Vol.2021-EMB-57 No.4
2021/6/28

組込みシステム向けFRP言語におけるモデル検査を用いた状態依存動作の検証

内藤 博[1,a)]　森口 草介[1,b)]　渡部 卓雄[1,c)]

概要：関数リアクティブプログラミング (FRP) は組込みシステム等のリアクティブシステムを簡潔に記述するための方法である．FRP では時間変化する値を時変値として表現し，システムをそれらの依存関係に基づいて表現する．組込みシステムの設計にはしばしば状態遷移モデルが用いられる．モデルの性質はモデル検査を用いて検証できるが，抽象度の違いからその性質を実装（コード）がみたしているか否かを示すのは一般に難しい．XStorm は状態遷移モデルを用いて設計された組込みシステムに適した純粋 FRP 言語であり，状態依存動作を陽に表現するための言語機構を備えている．本研究では XStorm で記述されたプログラムから状態遷移モデルを抽出してモデル検査器 Spin を用いて検証する方法を提案し，例題の記述および検証を通してその有効性について議論する．本手法で抽出されたモデルは設計モデルに近く，比較的抽象度の高い性質の検証が可能になる．

1. はじめに

関数リアクティブプログラミング *(Functional Reactive Programming, FRP)*[1] は，連続的に変化する入力に応じて出力を変えるリアクティブシステムを簡潔に表現するために開発されたプログラミングパラダイムであり，時間変化する値を**時変値** *(time varying value)* として扱い，時変値の依存関係に基づいてシステムを表現する．組込みシステムの多くはリアクティブシステムであり，状態遷移モデルを用いて設計される．モデルの性質はモデル検査を用いて検証できるが，抽象的なモデルで成立する性質が実際のコードで成立することを証明するのは難しい．XStorm[2] は状態依存動作を明示的に記述できる組込みシステム向け FRP 言語であり，設計モデルに近い実装を可能としている．本研究では XStorm のプログラムから状態遷移モデルを機械的に抽出してモデル検査器 Spin を用いて検証する方法を提案する．本手法の利点は，システムの実装から直接抽象度の高い検証を可能としていることである．また，例題の実装と検証を行い，本手法の有用性について議論する．

1 東京工業大学情報理工学院情報工学系
Department of Computer Science, School of Computing, Tokyo Institute of Technology, Meguro, Tokyo 152–8552, Japan
a) naito@psg.c.titech.ac.jp
b) chiguri@acm.org
c) takuo@acm.org

2. モデル生成手法

XStorm のプログラムから Spin のモデル記述言語 Promela による記述を抽出する手法について説明する．

2.1 例題

例として下のような switchmoduleの変換を考える．

```
1  switchmodule Main() {
2    in dt(0) : Int
3    out output(False) : Bool
4    init State1
5
6    state State1 {
7      node wait(0) = wait@last + dt
8      out node output = True
9
10     switch:
11       if wait >= 100 then State2 else Retain
12   }
13   state State2 {
14     node wait(0) = wait@last + dt
15     out node output = False
16
17     switch:
18       if wait >= 200 then State1 else Retain
19   }
20 }
```

XStorm ではシステムを moduleの組み合わせで表現し，状態遷移を持つ moduleを switchmoduleとして表現する．これは直前の更新からの経過時間 dtを入力として受け取る

switchmoduleである．State1，State2の 2 状態を持ち，経過時間に応じて状態が変化し，出力が変わる．

2.2 Promela の変数の生成

前節の XStorm プログラムから Promela のコードを抽出する．まず Promela の変数を以下のように生成する．

```
mtype = {State1, State2}
mtype Main = State1;
mtype NMain;
int LMaindt = 0;
int Maindt;
```

この例では状態 State1と State2を列挙型で定義し，現在の状態を表す Mainと次の状態を表す NMainを列挙型の変数としている．初期状態は State1であるため，Mainを State1で初期化している．また，XStorm の時変値は現在値と直前の値のみアクセスができるため，入力の時変値 dt の直前値 LMaindtと現在値 Maindtという変数を定義し，LMaindtを初期値 0 で初期化している．

2.3 非決定的な入力の生成

XStorm モジュールの入力となり得る値を以下のように Promela の非決定的分岐の構文を用いて生成する．

```
Maindt = LOW;
do
:: Maindt < HIGH -> Maindt = Maindt + 1
:: break
od;
```

Bool型の入力に対して Trueと Falseどちらも取りうるような非決定的分岐を設定し，この例のような Int型の入力に対しては上限値 (HIGH) と下限値 (LOW) を定義してその間の値が非決定的に選ばれるように設定する．これにより，検証の際にシステムが取りうる入力全てについての網羅的な探索が可能となる．

2.4 XStorm モジュールの動作のモデル化

XStorm モジュールの動作を状態遷移系として Promela でモデル化する．まずは XStorm のコードにしたがって次の状態を計算する．具体的には XStorm の時変値や状態の計算式をそのまま使用し 2.2 節で用意した各変数の値を計算する．Promela に無い構文については意味が変化しないように適宜変換を行う．また，現在の時変値および各モジュールの次の状態の計算が全て終わったところにラベルをつけて，その部分で性質の検証を行うこととする．

以上に続けて，現在の時変値を直前値に，次の状態を現在の状態に代入することでシステム全体を更新する．そして 2.3 節の入力の生成部分に戻る．以降，これをループし続けることでランダムな入力に対し状態と出力を変化させ続けるリアクティブシステムの挙動を表現している．

3. ケーススタディ

本研究では検証対象として話題沸騰ポット [3] を使用した．話題沸騰ポットは，沸騰，保温，給湯機能を持つタイマ付き給湯ポットである．実装したプログラム全体については [4] の/src/XStormPot 以下を参照のこと．コード行数は 625 行となった．また，2 章で述べた変換手法に従って，話題沸騰ポットのコードを Promela によるモデル記述に変換した．変換に際して状態爆発を抑えるため検証と関係しない部分の抽象化を手動で行った．抽象化後の Promela の記述全体は [4] の/src/PromelaPot 以下を参照のこと．抽象化前のコードが 493 行であったのに対し，抽象化後のコードは 353 行となった．仕様書の仕様から性質を抽出し，LTL 式で表現してモデルの上で成立するかを Spin を用いて検証した．ここでは検証した性質の例として「ポットが保温状態で給湯中なら，沸騰状態には遷移しない」を挙げる．これを LTL 式 `[](Main@check -> (Pot == keep && potsupply -> NPot != boil))`に変換し検証を行ったところ，反例がないことが確認できた．

4. 結論

本研究では XStorm のプログラムを Promela に機械的に変換し，Spin を用いて性質の検証を行う手法を提案した．XStorm によるシステムの実装から直接形式手法を用いた抽象度の高い検証が可能であることを例題（話題沸騰ポット）を通して示すことができた．

今回の提案手法では，Promela に存在しない型や構文についての変換規則を定めておらず，非決定的な入力や整数型，検証する性質と関係ない部分の抽象化を行っていないので，検証するシステムの規模によっては状態爆発により現実的な時間で検証が出来ない可能性がある．今後の課題としては，これらを踏まえたより具体的な変換規則の定義と，それを用いた自動変換機の作成が挙げられる．

謝辞 本研究の一部は JSPS 科研費 21K11822 および 19K20245 の助成を受けている．

参考文献

[1] Bainomugisha, E., Carreton, A. L., Cutsem, T. V., Mostinckx, S. and Meuter, W. D.: A Survey on reactive programming, *ACM Comput. Surv.*, Vol. 45, No. 52 (2013).
[2] 松村有倫：組込みシステム向け FRP 言語における状態依存動作のための抽象化機構，修士論文，東京工業大学情報理工学院 (2020).
[3] 組込みソフトウェア管理者・技術者育成研究会 (SESSAME)：話題沸騰ポット (GOMA-1015 型) 要求仕様書第 7 版，http://www.sessame.jp/workinggroup/WorkingGroup2/POT_Specification_v7.PDF (2005).
[4] 内藤 博，XStormPot, https://github.com/calcite1036/XStormPot (2021).