記号モデル検査を用いたインフラストラクチャーにおける障害伝搬の解析について

土屋 達弘[1,a)]　藤崎 泰正[1]

概要：形式手法の一種であるモデル検査について，伝統的な情報システムの検証以外の問題への適用可能性について議論する．具体的には，電力システム等のインフラストラクチャーにおける障害伝搬の解析への適用を考える．近年，インフラストラクチャーを相互に依存する複数のシステムからなるものと見なし，それらの相互作用の結果として障害伝搬をモデル化し，解析する手法が注目を集めている．そのようなモデルが，記号モデル検査の枠組みで表現，解析できることをモデル検査器 nuXmv を用いて具体的に示し，今後の展開の可能性について議論する．

On applicability of symbolic model checking to analysis of cascading failures in critical infrastructures

本稿では，形式手法の一種であるモデル検査について，伝統的な情報システムの検証以外の問題への適用可能性について議論する．具体的には，電力システム等のインフラストラクチャーにおける障害伝搬の解析への適用を考える．近年，インフラストラクチャーを相互に依存する複数のシステムからなるものと見なし，それらの相互作用の結果として障害伝搬をモデル化し，解析する手法が注目を集めている．特に，電力システムを電力網と通信網の二つのシステムからなるシステムオブシステムズ (SoS) とみなして，個々のシステムの一部の障害が，別のシステム（ネットワーク）の障害を導き，それがまた元のシステムの障害を導くというモデルによって，電力システムのカスケード障害を説明，解析するという方法が広く研究されている [1]．

単純な例として，電力網が a_1, a_2, a_3, a_4 の 4 コンポーネントから，通信網が b_1, b_2, b_3 の 3 コンポーネントから構成される電力システムを考える．また，コンポーネントが正常に稼働するためには，依存する他方のシステムのコンポーネントが正常でなければならないものとする．たとえば，a_3 が正常に稼働するためには，b_1, b_2, b_3 すべてが正常でなければならないものとする．各コンポーネントは正常状態と障害状態の 2 状態とし，障害は離散的な時間ステッ

```
MODULE main
VAR
  a1 : boolean;
  a2 : boolean;
  a3 : boolean;
  a4 : boolean;

  b1 : boolean;
  b2 : boolean;
  b3 : boolean;

ASSIGN
  init(b1) := TRUE;
  init(b2) := TRUE;
  init(b3) := TRUE;

  next(a1) := a1 & (b1 | b2);
  next(a2) := a2 & (b1 & b3 | b2);
  next(a3) := a3 & (b1 & b2 & b3);
  next(a4) := a4 & (b1 | b2 | b3);
  next(b1) := b1 & (a1 | a2 & a3);
  next(b2) := b2 & (a1 | a3);
  next(b3) := b3 & (a1 & a2);

INIT
  count(a1, a2, a3, a4) >= 3

CTLSPEC AG (a1 | a2 | a3 | a4 | b1 | b2 | b3)
```

図 1　障害伝搬を表現した nuXmv プログラム

[1] 大阪大学
1-5 Yamadaoka, Suita, Osaka 565–0871, Japan
[a)] t-tutiya@ist.osaka-u.ac.jp

```
-- specification AG ((((((a1 | a2) | a3) | a4) | b1) | b2) | b3)  is false
-- as demonstrated by the following execution sequence
Trace Description: CTL Counterexample
Trace Type: Counterexample
  -> State: 1.1 <-
    a1 = FALSE
    a2 = TRUE
    a3 = TRUE
    a4 = TRUE
    b1 = TRUE
    b2 = TRUE
    b3 = TRUE
  -> State: 1.2 <-
    b3 = FALSE
  -> State: 1.3 <-
    a3 = FALSE
  -> State: 1.4 <-
    b1 = FALSE
    b2 = FALSE
  -> State: 1.5 <-
    a2 = FALSE
    a4 = FALSE
```

図 2 CTL 式に対する反例

プに従って伝搬するものとする．上の例の場合，a_3 が正常状態で b_1, b_2, b_3 のいずれかがステップ t で障害状態となったならば，次の時間ステップであるステップ $t+1$ において，a_3 は障害状態に変化する．

図 1 は，この例を記号モデル検査器 nuXmv の入力言語を用いて表現したものである．このような nuXmv への入力を，以後単にプログラムと呼ぶ．このプログラムでは，キーワード `CTLSPEC` の下に，検証する性質を時相論理の一つである CTL によって記述している．この CTL 式が表しているのは，「常にいずれかのコンポーネントが正常である」，という命題である．検証の結果を図 2 に示す．これは上記の CTL 式が満たされないこと，および，その証拠となる反例を示している．具体的には，a_1 の初期状態が障害状態であった場合，ステップ 4 において全コンポーネントが障害状態となるシナリオを示している．

このプログラムで注意すべき点は，システムの初期大域状態を，特定の状態ではなく，状態集合として与えている点である．具体的には，`ASSIGN` 以下の宣言で b_1, b_2, b_3 はすべて正常であるという条件を課し，かつ，`INIT` 以下の

```
count(a1, a2, a3, a4) >= 3
```

という記述によって，a_1, a_2, a_3, a_4 の中で正常状態であるものの数が 3 以上という条件を設定している．つまり，特定の初期状態から単にシミュレーションを行うのではなく，複数の初期状態からのシナリオを並行に検査することを実現している．状態探索を一つ一つの状態を走査して行うのではなく，論理関数で表された状態集合をもとに行う記号モデル検査の性質を用いることで，このような暗黙的な並行処理が実現できる．

電力システムなどのインフラストラクチャーにおける障害伝搬の解析ツールとして，記号モデル検査は有効か，もしくは，どのように用いれば有効となるかについて議論したい．以下は具体的な論点の例である．

- 記号モデル検査を用いて以下の様な問題を解くことが可能か．
  - 障害の影響が最も大きいコンポーネント群の検出 (critical component detection)
  - 強化すべきコンポーネント群の特定 (component hardening)
- 2 分決定図 (Binary decision diagram; BDD) を用いた記号モデル検査と，充足可能性判定を用いた有界モデル検査[*1]や k-induction などの記号モデル検査とでは，どの方法がより有効か．
- シミュレーション等の方法に比べて，記号モデル検査は有用か．

謝辞

本研究は，JST, CREST, JPMJCR15K2 の支援を受けたものである．

参考文献

[1] Eusgeld, I., Nan, C., and Sven, D.: “System-of-systems” approach for interdependent critical infrastructures, Reliability Engineering & System Safety, Volume 96, Issue 6, Pages 679-686 (2011)

[2] Tsuchiya, T. and Fujisaki, Y.: Satisfiability-Based Analysis of Cascading Failures in Power System Networks, Proc. SICE International Symposium on Control Systems 2017 (Part of the 4th Multi-symposium on Control Systems), Okayama, 3A1-4, USB (2017)

[*1] 我々はこれまで有界モデル検査を援用したカスケード障害の解析手法を提案している [2].