B-002 CafeOBJ を用いたハイブリッドシステムの形式的な仕様記述と検証
Formal specification and verification of the hybrid system by using CafeOBJ

山岸 大悟 † 清野 貴博 † 緒方 和博 ‡ 二木 厚吉 †
Daigo Yamagishi Takahiro Seino Kazuhiro Ogata Kokichi Futatsugi

1. はじめに

離散系と連続系の混合システムであるハイブリッドシステムに関する研究は, システム制御と計算機科学両方の分野において注目されている, 自動車や飛行機などに見られるハイブリッドシステムは高信頼性が要求され, 形式的な仕様記述と, 記述された仕様を検証するための研究は, システムの安全性保証の面から重要である. ハイブリッドシステムの形式的な仕様記述と, 記述された仕様に対する自動検証を行うことを目指して, 本研究では形式仕様言語 CafeOBJ を用いたハイブリッドシステムの仕様記述手法と検証手法の提案を行う. 本稿では, CafeOBJ でリアルタイムシステムを記述するためのモデルである TOTS[1] と, ハイブリッドシステムのモデルである PTS[2][3] を参考にすることで, ハイブリッドシステムの例題に対して, CafeOBJ を用いた仕様記述を行なった.

2. 代数仕様言語 CafeOBJ

CafeOBJ は, 主に抽象データ型を記述するための始代数, および抽象機械を記述するための状態代数に基礎をおいている. CafeOBJ には, 一般のプログラム言語の型に相当するソートが 2 種類存在し, 抽象データ型は可視ソートによって表され, 抽象機械は状態ソートによって表現される. 状態ソートに関連して, 2 つの演算がある. 作用演算と観測演算である. 前者は抽象機械の状態を変化させるのに, 後者は抽象機械の内部状態を観測するために用いる.

3. TOTS

CafeOBJ を用いて状態機械を記述するための計算モデルに, OTS (Observational Transition System) がある. リアルタイムシステム記述のための計算モデルは, OTS に時間制約を加えた TOTS (Timed OTS) により構成される. TOTS は TOTS $\mathcal{S} = \langle \mathcal{O}, \mathcal{I}, \mathcal{T} \cup \{tick_r | r \in R^+\} \rangle$ で定義される [1].

- $\mathcal{O}$: 観測の集合. 集合 $\mathcal{O} = \mathcal{D} \cup \mathcal{C}$ は, 離散変数の集合 $\mathcal{D}$ と非負の実数 R^+ or ∞ の型を持つクロック変数 $\mathcal{C}$ で定義される. 各観測 $o \in \mathcal{O}$ は, 状態空間 Υ から任意のデータ型 $\mathcal{D}$ への関数 $o : \Upsilon \to \mathcal{D}$ である. 集合 $\mathcal{C}$ の要素には, マスタークロック $now : R^+$ が存在する.
- $\mathcal{I}$: 初期状態の集合 $\mathcal{I} \subset \Upsilon$ マスタークロック now は, 初期状態では 0 にセットされる.
- $\mathcal{T}$: 条件付き遷移規則の集合. 遷移規則 $\tau \in \mathcal{T}$ は現在の状態を次の状態へ写像する関数, $\tau : \Upsilon \to \Upsilon$ である. さらに $\mathcal{T}$ の要素として, 時間前進のための遷移規則である $tick_r$ がある. $\tau \in \mathcal{T}$ に付随する条件 $c_\tau : \Upsilon \to \{true, false\}$ を効力条件と呼ぶ. c_τ が $true$ のときに限り, 現在の状態を次の状態へ遷移させることが可能である.

† 北陸先端科学技術大学院大学, Japan Advanced Institute of Science and Technology (JAIST)
‡ (株) NEC ソフトウェア北陸, NEC Software Hokuriku, Ltd.

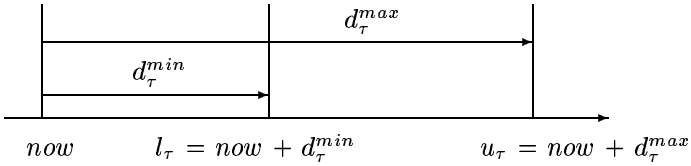

図 1: 遷移規則が及ぼす境界の変化

各遷移規則は, 下界 $\{ l_\tau : R^+ \}$ と上界 $\{u_\tau : \{R^+ \backslash 0\} \cup \infty\}$ の 2 つのクロック変数を持っている. 各遷移規則は, 下界と上界の間で遷移が生じなければならない. $\tau \in \mathcal{T}$ が効果を及ぼすとき, 遷移後の状態が持つ上界, 下界は, それに対応する定数 d_τ^{min}, d_τ^{max} を用いることにより, マスタークロック now からの変化として図 1 の様に設定される.

4. PTS を用いた仕様記述

本研究では, PTS (Phase Transition System) を参考にしたハイブリッドシステムの仕様記述を行う. PTS Φ は 5 つ組 $\Phi = \langle V, \Theta, \mathcal{T}, \mathcal{A}, \Pi \rangle$ で定義される [2][3].

- V : システム変数の有限集合である. 集合 $V \cup I$ は, 離散変数の集合と連続変数の集合に分類できる. 離散変数の型は任意であるが, 連続変数の型は実数型である. さらに $t \in I$ となるマスタークロック t を導入する.
- Θ : 初期条件. すべての初期状態を特徴付ける表明である. $\Theta \Rightarrow t = 0$ が要求される.
- $\mathcal{T}$: 条件付遷移の集合, 状態遷移 $\tau \in \mathcal{T}$ は, 関数 $\tau : \Sigma \to 2^\Sigma$ である. $\tau \in \mathcal{T}$ は, 各状態 $s \in \Sigma$ を, 状態遷移関数 $\rho_\tau(V, V')$ により, 次状態 $s' : \tau(s) \subseteq \Sigma$ に写像する.
- $\mathcal{A}$: アクティビティの有限集合. $\alpha \in \mathcal{A}$ はアクティビティ関係 $p_\alpha \to I(t) = F^\alpha(V^0, t)$ によって表現される. ここで, p_α は α の '*activation condition*' と呼ばれる D 上の述語である. p_α が s 上で成り立つならば, アクティビティ α はアクティブといわれる. $F^\alpha(V^0, t)$ は常微分方程式によって表現される変化の関数である.
- Π : 時間前進条件. 時間前進の制限をするための条件である.

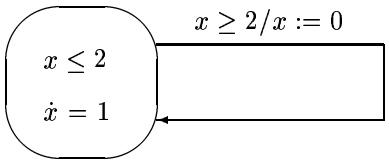

図 2: ハイブリッドシステムを用いた例題

5. ハイブリッドシステムを用いた例題

ハイブリッドシステムを CafeOBJ で記述するために, 図 2 ような例題を考える. 図 2 では, 有効グラフのラベルは, “条件/動作” を意味している. この例題では, $x \geq 2$ ならば x を 0 に更新してフェーズが遷移することを表す. フェーズ内部では, $x \leq 2$ の制約で, 常微分方程式 $\dot{x} = 1$ に従って x の値が変化する.

5.1 CafeOBJ による記述

例題に対して CafeOBJ を用いた記述を行なう. PTS における各要素は TOTS において以下の様に対比が可能である.

- PTS におけるシステム変数の集合 $\mathcal{V}$ は, TOTS における観測の集合 $\mathcal{O}$ に対応が可能である. ただし TOTS においては, クロック変数の型は, 非負の実数で定義されている.
- PTS における初期条件 Θ, 条件付遷移規則の集合 $\mathcal{T}$ は , そのまま TOTS の $\mathcal{I}$, $\mathcal{T}$ に対応可能である.

アクティビティの集合 $\mathcal{A}$ は, マスタークロック *now* を操作するための作用演算 *tick* に, 常微分方程式による観測値の変化を加えて記述する. 時間前進条件 Π は, フェーズ内部で定義された変数の制約条件になる. 以下は CafeOBJ のコードとその解説である.

```
*[Sys]*
-- initial constractor
      op init : -> Sys
-- Ordinary Differential equation Definition.
      op func : Real+ -> Real+
-- observations
      bop now : Sys -> Real+
      bop x : Sys -> Real+
```

‘*[’,‘]*’ の間に使用する状態ソートの記述を行う. CafeOBJ における観測の定義は, ‘bop’ の後に関数名を指定することによって行う. ‘:’ の後には引数の型を指定し, ‘->’ の後に返り値の型を記述する. ‘now’, ‘x’ はそれぞれ, マスタークロック, フェーズ内の値を表現する観測演算名である. ‘init’, ‘func’ はそれぞれ, システムの初期状態を表わす定数と常微分方程式を表す関数である. 各観測に対応する初期条件は以下のように記述される.

```
eq now(init) = 0 .
eq x(init)  = 0 .
```

各観測値に対する初期条件を ‘eq’ の後に等式を記述することにより表している. 以下は各遷移規則に対する定義とそれが必要とする効力条件, 遷移後の状態変化に対する記述である.

```
-- set of transitions: action
bop action : Sys -> Sys
-- effective condition:
op c-action : Sys -> Bool
eq c-action (S:Sys) = 2 <= x(S) .
-- behavior:
ceq now (action (S:Sys)) = now(S)
    if c-action (S) .
ceq now (action (S:Sys)) = now (S)
    if not (c-action (S)) .
ceq x (action (S:Sys)) = 0
    if c-action (S) .
ceq x (action (S:Sys)) = x (S)
    if not (c-action (S)) .
```

遷移規則は引数として 1 個の状態ソートを持ち, 返り値として状態ソートを持つ関数として宣言される. 効力条件は ‘c-action’ で定義する. ここでは, 内部状態の観測値が 2 以上になった場合に状態が遷移可能であるように定義している. ‘behavior’ 以下は, 遷移規則が起こす効果であり, すべての観測値に対しての等式を宣言する必要がある. 上記のコードでは, 観測値 x が遷移が生じた後は 0 になるように設定している. 時間前進規則 ‘tick’ に関しては, 以下のように記述される.

```
-- set of transitions: tick
bop tick : Real+ Sys -> Sys
-- effective condition:
op c-tick : Real+ Sys -> Bool
eq c-tick (R:Real+,S:Sys) = x(S) + func(R) <= 2 .
-- behavior:
ceq now (tick (R:Real+,S:Sys)) = now(S) + R
    if c-tick (R,S) .
ceq now (tick (R:Real+,S:Sys)) = now (S)
    if not (c-tick (R,S)) .
ceq x (tick (R:Real+,S:Sys)) = x(S) + func(R)
    if c-tick (R,S) .
ceq x (tick (R:Real+,S:Sys)) = x (S)
    if not (c-tick (R,S)) .
```

‘tick’ の効力条件には, PTS の時間前進条件を記述する必要がある. ‘tick’ が観測値 ‘now’ に及ぼす効果は, 現在時刻 ‘now’ に進行時間 ‘R’ の前進を加えたもので記述される. ‘x’ に及ぼす効果は, 現在の観測値 x に常微分方程式の解を返す関数 ‘func’ を加えたものを記述する.

5.2 CafeOBJ を用いた検証

CafeOBJ では検証したい性質を CafeOBJ の項で記述し, CafeOBJ 仕様に基づき, 証明譜を記述する. 証明譜を CafeOBJ 処理系を用いて実行することで, 検証が可能である.

6. まとめと今後の課題

本稿では, TOTS と PTS を参考にすることで, CafeOBJ を用いた簡単な例題の仕様記述を行った. 今後は, より発展させた例題に対する仕様記述と検証を行う予定である. さらに, CafeOBJ でハイブリッドシステムを記述するための計算モデル作成のために, PTS で用いられているフェーズ列を状態列に変換する手法の考察も行ないたい.

参考文献

[1] K. Ogata and K. Futatsugi: Modeling and Verification of Distributed Real-time Systems Based on CafeOBJ, *16th IEEE International Conference on Automated Software Engineering*, IEEE CS Press, 185-192, 2001.

[2] Kesten. Y., Manna, Z. and Pnueli, A.: Verification of Clocked and Hybrid Systems, *Acta Informatica*, Vol.36, No.11, pp.836-912 2000.

[3] 山根 智 : ハイブリッドシステムのモジュールの仕様記述と検証の手法, 情報処理学会論文誌, Vol.44,No.3, pp.897-914,2003.