

実行可能な形式仕様言語 CafeOBJ(4)

6V-7 CafeOBJによるZ仕様のアニメーションの枠組み*

谷津弘一^{†,1} 中川中^{†,‡} 澤田寿実^{†,§} 本間毅寛^{†,2} 二木厚吉[§]

[†] 情報処理振興事業協会 (IPA) 技術センター

[‡] (株)SRA ソフトウエア工学研究所

[§] 北陸先端科学技術大学院大学

1 はじめに

仕様を記述する上で、仕様の実行が可能であることは大きなメリットをもたらす。それは、実行により仕様の内容をより深く理解できるからである。Zは集合論に基づく model-based な形式仕様言語であり高い表現力を持つが、Zで記述された仕様をそのまま実行することはできない。Z仕様の実行には他言語によるZ仕様のアニメーションが必要となる。

本稿では、IPAで開発されている実行可能な形式仕様言語 CafeOBJによるZ仕様のアニメーションの枠組みについて述べる。CafeOBJは、代数仕様言語 OBJに一方の書き換え規則を加えたもので、OBJと同様に項書き換え系により実行可能であるという特長を持つ。CafeOBJを用いることにより、Z仕様と同じレベルの抽象度を保ちながらのアニメーションが可能となる。

端的にいうと、CafeOBJモジュールは2-categoryを表す。本稿では、まずZ仕様に2-categoricalな解釈を与え、その解釈を基にZ仕様をCafeOBJモジュールで表現するための指針を導くことにより、CafeOBJによるZ仕様のアニメーションの枠組みを明らかにする。

2 Z仕様の2-categoricalな解釈

Zは対象を

- 状態空間
- 初期状態
- 状態に作用する操作

の3つで表現するスタイルをとる。この3つはスキーマと呼ばれるもので記述される(これらを記述するスキーマを、各々状態スキーマ、初期状態スキーマ、操作スキーマと呼ぶ)。本

*An executable specification language CafeOBJ(4)- A framework of CafeOBJ animation of Z specification: Hirokazu Yatsu, Ataru Nakagawa, Toshimi Sawada, Takehiro Honma, Kokichi Futatsugi

[†]Information-Technology Promotion Agency, Japan(IPA) Software Technology Center

[‡]Software Research Associates,Inc. Software Engineering Laboratory

[§]Japan Advanced Institute of Science and Technology, Hokuriku

¹日本ユニシス(株)より出向中

²(株)SRAより出向中

稿では、以下のように考えてZ仕様の記述対象を2-categoryと見做す。

1. Z仕様に見える型はオブジェクトである。
2. 状態スキーマはオブジェクトであり、属性は状態スキーマから自身が取る値の型への射である。
3. 初期状態スキーマは1から状態スキーマへの射である。
4. 操作スキーマは入力でインデックス付けられた
 - (a) domainとcodomainを共に状態スキーマとする射
 - (b) 事前条件から定まる状態スキーマのサブオブジェクト
 - (c) 事前/事後条件から定まる2-cell

の族である。

誕生日帳[4]を例にとってみよう。

[NAME, DATE]

```

BirthDayBook
known : P NAME
birthday : NAME → DATE

known = dom birthday
    
```

```

InitBirthDayBook
BirthDayBook
birthday = ∅
    
```

```

AddBirthDay
ΔBirthDayBook
name? : NAME
date? : DATE

name? ∉ known
birthday' = birthday ∪ { name? ↦ date? }
    
```

上の例では、次の5つはオブジェクトである。

NAME, DATE, *BirthDayBook*,
P NAME, NAME → DATE

*known*と*birthday*は*BirthdayBook*から各々の型への射であり、*InitBirthdayBook*は1から*BirthdayBook*への射である。

```
(-).known : BirthdayBook → P NAME
(-).birthday : BirthdayBook → (NAME ↔ DATE)
InitBirthdayBook : 1 → BirthdayBook
```

入力を固定して考えると、*AddBirthday_{name?,date?}*は*BirthdayBook*から*BirthdayBook*への射である。さらに事前条件から*BirthdayBook*のサブオブジェクト*Ker_{name?,date?}*が特徴付けられ、事前/事後条件から次のような2-cell $\delta_{name?,date?}$ が定まる。

$$\delta_{name?,date?} : \text{AddBirthday}_{name?,date?}(-).birthday |_{Ker_{name?,date?}} \rightarrow (-).birthday \cup \{name? \mapsto date?\} |_{Ker_{name?,date?}} : Ker_{name?,date?} \rightarrow (NAME \leftrightarrow DATE)$$

3 CafeOBJ モジュールによる Z 仕様の表現

CafeOBJでは、オブジェクト及び射を各々ソートと演算と見做し、射と2-cellを等式と書き換え規則で定義して2-categoryを表現する。

前節で挙げたオブジェクトをソートに、属性を演算に置き換えて、まず誕生日帳の状態スキーマをCafeOBJモジュールで表現してみよう。

```
module BIRTHDAYBOOK-STATE is
  sorts BirthdayBook Name Date .
  using FUNCTION[Name,Date]*
  (sort PDom to Names,
   sort PRan to Dates,
   sort Function to Name->>Date) .
  op _.known : BirthdayBook -> Names .
  op _.birthday :
    BirthdayBook -> Name->>Date .
  vars b : BirthdayBook .
  eq b.known = dom(b.birthday) .
endm
```

ここで、FUNCTIONはZの部分関数を表現するパラメータ付きモジュールであり、NamesとName->>Dateは各々Nameのべき集合とNameからDateへの部分関数の集合を表すソートである[6]。上の記述では、ソートBirthdayBookが状態スキーマ*BirthdayBook*を表現するよう、等式b.known = dom(b.birthday)をモジュールに加えてある。

$\delta_{name?,date?}$ のhorizontal domain $Ker_{name?,date?}$ は入力name?とdate?に依存しており、ソートとして記述することはできない。CafeOBJでは、入力に依存するサブオブジェクトの特徴付けは、等式や書き換え規則に特徴付ける条件を加えることによって自然に表現される。誕生日帳のZ仕様は、次のような条件付き書き換えを用いたCafeOBJモジュールで表現される。

```
module BIRTHDAYBOOK is
  using BIRTHDAYBOOK-STATE .
  op InitBirthdayBook : -> BirthdayBook .
  op AddBirthday___ :
```

```
Name Date BirthdayBook ->
  BirthdayBook .
var name? : Name .
var date? : Date .
var b : BirthdayBook .
eq InitBirthday.birthday = {} .
crl (AddBirthday name? date? b).birthday =>
  (b.birthday) U { < name?,date? > }
  if not(name? in (b.known)) .
endm
```

CafeOBJでは、1からの射は引数の無い演算で表される。前節の解釈ではAddBirthdayを入力でインデックス付けられる射の族としていたが、ここではソートName、Date、BirthdayBookの項を引数に取る演算にしている。また、書き換え規則に条件を加え、書き換え可能な項の集合で $Ker_{name?,date?}$ を表すようにした。これにより、上記の条件付き書き換え規則は2-cell $\delta_{name?,date?}$ を表現する。

4 おわりに

Z仕様をCafeOBJモジュールで表現するための指針について述べた。仕様のアニメーションは仕様の内容の理解を助けるが、仕様の正しさに対する考慮を奪いがちである。アニメーションはまさに諸刃の剣であり、仕様の正しさに対する考慮を忘れずにアニメーションを行なうことが必要である。

本稿の指針では、条件が等式で表現できるZ仕様に関りCafeOBJモジュールにより表現でき、アニメーションが可能になる。今後の課題として、より多くのZ仕様のアニメーションが可能になるよう一般の条件から等式を導き出す方法の考案と、アニメーションを支援する系を作成することを挙げておく。

謝辞 本稿で報告した試みは情報処理振興事業協会技術センターにおける実行可能な形式仕様言語システムの研究開発プロジェクトで行われた。

参考文献

- [1] J.Goguen, T.Winkler, J.Meseguer, K.Futatsugi and J.-P.Jouannaud, Introducing OBJ, Technical Report SRI-CSL-92-03, Computer Science Laboratory, SRI International, 1992. To appear In J.A.Goguen editor, Application of Algebraic Specification using OBJ, Cambridge University Press.
- [2] G.M.Kelly and R.Street, Review of the Elements of 2-categories, Lecture Notes in Mathematics 420, Springer, 1974.
- [3] R.A.G.Seely, Modelling Computations: A 2-categorical Framework, In Symposium on Logic in Computer Science, IEEE, 1987.
- [4] J.M.Spivey, The Z Notation: A Reference Manual 2nd ed., Prentice Hall, 1992.
- [5] Nakagawa, A.T., et al., Declarative Semantics of CafeOBJ, technical report in preparation.
- [6] 谷津, 本間, 中川, OBJによるZ仕様の検証の試み, ソフトウェアシンポジウム, 1994.