

CPS-IoTゼミ 発表資料

確率的モデル検査ツール PRISMの検証

2017/08/31

種村 嘉高
猿渡 真之介
関司 達

まえがき

CPS/IoTソリューションではセンサーから膨大で多様データを収集/解析する必要があるため、IoT固有の非機能要件(例：セキュリティ、転送速度、データ処理速度)を満たす必要がある。

これらはシステム開発初期の段階からパフォーマンスを事前検証できることが望ましいが、時間面/費用面の制約から難しい場合が多い。

確率モデル検査ツールは時間要素と確率要素を含めたモデルの定量的評価を行う事が出来るため、CPS/IoTソリューションのパフォーマンス検証に有効と考えられる。

例として簡易なネットワークモデルを題材とし、確率モデル検査ツールの有効性を検証する。

目的

確率モデル検査ツールとしてPRISMを選定した。
PRISMを使って、以下の検証を実施する事を目的とする。

1.PRISMの有効性

- 今回の例題に対する、確率モデルの定量的評価の有効性

2.PRISMの使い方

- 確率モデル検査ツールとしての利用方法、ノウハウ

背景

PRISMの選定理由

- 確率モデルを扱うための代表的なツールであるため

その他のモデル検査器として

- ・ Process Analysis Toolkit
- ・ Uppaal

を検討したが、確率モデルを扱うことが出来なかったため、採用を見送った。

PRISMとは

PRISMはランダムまたは確率的な挙動を示すシステムのモデリングと分析のための確率モデル検査ツールである。

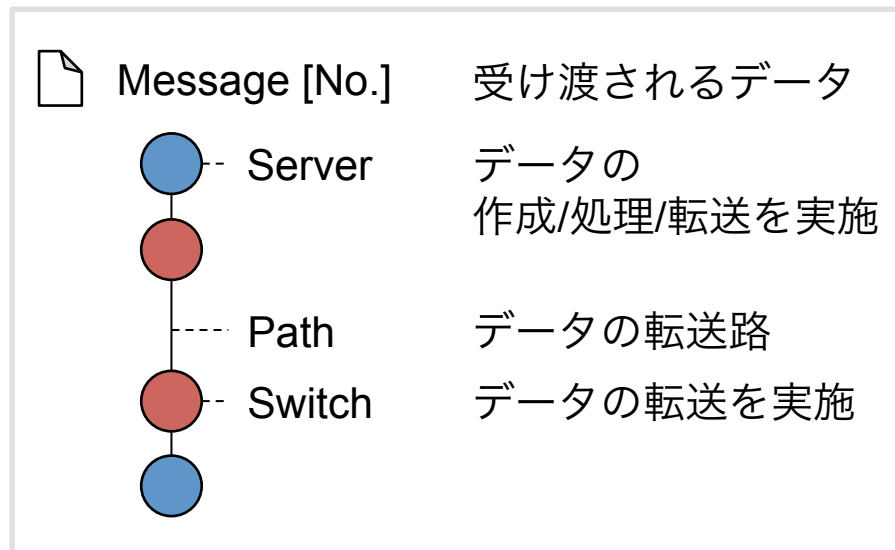
PRISMはいくつかのタイプの確率モデルを構築し、論理式が真となる確率を検査できる。

状態遷移系として、以下3種類のマルコフ連鎖を対象とすることができる。

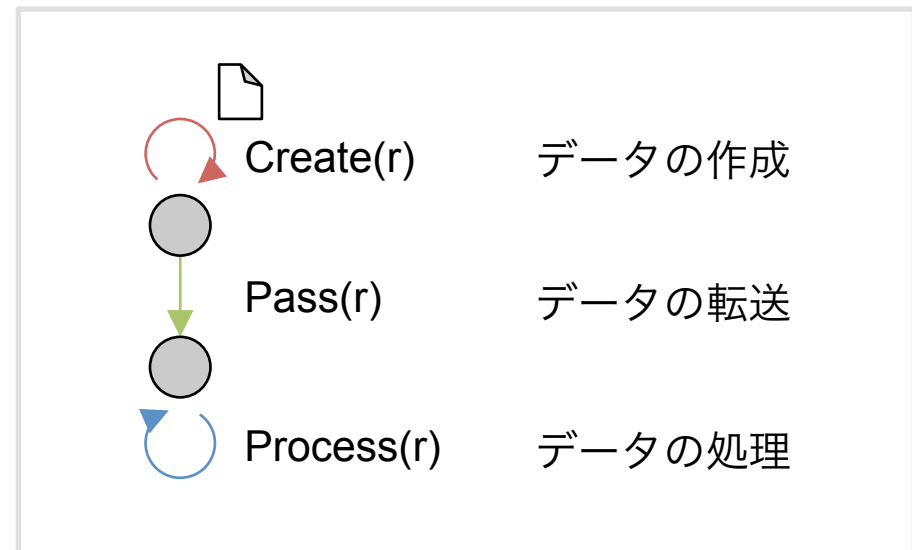
- 離散時間マルコフ連鎖 (DTMC)
- 連続時間マルコフ連鎖 (CTMC)
- マルコフ決定プロセス (MDP)

作成モデル(1/2)

- モデルの「登場人物」と「イベント」を以下のように定義



登場人物



イベント ()内は発生確率

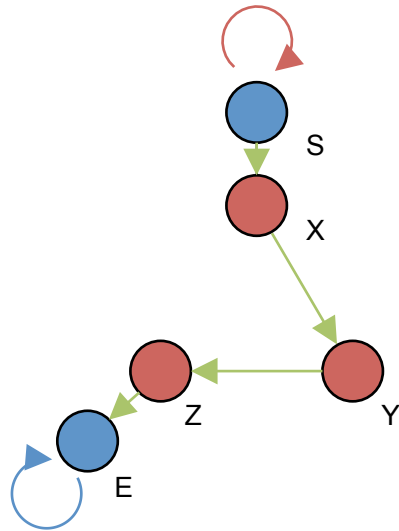
作成モデル(2/2)

その他制約事項

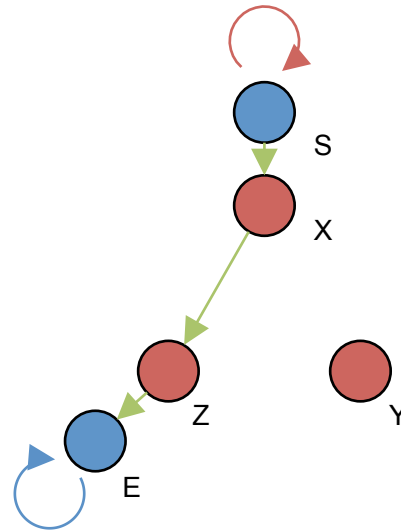
- メッセージの生成・転送・処理は非同期に規定確率に基いて実行される
- メッセージはネットワーク中複数存在することが可能
- メッセージが作成されてから処理されるまでの間は必ずいずれかのサーバ、もしくはスイッチ上に存在する
- サーバ、スイッチは一度に一つのメッセージしか保有できない
 - メッセージ生成時や転送時に既にメッセージが存在する場合は

評価環境

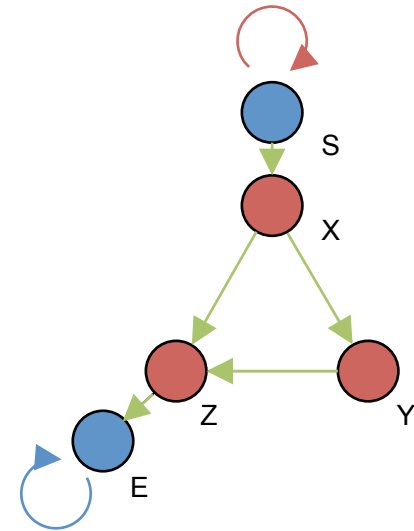
- 現実的に起こりうるネットワークのトポロジを想定して、以下の**3パターン**でモデル化。(Case3が最速になると想定)



Case1:STP
シングルパス(非最短経路)



Case2:STP
シングルパス(最短経路)



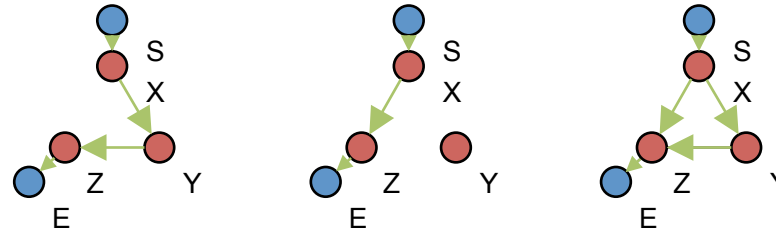
Case3:SDN
マルチパス

評価環境

- 前頁3パターンでS-X,Z-E間でPassが発生する確率を変化させ遅延を評価
 - X-Y,Y-Z,X-Z間でのPassの発生確率を**5**(回/秒)
 - SでのCreate、EでのProcessの発生確率はともに**100**(回/秒)
 - メッセージを10個作成・処理されるまでに要した時間の期待値を評価
(評価時間は30秒)

実験結果

スイッチ間転送 **5**(回/秒)
サーバ処理 **100**(回/秒)



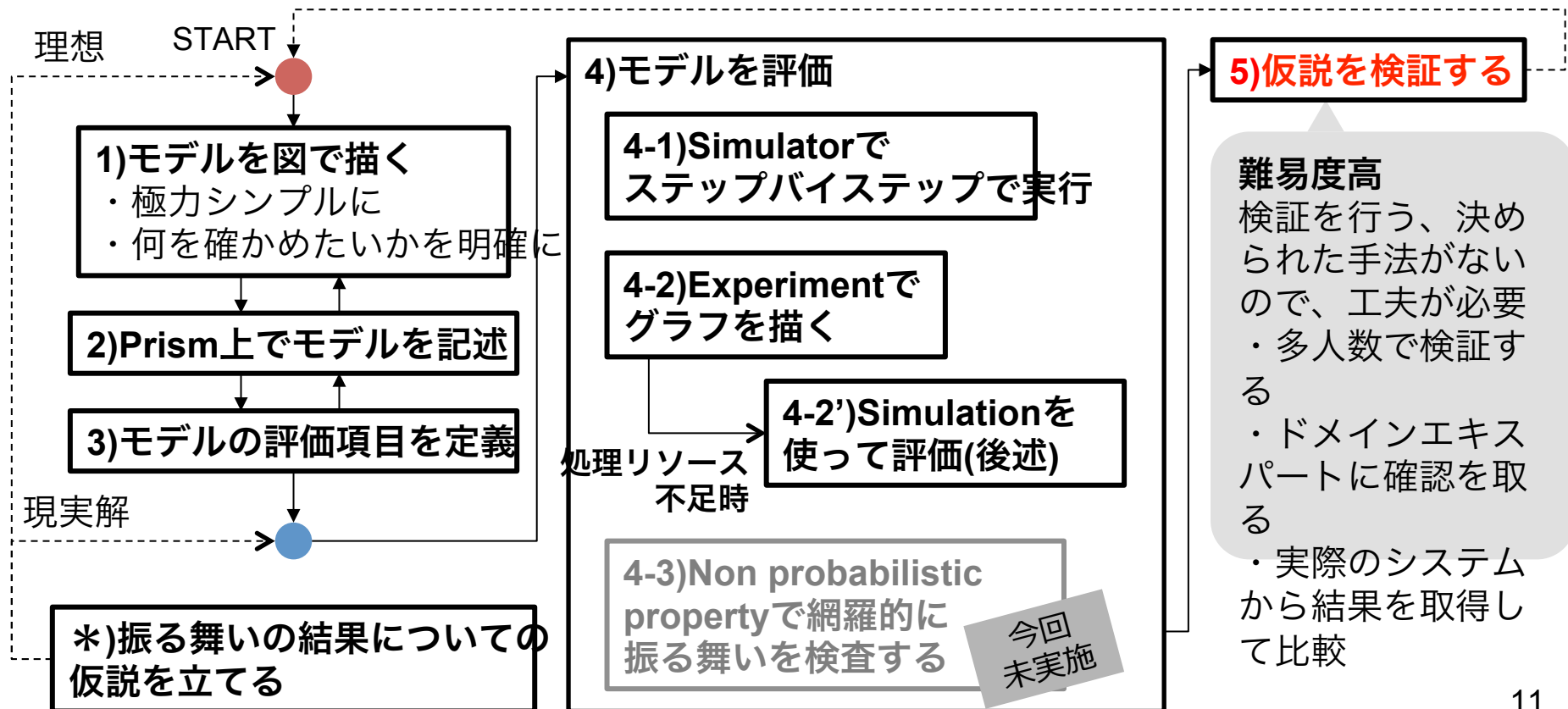
S-X Z-E間 転送	Case1	Case2	Case3
50(回/秒)	9.24±0.18	4.41±0.10	3.75±0.08
5(回/秒)	13.18±0.21	9.40±0.17	9.33±0.18

10メッセージの総滞在時間

- ・ より効率的な転送パス(1<2<3)を活用することで生まれる速度差を**定量的に**評価
- ・ 効率的な転送パスを採用しても他にボトルネックがある場合(Case3のZ-E間)に大きな差が生まれないケースが存在する事も確認

PRISMのベストプラクティス

戻る



モデルの記述(1/3)

- モデルの記述の仕方

- モデルのモジュールの取り方で記述しやすさが異なった

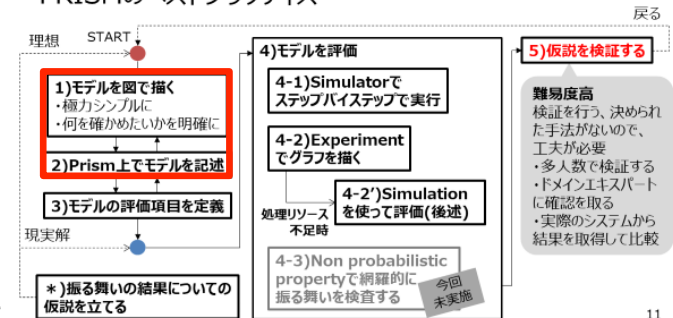
- i. スイッチを基本のモジュールとする

- テンプレート
→スイッチを追加しやすい
- 評価のために各データのモジュールが必要
→送るデータを増やす際の記述量が多い

- ii. データの状態(位置情報など)を基本のモジュールとする

- データをID番号で管理
→ID番号をスケールリングすることで送るデータの数を簡単に変更できる
(今回はii.が重要)

PRISMのベストプラクティス



11

モデルの記述(2/3)

- モデルの記述の仕方
モジュールのテンプレート

PRISMにはモジュールの記述を置換する記述法がある

→モジュールのテンプレートを作成、テンプレートの変数を具体的な変数
に置換することで容易にモジュールを追加

テンプレート

```

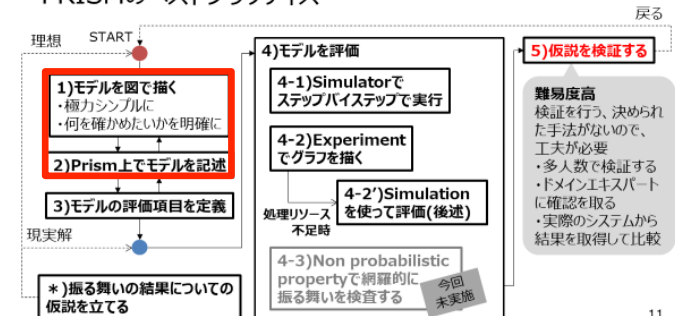
138 module switchTemplate
139   data0 : [0..1000] init 0;
140
141   // receive data: success case
142   [passA0] (dataA>0) & (data0=0) -> (data0'=dataA);
143   // receive data: failure case
144   [passA0] (dataA>0) & (data0>0) -> true;
145
146   // receive data: success case
147   [passB0] (dataB>0) & (data0=0) -> (data0'=dataB);
148   // receive data: failure case
149   [passB0] (dataB>0) & (data0>0) -> true;
150
151
152   // send data: success case
153   [pass0L] (data0>0) & (dataL=0) -> passrateL : (data0'=0);
154   // send data: failure case
155   [pass0L] (data0>0) & (dataL>0) -> passrateL : true;
156
157   // send data: success case
158   [pass0R] (data0>0) & (dataR=0) -> passrateR : (data0'=0);
159   // send data: failure case
160   [pass0R] (data0>0) & (dataR>0) -> passrateR : true;
161
162
163 endmodule
  
```

テンプレートを利用したモジュール

```

166 module switchX = switchTemplate[data0=dataX,
167   dataA=dataS,
168   dataB=dataB,
169   dataR=dataZ,
170   dataL=dataY,
171   passA0=passSX,
172   passB0=passBX,
173   pass0R=passXZ,
174   pass0L=passXY
175 ]
176
177 endmodule
178
179 module switchY = switchTemplate[data0=dataY,
180   dataA=dataX,
181   dataB=dataB,
182   dataR=dataZ,
183   dataL=dataL,
184   passA0=passXY,
185   passB0=passBY,
186   pass0R=passYZ,
187   pass0L=passYL
188 ]
189
190 endmodule
191
  
```

PRISMのベストプラクティス



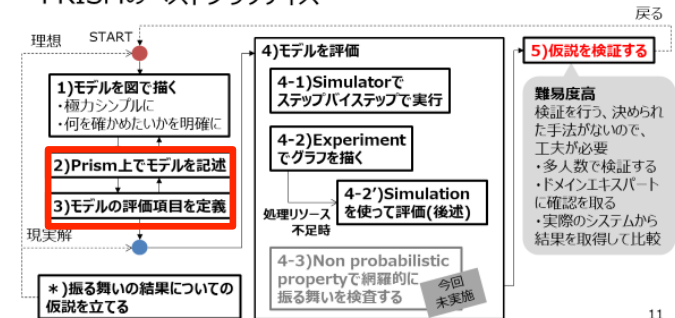
モデルの記述(3/3)

- モデルの記述の仕方
監視モジュールの設置

時間の計測には微小時間ごとにシステムの状態を確認する監視モジュールを用意、計測対象の状態なら微小時間を返す
監視モジュールが返した時間の累積をシステムがその状態であった時間とする



PRISMのベストプラクティス



11

モデルの評価(1/2)

- モデルのバリデーション

方式として、ExperimentとSimulationがあり、それぞれの意味は

Experiment：確率的に解を求める

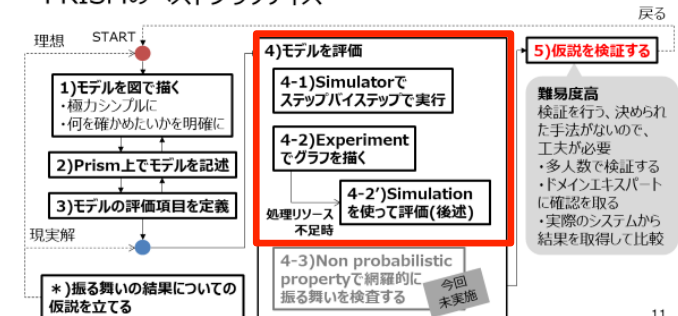
Simulation：統計的に解を求める

である。計算量（処理時間）、正確性のトレードオフになっている。

	計算量（処理時間）	正確性
Experiment	△(大きい)	◎(高い)
Simulation	◎(小さい)	△(低い)

Experimentを実行した際にPRISMが固まる、落ちることがあり、その場合はSimulationで代用した。

PRISMのベストプラクティス



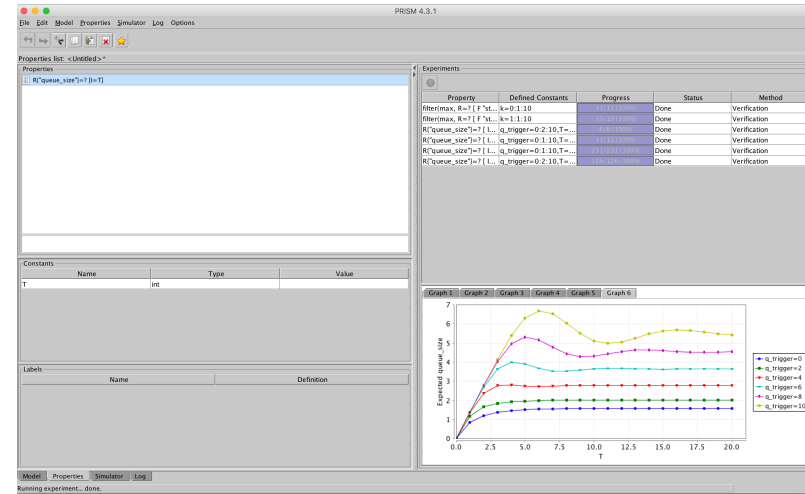
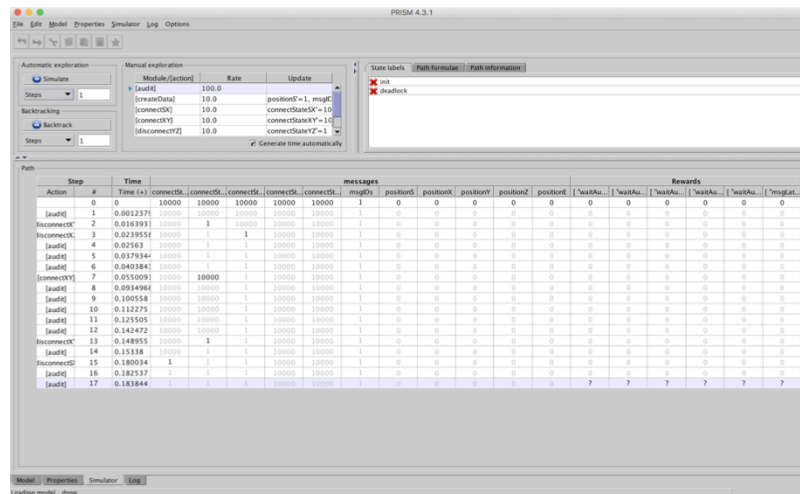
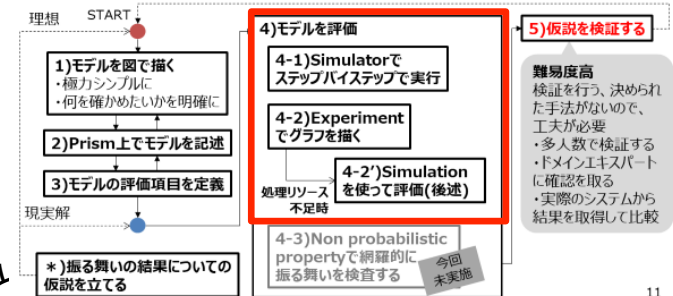
11

モデルの評価(2/2)

● モデルのバリデーション

モデルの変更後や実験結果の妥当性が疑われると
→シミュレータで意図しない挙動がないか、実験のパラメータを変化させて
グラフの形状の特徴や変化を確認する

PRISMのベストプラクティス



結論

1.PRISMの有効性

現実の課題をモデル化して広範囲な条件下で確率モデルの定量的評価を行い、仮説を検証する事が出来た。

2.PRISMの使い方

PRISMを使った確率モデルの定量的評価の手順を確立する事が出来た。

マルコフ過程/マルコフ連鎖

マルコフ過程 (Markov process)

未来の挙動が現在の値だけで決定され、過去の挙動と無関係であるという性質 (=マルコフ性)を持つ確率過程を指す。

マルコフ連鎖 (Markov chain)

確率過程の一種であるマルコフ過程のうち、とりうる状態が離散的（有限または可算）なもの（離散状態マルコフ過程）を指す。

離散時間マルコフ連鎖

離散時間マルコフ連鎖 (Discrete-time Markov Chain)

時間が離散的に推移するマルコフ連鎖を指す。

連続時間マルコフ連鎖

連続時間マルコフ連鎖 (Continuous-time Markov Chain)

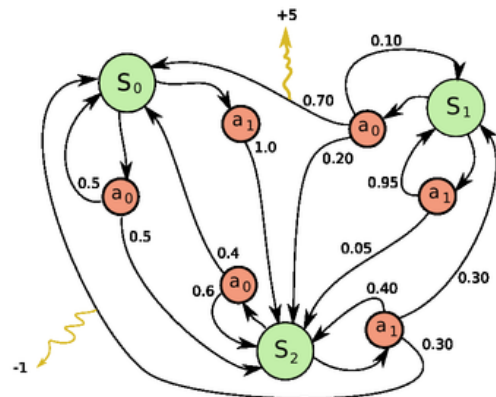
時間が連続的に推移するマルコフ連鎖を指す。

マルコフ決定過程

マルコフ決定過程 (Markov Decision Process)

状態遷移が確率的に生じる動的システム（確率システム）の確率モデルであり、状態遷移がマルコフ性を満たすもの指す。

マルコフ連鎖に（選択可能な）行動、および（行動を計画する動機を与える）報酬を追加し拡張したものである。



3つの状態と2つの行動をもつ簡単な MDP の例