率

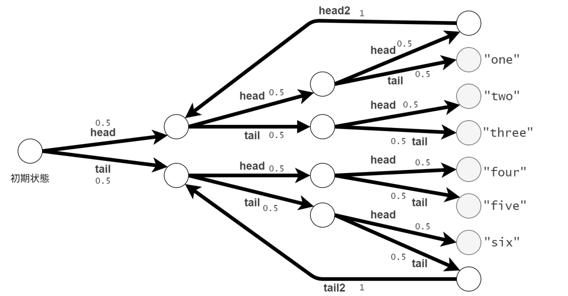

図 2　サイコロ再現の LMNtal モデルから構築された DTMC

的な性質について検証できる．例えば，初期状態から `head`, `head`, `tail` の順に辿った状態とラベル `"one"` の対応付けを PRISM に与えると，その状態に到達する確率を求めることができる．`P=? [F "one"]` は当該の状態に到達する確率を求める PRISM 特性仕様言語の記述例である．これに対する PRISM の実行結果として，0.166666... ≃ 1/6 という値が得られた．

また，サイコロの出目に相当する状態とラベル `"one"`，`"two"`，`"three"`，...の対応，およびコイントス前の各状態へ報酬 1 を与える設定を PRISM に与えることで，次の PRISM 特性仕様により初期状態からサイコロの値が確定するまでに必要なコイントスの回数の期待値を求めることができ，結果として `3.66666...` の値が得られた．

```
R=? [F ("one"|"two"|"three"|"four"|"five"|"six")]
```

6.2　複数人が手を繋ぐモデル

複数の人が左右の手を繋ぐモデルを題材に，ルールのマッチ数が複数となる LMNtal モデルを示す．ソースコード 2 に LMNtal モデルを示す．`h(l,r)` の `h` は 1 人の人を表し，`l` と `r` はそれぞれ左手と右手を表す．ルール `hold` により，自分自身，または他の人同士の左手と右手を繋ぐことができる．ルール `hold` には重み 1 を与える．

このモデルによって構築される DTMC を図 3 に示す．初期状態の LMNtal グラフにおいて，ルール `hold` について，自分自身の左右の手を繋ぐマッチ数は 3 つ，他の人同士の左右の手を繋ぐマッチ数は 6 つある．そのため，状態 `"s1"` への遷移確率は 0.333333 ≃ 1/3，状態 `"s2"` への遷移確率は 0.666666 ≃ 2/3 となる．

ソースコード 2　handshake.lmn

```
h(l,r), h(l,r), h(l,r).
hold@@ l(X),r(Y) :- lr(X,Y).
```

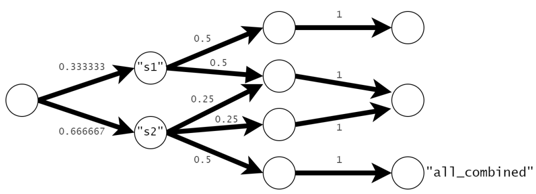

図 3　手を繋ぐモデルから構築された DTMC

このモデルについては，以下の PRISM 特性仕様言語により，最終的に全員が手を繋ぎ大きな輪を形成する状

態に到達する確率を求めることができる．結果として 0.333333... ≃ 1/3 という値が得られた．

```
P=? [F "all_combined"]
```

人数が 4 人，5 人の場合についても，同様に全員が手を繋ぎ大きな輪を形成する状態に到達する確率を求めた結果，それぞれ 0.249999... ≃ 1/4，0.199999... ≃ 1/5 という値が得られた．

6.3 簡易 WSN モデル

データ送信に失敗する可能性を持つ簡易な無線センサネットワーク（Wireless Sensor Network, WSN）モデル [8] を題材に，LMNtal による MDP の表現方法を具体的に示す．本モデルは，送信元ノード（アトム `source` がある膜）と受信先ノード（アトム `dest` がある膜）から構成され，送信元が保有するデータをリンクを介して宛先に送信しようとするシナリオを想定している．ソースコード 3 に LMNtal モデルを示す．

初期状態では，送信元ノードがアトム `data` を持ち，アトム `p` を接続するリンクを介して受信先ノードと接続されている．さらに，状態を制御するためのアトム `action(none)` が存在し，`action_send` か `action_wait` のいずれかのアクションが選択される．ここで，`action_send` は送信元ノードがデータを受信先ノードに送信するアクション，`action_wait` は送信元ノードがデータの送信を待機するアクションである．

`action_send` が選択されると，`action(none)` が `action(send)` に書き換えられる．その後 `send_succ` または `send_fail` のいずれかが，それぞれの重み w_{succ}，w_{fail} に基づいて確率的に選択される．このモデルによって構築される MDP を図 4 に示す．白い遷移はアクションルールによる遷移を表し，黒い遷移は確率ルールによる遷移を表す．各状態は，送信の成否やデータの位置によって異なる LMNtal グラフ構造を持ち，アクションの選択とそれに続く確率的な遷移により次の状態へと遷移する．

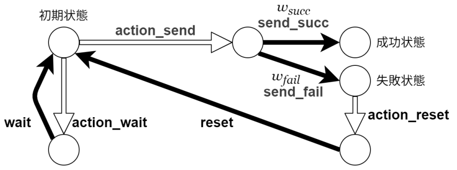

図 4 簡易 WSN モデルの LMNtal モデルから構築された MDP

ソースコード 3 simple-wsn.lmn

```
action(none), {source, data, p(L)}, {dest, p(L)}.

action_send@@
{source, data, p(L)}, {dest, p(L)} \
action(none) :- action(send).

send_succ@@
action(send),{source, data, p(L)},{dest, p(L)} :-
action(none),{source, p(L)},{dest, data, p(L)}.

send_fail@@
action(send),{source, data, p(L)},{dest, p(L)} :-
action(none),{source, data, failed, p(L)},{dest, p(L)}.

action_wait@@
{source, data, p(L)}, {dest, p(L)} \
action(none) :- action(wait).

wait@@
{source, data, p(L)}, {dest, p(L)} \
action(wait) :- action(none).
...
```

7 関連研究

グラフ書換えに基づくモデリング言語へ確率的要素を導入する研究として，Archibald らによる Probabilistic Bigraphs [8] が挙げられる．これは，Bigraph による DTMC，CTMC，MDP の表現をサポートしており，拡張においてルールへ重みを付与するアプローチが採用されている．Bigraphs は place graph と link graph を主な構成要素とする一方で，LMNtal は膜，ハイパーリンクといった構造も持つため，複雑な構造の表現において LMNtal が有用となる場合が考えられる．

8 まとめと今後の課題

本研究では，グラフ書換え言語 LMNtal に確率を導入し，確率モデル検査ツール PRISM と連携することで，LMNtal を用いて確率的な振る舞いを含むシステムのモデル化し，性質の検証を行う手法を提案した．LMNtal モデルの遷移に確率の値を直接指定することが難しい問題を，ルールへの重みの付与と正規化により解決し，DTMC および MDP の表現を可能にした．この拡張により，システムの進行によってネットワークの接続関係が変化する確率モデルのように，PRISM モデルでは表現が難しいシステムのモデル化が LMNtal を用いて容易に行えることが期待される．

今後は，トランスレータにラベル付け支援などの機能を追加し，LMNtal 統合開発環境 LaViT への統合を目指す．最終的には LaViT 上で確率モデル検査を容易に実行できるインターフェースを提供することが目標である．また，LMNtal の構文を拡張し，確率やアクションの表現を可能にすることが望ましい．さらに，将来的には DTMC や MDP に加え，CTMC や確率時間オートマトン（PTA）といった連続時間を扱うモデルへの対応も検討する．

本研究の一部は，科学研究費補助金 23K11057 の援助を得て実施した．

参考文献

[1] Ueda, K.: LMNtal as a hierarchical logic programming language, Theoretical Computer Science, Vol. 410 (2009), pp. 4784-4800.

[2] 後町 将人, 堀 泰祐, 上田 和紀: LMNtal 実行時処理系の並列モデル検査器への発展, コンピュータソフトウェア, Vol. 28 (2011), pp. 4_137-4_157.

[3] Kwiatkowska, M., Norman, G. and Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems, CAV 2011, LNCS 6806, Springer Berlin Heidelberg, pp. 585-591.

[4] 綾野 貴之, 堀 泰祐, 岩澤 宏希, 小川 誠司, 上田 和紀: 統合開発環境による LMNtal モデル検査, コンピュータソフトウェア, Vol. 27 (2010), pp. 4_197-4_214.

[5] Baier, C. and Katoen, J. P.: Principles of model checking, MIT Press, 2008.

[6] Bournez, O. and Hoyrup, M.: Rewriting logic and probabilities, RTA 2003, LNCS 2706, Springer Berlin Heidelberg, pp. 61-75.

[7] Knuth, D. and Yao, A.: The complexity of nonuniform random number generation. In: Algorithms and Complexity, New Directions and Results, Academic Press, 1976, pp.357-428.

[8] Archibald, B., Calder, M. and Sevegnani, M: Probabilistic bigraphs, Formal Aspects of Computing, Vol. 34 (2022), No. 2, pp. 1-27.