新・組み込みソフトへの数理的アプローチ

～形式仕様記述をどのように使うか～

第4回

充足問題（4）
――真理表から場合分け表へ

藤倉 俊幸

はじめに

前回（2009年3月号，pp.150-155）は真理表を使って前提から言えることを絞り込む方法と，逆に結論からそれを導く前提を出す方法について説明した．その際に，嘘つきが混じっていても，場合分けをすれば結論や前提に到達できることも示した．ただし，前回の嘘つきは必ず逆のことを言う質の良い嘘つきだった．実際の場合は，本当のことを言ったり逆のことを言ったりする質の悪い挙動をするのが普通である．今回は，質の悪い嘘つきがいる場合の対応について説明する．次に，相手がわからないと言った時に何がわかるのかについて説明する．

1 「天国への道問題」拡張版

● 天使と悪魔の問題に人間を追加

前回の質の良い嘘つきが居る場合のサンプルとした天国への道のオリジナル問題は，以下のようなものであった．前回も引用した文献(1)を使う．

> *「道が二つに分かれていて，一方は天国，一方は地獄に通じている．そこには天使と悪魔が居て，一回だけどちらかに質問することができる．ただし，天使は本当のことを言うが，悪魔は逆のことを言う．また，天使と悪魔の見分けは付かない．どのような質問をすれば良いか」*

この問題を拡張するために，天使と悪魔のほかに人間を追加する．そして，人間は質の悪い嘘つきで，本当のことを言ったり，逆のことを言ったり，あるいは単純に間違ったりする．つまり，分かれ道に立っているのは3人で，そのうちの誰かに質問をすることができる．しかし，たまたま人間に質問をしてしまうと何も情報を得られないことになる．なので，1回の質問では天国に行く道を確定することはできず，質の悪い人間を排除するために1回質問をして，残りが悪魔か天使だけになったところで2回目の質問として前回の質問をする．前回の質問というのは，

> *「あなたに，右の道は天国への道ですかと聞いたらイエスと答えますか」*
>
> *「あなたが天使で右が天国への道であるか，またはあなたは悪魔で右が地獄への道ですか」*
>
> *「あなたが天使であることと，右が天国であることは同じことですか」*

などである．

● 問題の解法

とりあえず3人をA，B，Cで区別する．この場合の可能な組み合わせは**表1**のようになる．このような場合分け表を漏れなく生成することは，ソフトウェアを設計したりテストしたりするには重要である．この例では3から3取る順列になり，全部で${}_3P_3 = 3 \times 2 = 6$通りあることはすぐに分かるが，実際の順列を漏れと重複なく生成するのは，数え上げの技術が必要になる．数え上げの考え方を学んだり自分でコツコツ数え上げるには文献(2)などが参考になる．しかし，それだけでは仕事に使うには効率が悪い．つまり，仕事で使うにはツールが必要で，ツールを使いこなし出力を確認する考え方を学んでおく必要がある．実際に生成するツールとしては，Mathematicaを利用できる．**表**

表1　場合分け

A	B	C
天使	悪魔	人間
天使	人間	悪魔
悪魔	天使	人間
悪魔	人間	天使
人間	天使	悪魔
人間	悪魔	天使

表2　質問構築

A	B	C	論理値	答え
天使	悪魔	人間	F	No
天使	人間	悪魔	T	Yes
悪魔	天使	人間	T	No
悪魔	人間	天使	F	Yes

KEYWORD ―― 真理表，天国への道問題

1 は以下の Mathematica コマンドで生成した．

```
TableForm[ Permutations[{天使, 悪魔, 人間}] ]
```

本題に戻る．まず A に質問してその答えによって B か C に質問をすることにする．A に対する質問は，B と C のどちらが人間ではないかが区別できれば良いことになる．A が人間であれば，B と C は人間ではないのでどちらに質問しても良いことになる．したがって，作るべき質問は，A が天使の場合と悪魔の場合を考慮して，B と C が人間かどうか見分ける質問と言うことになる．そこで，B が人間だったら A がハイ（Yes）と答えるような質問を考えてみる．それは，**表 2** の論理値を返すような質問である．実際に得られる答えは A が悪魔の場合は逆になることを忘れてはいけない．

場合分けが必要なので，A が天使の場合，すなわち**表 2** の上二つの場合について考える．この時，B が人間かどうかを知りたいのだから，素直に「B は人間ですか?」と聞けばよい．ただし，それだけだと A が悪魔の場合もあるので「あなたが天使の場合」という条件を付ける．したがって質問は，

「あなたが天使である場合 B は人間ですか」

「あなたが天使ならば B は人間ですか」

のようになる．A が悪魔の場合は，同じようにして，

「あなたが悪魔の場合 C は人間ですか」

「あなたが悪魔ならば C は人間ですか」

とすれば，**表 2** の下半分の場合の論理値を得る質問になる．したがって，両方を合わせると，

「あなたが天使ならば B は人間であり，あなたが悪魔ならば C が人間ですか」

と質問すればよい．

「○○ならば××」では○○が偽であれば全体は真になるので両方の AND をとれば全体の質問文が作れる．基本は素直に聞きたいことをそのまま質問文にすればよいのである．そして，疑問文を××のところに入れて，場合分けの条件を○○のところに入れる．そしてさらに場合分け全体を AND でつなげれば完成である．

前回は，真理表で任意の組み合わせで真になる質問文の作り方を説明したが，“それは聞きたいことを疑問文にする”の部分に対応する．それに場合分けを組み合わせるとこのようになるのである．このようにすれば，**表 2** の答えを得ることができるので，A が「Yes」と答えたら B が人間なので C に 2 回目の質問をすればよい．「No」と答えたら B に 2 回目の質問をすればよい．

さて，任意の疑問文を作る技術はプログラムの分岐判断を作る際に利用できる．つまり，`if` 文や `while` 文の条件部分の論理式を作る際に利用できる．「○○ならば××」の“ならば”は C 言語などにはないが「○○でないかまたは××」とすることでコーディングできる．

このようにして論理式を作るとだらだらと長くなる．もし評価時間が気になるのであれば，論理圧縮をすればよい．しかし，そうするとコーディングの意図がわからなくなるので，単に長いのが気に入らないだけならソース・コードはそのままで，実行最適化はコンパイラに任せた方がよい．詳細設計の段階では，どのような場合分けがあるのか，先に挙げた Mathematica のようなツールを使って場合分けを網羅的に検証しておく必要がある．そして，さらにその上流工程，つまり基本設計や要求分析では Mathematica に入れるべき情報，すなわちどのような場合があるのかが明確になっている必要がある．著者はこのようなアプローチが形式的なアプローチであり，ソフトウェアの品質を向上させる近道であると考えている．

コラム 1　今回使用した Mathematica 関数

`TableForm[m]`：マトリックス m をテーブル形式で出力する．

`Strings[l, n]`：リスト l の要素から作られる長さ *n* の可能なすべての文字列を生成する．l から *n* 個を取り出す重複順列を生成する．

`Permutation[l]`：リスト l から順列を生成する．

`Select[l, f]`：リスト l の要素に対して関数 *f* を実行して結果が真になる要素のリストを生成する．

`Count[l, e]`：リスト l の中に要素 e がいくつあるか返す．

`Count[#1,白]<3&`：名前なし関数，リスプのラムダ関数，#1 が引き数，&が関数の終了を表す．この場合，引き数として与えられたリスト #1 の中に“白”が 3 個未満であれば真を返す関数を定義している．

2　赤い帽子問題

● アリスとベティ問題の続き

次に相手がわからないと言った場合に，何が分かるのかについて説明する．こちらは，前回扱った，アリスが犯人