

代数的仕様からのドキュメント自動生成について

8 J-4

片岡 充照<sup>†</sup> 工藤 朋之<sup>‡</sup> 石原 靖哲<sup>‡</sup> 関 浩之<sup>‡</sup>  
<sup>†</sup>松下電器産業(株) 中央研究所 <sup>‡</sup>大阪大学 基礎工学部

1 はじめに

ソフトウェアの生産性と品質を向上させる手法として、実装者に提供する外部仕様を代数的記述法 [1] を用いて記述する方法が注目されている。ところが外部仕様の記述に代数的記述のみを用いる場合には、実装者が代数的記述法に精通していることが必須である。一方、代数的記述による外部仕様(以下、単に代数的仕様と呼ぶ)の補助として自然語のドキュメント(以下、単にドキュメントと呼ぶ)を用意する場合には、代数的仕様とドキュメントの間の整合性に問題が生じる。本稿ではこれらの問題を解決するため、抽象的順序機械型仕様 [2] と呼ばれる代数的記述の部分クラスからドキュメントを自動的に生成する方法について考察する。またドキュメントの生成システムについても述べる。

2 抽象的順序機械型仕様による外部仕様

抽象的順序機械型仕様(以後順序機械仕様と呼ぶ)は代数的記述法の一方法であり、関数は順序機械の状態を変化させる状態遷移関数と状態を観測する観測関数に分類される(図1)。

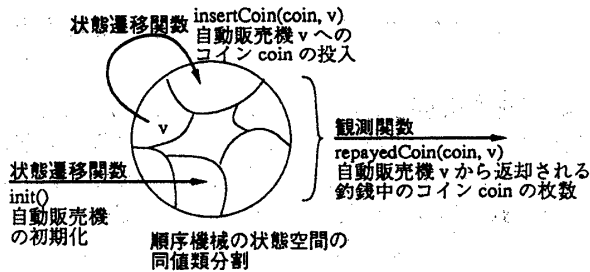


図1: 抽象的順序機械の例(自動販売機)

順序機械仕様の公理では各状態遷移関数で表される状態遷移後の観測関数の値を、遷移前の状態における観測関数の値を用いて定義する。なお、筆者らは実際のシステムに適用できるように拡張した順序機械仕様記述法を考案し、開発中のマルチメディアデータベースシステム DAL/MA[3] の外部仕様を用いている。

図2に順序機械仕様の例を示す。init, insertCoin, buy は状態遷移関数、repayedCoins と insertedMoney

```
Specification
Vendor
State Transition Functions
o   init:                               -> Vendor
o   insertCoin: CoinType, Vendor -> Vendor
o   buy:   Vendor -> Vendor
Observation Functions
insertedMoney:   Vendor -> Nat
repayedCoins:   CoinType, Vendor -> Nat
Axioms
v: Vendor;
coin, newCoin : CoinType;

insertedMoney(init()) = 0;
repayedCoins(coin, init()) = 0;

/* 公理 A */
insertedMoney(insertCoin(newCoin, v)) =
    insertedMoney(v)+CoinType'value(newCoin);

repayedCoins(coin, insertCoin(newCoin, v)) = 0;
insertedMoney(buy(v)) =
    if (insertedMoney(v)>=110) 0;
repayedCoins(coin, buy(v)) =
    if (insertedMoney(v)>=110)
        switch(coin) {
        case CoinType'TEN:
            (insertedMoney(v)-110)%100)/10;
        case CoinType'HUNDRED:
            (insertedMoney(v)-110)/100;
        }
}
```

図2: 代数的記述による外部仕様の例(自動販売機)

は観測関数である。insertCoin は遷移前の状態を表す引数(第2引数)の他に、CoinType 型の引数(第1引数)をもつ。insertedMoney は状態を表す引数をもち、Nat 型の値をとる。また、公理 A は、「関数 insertCoin(newCoin, \*) で表される状態遷移が起こった後の観測関数 insertedMoney の値は、遷移前の状態における insertedMoney の値と newCoin の額面との和となる」と定義している。

3 外部仕様に関するドキュメントの自動生成

一般の自然語のドキュメントでは、省略や参照といった照応関係が読者の理解を助ける一方、曖昧さや不正確さの一因となる場合がある。ところが、本稿で問題とするドキュメントでは、代数的仕様に十分精通していない実装者に、正確に外部仕様の内容を伝達する目的で用いられる。従って、ドキュメントは曖昧さのない正確な表現で記述されていることが望ましい。また、ドキュメントはもとの代数的仕様と相互補完的に用いられるとし、実装者は必要なら代数的仕様を参照するものとする。従って、変換はテンプレートマッチング方式で十分と思われる。

Document Generation from an Algebraic Specification  
 Mitsuteru Kataoka<sup>†</sup>, Tomoyuki Kudo<sup>†</sup>, Yasunori Ishihara<sup>†</sup>, and  
 Hiroyuki Seki<sup>‡</sup>  
<sup>†</sup> Central Research Laboratories, Matsushita Electric Industrial  
 Co., Ltd.  
<sup>‡</sup> Dept. of Information & Computer Sciences, Osaka University

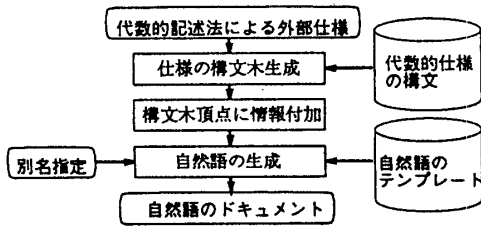


図 3: 自動生成システム

表 1: 別名指定の例 (自動販売機)

変数と関数	別名
init	初期化した後の Vendor
insertCoin(C, V)	C を投入した後の V
buy(V)	商品を購入した後の V
repayedCoins(C, V)	V における, 釣銭中の C の枚数
insertedMoney(V)	V への投入金額
CoinType'TEN	10 円硬貨
CoinType'HUNDRED	100 円硬貨
CoinType'value(C)	C の額面

図 3 は代数的仕様からドキュメントを生成するシステムのブロック図である。本システムは約 300 節の Prolog プログラムで記述されている。

順序機械仕様では, 左辺の観測関数の引数に状態遷移関数がちょうど一回しか現れず, かつ, 右辺には状態遷移関数が現れない。これより, 以下のようにして状態遷移モデルに基づいた理解しやすい日本語文を比較的容易に生成することができる。ただし, 代数的仕様における順序機械名を  $M$  とし,  $s$  を  $M$  の状態を表す変数,  $OF$  を観測関数,  $STF$  を状態遷移関数とする。

- 公理左辺の変換: 例えば, 左辺  $OF(a_1, STF(a_2, s))$  は, 「 $a_2$  について  $STF$  した後の  $M$  の  $a_1$  に対する  $OF$  は」と変換する。
- 公理右辺の変換: 右辺に現れる観測関数  $OF$  は, 遷移前の状態を引数にとる。そこで, 例えば, 右辺中の式  $OF(a_1, a_2, s)$  は, 「もとの  $M$  の  $a_1, a_2$  に対する  $OF$ 」と変換する。

さらに, 以下のような工夫を行っている。

- 条件分岐の変換: ネストの深さに依存して, 場合分けを表す用語を選択したり, インデントをつける。
- 別名指定の利用: もとの代数的仕様の記述者が, 生成されたドキュメントを確認した後, 必要ならば, 理解しやすい別名を指定することができる (表 1)。

本システムに, 図 2 の代数的仕様と表 1 の別名指定を入力としたとき, 生成されるドキュメントを図 4 に示す。

#### 4 おわりに

今後は, 代数的記述法の意味定義の簡潔さを利用し, 省略表現を導入したいと考えている。例えば, 図 4 のよう

[名前]

Vendor

[形式]

(i) 状態遷移関数

Vendor init();

Vendor insertCoin(CoinType, Vendor);

Vendor buy(Vendor);

(ii) 観測関数

Nat insertedMoney(Vendor);

Nat repayedCoins(CoinType, Vendor);

[機能説明]

$v$  を Vendor 型の仮引数,

$coin$  を CoinType 型の仮引数,

$newCoin$  を CoinType 型の仮引数とする。

< 状態遷移関数 init >

初期化した後の Vendor への投入金額は,

0 となる。

初期化した後の Vendor における, 釣銭中の coin の枚数は,

0 となる。

< 状態遷移関数 insertCoin >

$newCoin$  を投入した後の Vendor への投入金額は,

(元の Vendor への投入金額 +  $newCoin$  の額面) となる。

$newCoin$  を投入した後の Vendor における, 釣銭中の coin の枚数は,

0 となる。

< 状態遷移関数 buy >

商品を購入した後の Vendor への投入金額は,

元の Vendor への投入金額  $\geq 110$  の場合

0

となる。

商品を購入した後の Vendor における, 釣銭中の coin の枚数は,

元の Vendor への投入金額  $\geq 110$  の場合

coin が 10 円硬貨の場合

(元の Vendor への投入金額 - 110)/10

coin が 100 円硬貨の場合

(元の Vendor への投入金額 - 110)/100

となる。

図 4: 生成されたドキュメントの例 (自動販売機)

に, 公理を状態遷移関数ごとに分類して変換・出力する場合, これらの各グループにおいて二つめ以降の公理に対する出力文では, 状態遷移関数に関する表現を省略する。また, 観測関数を定義する公理の右辺に他の観測関数を用いるという観測関数間の依存関係を利用した, ドキュメント生成のプランニングを検討したいと考えている。

謝辞 日頃よりご指導頂く, 奈良先端科学技術大学院大学 嵩忠雄教授並びに松下電器産業 (株) 奥井順室長に感謝します。

#### 参考文献

- [1] Goguen, J. A., Thatcher, J. W. and Wagner, E. G.: "An Initial Algebra Approach to the Specification, Correctness and Implementation of Abstract Data Types", IBM Research Report, RC 6487 (1976).
- [2] 杉山, 谷口, 嵩: "基底代数を前提とする代数的仕様", 信学論 (D), Vol. J64-D, No. 4, pp. 324-331 (1981-04).
- [3] 渡守武, 片岡, 田中, 邵, 奥井: "マルチメディアデータベースシステム「DAL/MA」", 信学'93 春大。