組み込みソフトへの数理的アプローチ

第8回

時相論理式を見る
── LTSA で時相論理式をステート・マシンに変換する

藤倉 俊幸

1 はじめに

前回（2007年7月号，pp.174-177）は，Strong Until と Weak Until の違いを LTSA によって観察しようとしたところまで説明した．今回はこの続きで，LTSA を使って LTL 式を目に見える形に変換して観察することにする．

モデル検査ツールは，検査対象をステート・マシンで記述してそれが「ある性質」を満たすかどうか検査する（**図1**）．このとき，検査したい「ある性質」は，検査する人が時相論理式で記述する．検査ツールは与えられたその時相論理式をある種のステート・マシンに変換して，検査対象のステート・マシンと合成して検査を行う．「ある種のステート・マシン」に変換することで時相論理式を見られるようになる．ここで言う「ある性質」というのは，こう動いて欲しいという表現であり，時相論理式で表現するよりはフローチャートやステート・マシンで表した方がプログラムを読める人には分かりやすい．

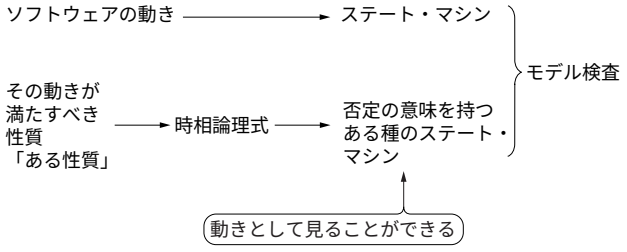

図1　時相論理式を見る

2 Büchi automata

LTSA の場合は時相論理式を Büchi ステート・マシン（Büchi

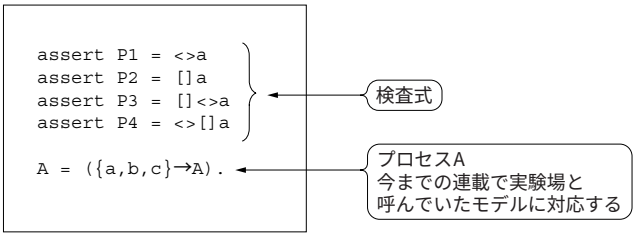

図2　Büchi オートマトンを生成するための LTSA モデル

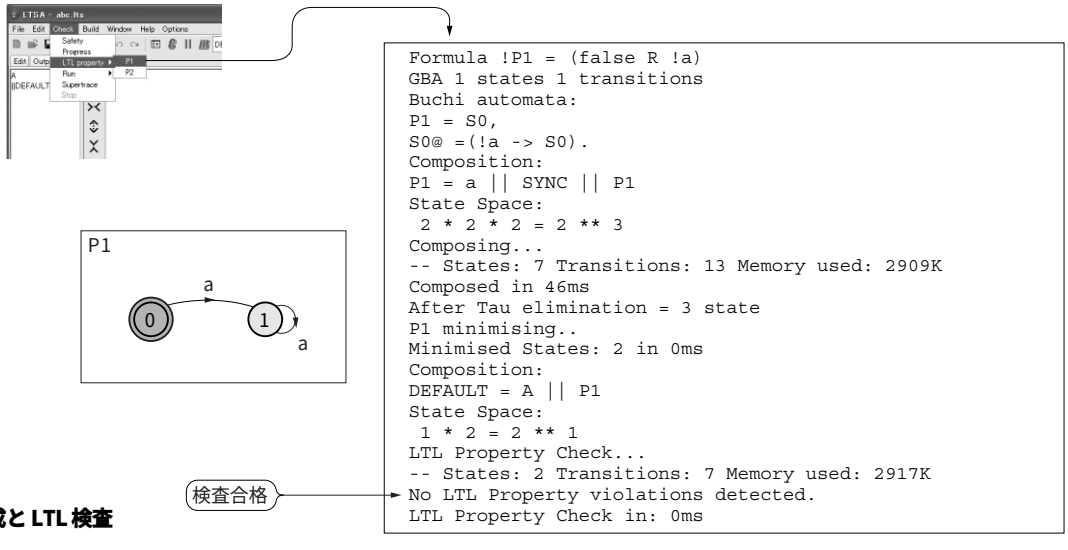

図3
Büchi オートマトン生成と LTL 検査

KEYWORD ──LTSA, NuSMV, Büchi automata, Strong Until, Weak Until

```
MODULE main
IVAR
  a: boolean;
  b: boolean;
  c: boolean;

LTLSPEC G a
LTLSPEC F a
```

```
MODULE main
IVAR
  act : {a, b, c};

LTLSPEC G (act=a)
LTLSPEC F (act=a)
```

図4　NuSMVの場合

automata，以下BAと書く）に変換する．Büchiは，「ビュヒ」とか「ビヒ」と発音するスイスの数学者の名前である．

とりあえず簡単なモデルで試してみよう．**図2**のようなテキスト・ファイルを作成してLTSAに読み込ませる．P1の<>aは「いずれaが実行される」，P2の[]aは「常にaが実行される」という意味である．プロセスAは，アクションとしてa，b，cの3種類を持つが，実行順序については何も制約を持っていないプロセスである．

普通にプロセスAをコンパイルする．そして，Checkメニューから定義してあるアサーションを選択する．すると，LTSAがアサーションで指定したLTL式からそれに対応する否定の意味を持ったBAを生成して，それとプロセスAを合成してプロセスAがLTL式を満足するかどうか検査してくれる．P1の検査のようすと結果を**図3**に示す．P1が成立するという結果が得られた．

ほかのモデル検査ツールだと，この状況ではbとcだけ実行するループを検出して，「いずれaが実行される」は偽になる．そのようなNuSMVモデルを2通り**図4**に示す．これは，SPINやNuSMVのように変数を使用して変数値によって状態を表しているか，LTSAのように直接アクションの実行によって状態を表しているかの違いである．変数を導入する前から検証ができるLTSAは，変数によらず動作を検証する必要がある要求工程で便利である．しかし，注意するべき点でもある．

例えば，プロセスAを，

```
A=（b->c->A）．
```
……(1)

のように変更してもP1は成立してしまう．これは，プロセスAはbとcの実行に関しては制約を与えているがaについては

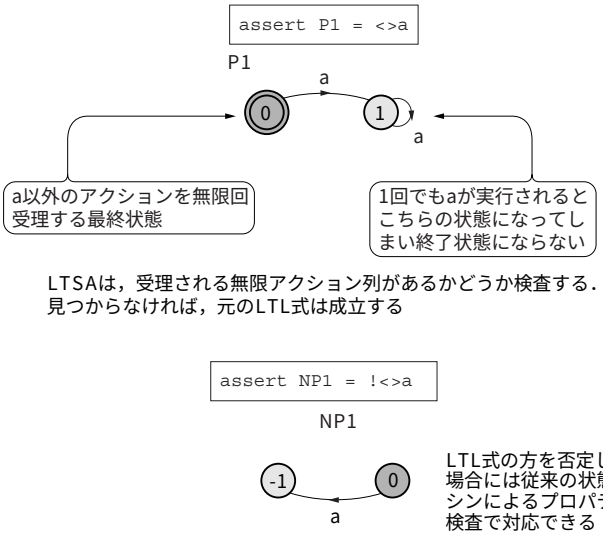

図5　Büchiオートマトンの意味1

何も制約していないので，AとP1を合成した際にaは自由に実行できてしまうためである．プロセスAを，

```
A=（b->c->A）+ {a}．
```
……(2)

のようにしてaの実行を制約する必要がある．こうすればP1は成立しなくなる．

BAの2重丸で表示される状態は，そのBAに含まれないアクションを無限回受理する最終状態である．bとcだけを繰り返す場合，**図5**のP1は状態0にずっと留まることになるのでP1はその実行系列を受理することになり，元のLTL式の否定が成立する．つまり，元のLTL式は否定されてしまう．そのようなステート・マシンの例は**図6**のA2のようなものである．実際にP1でLTL Property checkを実施すると以下のようにbとcを繰り返す反例が報告される．

```
LTL Property Check...
-- States: 4 Transitions: 8 Memory ...
Finding trace to cycle...
Depth 2 -- States: 2 Transitions: 4 ...
Finding trace in cycle...
Depth 1 -- States: 1 Transitions: 1 ...
```

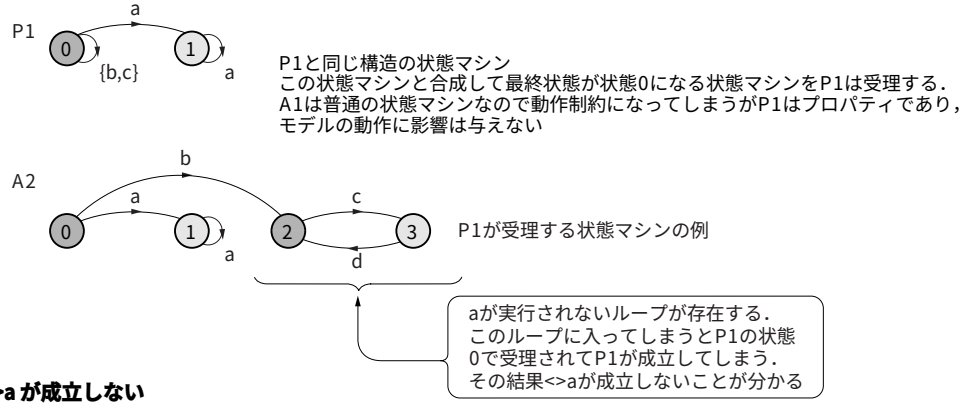

図6
P1が受理する =<>a が成立しない