数学・確率問題を対象とした条件記述の自動解釈

岩間 純輝 †　佐藤 理史 ‡　小川 浩平 ‡　宮田 玲 ‡
† 名古屋大学工学部　‡ 名古屋大学大学院工学研究科
iwama.junki@h.mbox.nagoya-u.ac.jp

1 はじめに

大学入試問題では、言語運用能力をはじめ、多岐に渡る知識やスキルが問われる。このうち、国語・英語以外の教科の問題では、問題文を分野知識 (科目固有の知識) に基づいて解釈し、何が求められているかを理解した上で、その答を分野知識を駆使して導くことが必要となる。

2011 年から 6 年間に渡って実施された「ロボットは東大に入れるか」プロジェクトでは、主要 5 教科 (国語、英語、数学、社会、理科) の大学入試問題の自動解法が研究された [1]。このうち、数学では、問題文を実閉体の一階論理式に変換し、得られた論理式に含まれる限量子を RCF-QE アルゴリズムで消去することによって解を求める方法が、中心的な解法として採用された。その一方で、この枠組みに適合しない「確率・期待値」問題の自動解法 [2, 3] は、十分には研究されなかった。

「確率・期待値」問題は、次のような特徴を持つ。

1. 問題を解くために必要な数学的概念 (試行、結果、標本空間、事象、確率、期待値) は、それほど複雑ではない。
2. サイコロ・玉・カードなどの一般的な素材を用いて仮想世界が記述され、その世界の中で問題が記述される。
3. 事象を規定する条件の記述は、かなり複雑になりうる。
4. 問題の解答に必要な属性 (典型的には、期待値問題における「得点」) の計算方法が、問題文中で定義される。

これらの特徴のうち、本研究では 3 番目の特徴に焦点を当て、事象を規定する条件記述を正しく解釈する問題に取り組む。本研究の特徴は、試行によって得られる結果の構造に着目し、この構造に基づいて条件記述を再帰的に解釈する点にある。

例題 1		サイコロを同時に 2 個振る。
	(1)	出た目の和が偶数となる確率を求めよ。
	(2)	出た目が 2 つとも偶数である確率を求めよ。
例題 2		サイコロを 2 回振る。 1 回目に 2 以下または 5 以上の目が出て、かつ 2 回目に奇数の目が出る確率を求めよ。

図 1　数学・確率問題の具体例

2 数学・確率問題の構造

対象とする数学・確率問題の具体例を図 1 に示す。一般に、問題文は次の 4 種類の記述から構成される。

1. **仮想世界の記述**
   どんなオブジェクトがいくつあるか
2. **試行の記述**
   標本空間はどのように規定されるか
3. **確率の計算要求**
   ある事象が起きる確率を求めることを要求する
4. **定義**
   特定の事象や、結果に対する属性を定義する

大学入試のセンター試験で出題される確率問題は、おおよそ、次の式で解ける。ここで、$|\cdot|$ は、集合の要素の数を表す。

$$確率 = \frac{|\,事象\,|}{|\,標本空間\,|} \tag{1}$$

標本空間が有限集合の場合は、要素を列挙すれば、この式の分母の値が求まる。さらに、事象を規定する条件を計算可能な形式に翻訳して標本空間の全要素に適用すれば、事象が求まり、分子の値が求まる。つまり、解くべき問題は、「出た目の和が偶数となる」のような日本語の条件記述を、この条件の成否を判定するプログラムに翻訳することになる。

3 4 種類の結果型

本研究では、条件記述を解釈するために、結果 (標本空間の要素) の内部構造 (階層構造) に着目する。この内部構造の型として、次に示す 4 種類を設

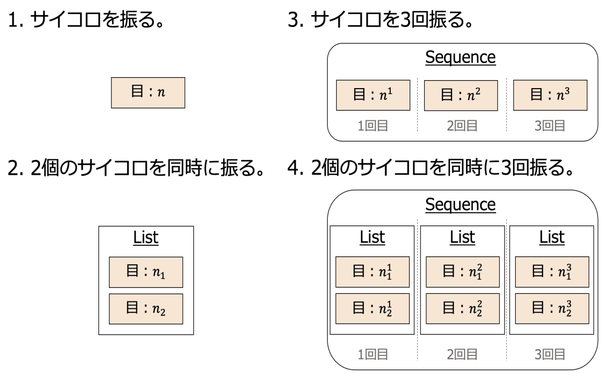

図 2　結果の内部構造の 4 種類の型

定する (図 2)。以下では、これを**結果型**と呼ぶ。

1. **O 型** (オブジェクト)
   オブジェクト 1 個
2. **LO 型** (リスト > オブジェクト)
   n 個のオブジェクトで構成されるリスト
3. **SO 型** (シーケンス > オブジェクト)
   k 個のオブジェクトで構成されるシーケンス
4. **SLO 型** (シーケンス > リスト > オブジェクト)
   n 個のオブジェクトで構成されるリストの、長さ k のシーケンス

上記の 4 種類の型は、試行の回数と 1 回の試行に関与するオブジェクトの個数によって定まる。

4　条件記述を構成する要素

条件記述を構成する要素を、すべて関数としてモデル化する。関数は、以下の 6 種類に分類する。

1. 定数 (属性値)
2. (オブジェクトの) 属性値の取得
3. (数値) 演算
4. (真偽値を返す) 述語
5. 述語の成立 (真値) に対する個数・回数・順序の制約
6. 論理演算

オブジェクト、リスト、シーケンスのそれぞれの階層に対して用意した基本関数とその主要な日本語表現 (キーワード) を、表 1、表 2、表 3 に示す。条件記述は、これらの基本関数の合成関数として表現する。

上記に示した 6 種類のうち、定数 (属性値) と属性値の取得は、オブジェクトに対してのみに適用可能な関数である[1)]。一方、数値演算は、リストとシーケンスに対して適用可能であり、述語、論理演算は、すべての階層において適用可能である。制約は、個数制約がリストに対して、回数制約と順序制約がシーケンスに対して適用可能である。個数制約と回数制約は、抽象的な関数としては同一であるが、日本語表現では異なる接尾辞が用いられるので、区別する。

1）オブジェクトとしては、サイコロ、玉、カードの 3 種類を想定し、それらのオブジェクトの属性として、番号・数字と色を想定した。

表 1　オブジェクトに対する基本関数

種類		具体例
定数 (属性値)		N(整数) 赤, 青, 黄, 緑, 白, 黒 (色)
属性値の取得		目 (サイコロ) 色, 数字, 番号 (玉・カード)
述語	非数値	である (=), でない (≠)
	整数	等しい (=), 等しくない (≠) より大きい (>), 以上 (≥) より小さい (<), 以下 (≤) 偶数, 奇数, 素数, 正, 負 n で割り切れる・ない n の倍数, n の約数
論理演算		かつ (AND), または (OR)

表 2　リストに対する基本関数

種類			具体例
数値演算			和, 積, 差, 最大値, 最小値
述語			(表 1 と同様)
制約	個数	特定	n つ, ちょうど n つ, n つだけ
		下限	n つ以上, 少なくとも n つ
		上限	n つ以下, 多くとも n つ
		全部	すべて, n つとも, n つの
論理演算			(表 1 と同様)

表 3　シーケンスに対する基本関数

種類			具体例
数値演算			(表 2 と同様)
述語			(表 2 と同様)
制約	回数	特定	n 回, ちょうど n 回, n 回だけ
		下限	n 回以上, 少なくとも n 回
		上限	n 回以下, 多くとも n 回
		全部	すべて, n 回とも, n 回の
	順序	特定	n 回目
		登場	n 回目に初めて, n 回目に m 度目
		連続	n 回続けて, n 回目から m 回目
論理演算			(表 2 と同様)

5　条件記述の解釈手順

本節では、日本語で書かれた条件記述を正しく解釈し、実行可能な条件判定プログラムに変換するアルゴリズムを示す。入力は、条件記述 (文字列) と結果型であり、出力は、それぞれのレベルに対して定義する合成関数である。

```
list.def 出た目の和が偶数となる (x)
  x.map{|o| o.send('出た目')}.sum.even?
end
object.def 出た目 (x)
  x.get_attr('目')
end

list.def 出た目が2つとも偶数になる (x)
  x.map{|o| o.send('出た目が偶数になる')}.all?
end
object.def 出た目が偶数になる (x)
  x.get_attr('目').even?
end
```

図 3　例題 1 の解釈

5.1　要素への分解

まず、条件記述を文節に分解し、キーワードを手掛かりに結合・分解・並び替えを行って、表 1–3 に示した要素 (日本語表現) の並びに変換する。たとえば、図 1 の例題 1 に対しては、以下のような要素の並びを得る。

(1)　a. [出た目の, 和が, 偶数となる]
　　b. [出た目が, 2 つとも, 偶数である]

これらの要素のうち、「出た目 (の・が)」は属性値の取得 (オブジェクト階層)、「2 つとも」は個数制約 (リスト階層) であることが一意に定まる。「和 (が)」は、リストとシーケンスの両方の階層に対して可能であるが、例題 1 の結果型は LO 型なので、リスト階層と定まる。残りの「偶数 (となる・である)」がどの階層に属するかは、この段階では定まらない。

5.2　構成要素の並びの解釈と関数の合成

次に、要素の並びを、結果型の階層構造にしたがって再帰的に解釈する。それぞれの階層では、与えられた要素の並びから、その階層の要素を抜き出し、残りを下位の階層に渡す (再帰呼び出し)。下位階層の処理結果として関数名が得られると、その関数名と抜き出した要素を組み合わせて合成関数を定義し、その関数名を上位の階層に返す。例題 1 に対して定義される合成関数を図 3 に示す。

要素への分解の過程で階層が一意に定まらなかった要素の階層は、以下のルールで決定する。

1. 数値演算の階層決定
   SLO 型の場合は、リスト階層と判定する。
2. 述語の階層決定
   (a) 要素の並びに数値演算が含まれる場合、その数値演算と同じ階層と判定する。
   (b) 要素の並びに制約表現が含まれる場合、その制約表現よりも下位の階層と判定する。

要素の並び (1a) をリスト階層で解釈する場合、「和 (が)」が数値演算 (リスト階層) なので、「偶数 (となる)」もリスト階層と判定する (ルール 2a)。残りの「出た目 (の)」は、下位のオブジェクト階層に渡される。一方、要素の並び (1b) をリスト階層で解釈する場合、個数制約「2 つとも」がリスト階層なので、残りの「出た目 (が)」と「偶数 (である)」を下位のオブジェクト階層に渡す (ルール 2b)。

5.3　論理演算を含む場合

条件記述に論理演算子 (AND と OR) に相当する記述が含まれる場合は、5.2 節の処理を行う前に、その論理演算が結びつける引数の範囲を決定する。AND と OR の引数は、必ず同じ階層の要素である。この制約に基づき、以下のように引数の範囲を定める。

1. 直前と直後の要素が同一種類の関数であれば、その両者を論理演算の引数とみなす。
2. そうでなければ、先頭から直前までと、直後から末尾までを、論理演算の引数とみなす。

たとえば、図 1 の例題 2 は、要素への分解によって、次のような並びが得られる。

(2) [1 回目に, 2 以下, <u>または</u>, 5 以上の, 目が, 出て, <u>かつ</u>, 2 回目に, 奇数の, 目が, 出る]

この並びに含まれる「または」の直前は「2 以下 (述語)」で、直後は「5 以上の (述語)」なので、この OR の引数は、これらの要素と解釈する。一方、「かつ」の直前は「出て (述語)」で、直後は「2 回目に (回数制約)」なので、トップレベルの AND と解釈する。以上により、

(3) [[1 回目に, [2 以下, OR, 5 以上の], 目が, 出て], AND, [2 回目に, 奇数の, 目が, 出る]]

が得られる。その後、括弧をはずし、以下のような論理演算を含まない形式に展開する。

(4)　a. [1 回目に, 2 以下の, 目が, 出る]
　　b. [1 回目に, 5 以上の, 目が, 出る]
　　c. [2 回目に, 奇数の, 目が, 出る]

この際、分割後の条件記述が自然になるように、整形する。たとえば、「5 以上<u>の</u>」より、「の」を補完して「2 以下<u>の</u>」とする。

```
seq.def １回目に２以下または５以上の目が出て、
                    かつ２回目に奇数の目が出る (x)
  ( x.send('１回目に２以下の目が出る') ||
    x.send('１回目に５以上の目が出る') ) &&
    x.send('２回目に奇数の目が出る')
end
seq.def １回目に２以下の目が出る (x)
  x.nth(1).send('２以下の目が出る')
end
object.def ２以下の目が出る (x)
  x.get_attr('目').leq(2)
end
```

図 4 例題 2 の解釈 (一部)

表 4 実験結果

	O 型	LO 型	SO 型	計
実験 1	2/2	24/28	6/8	32/38 (84%)
実験 2	1/1	29/31	11/16	41/48 (85%)

これらの条件記述のそれぞれに、5.2 節の処理を適用し、最後に論理演算で結合すれば、図 4 に示す合成関数が得られる。

6 実験と検討

アルゴリズムの開発に使用しなかった評価用データを用いて、作成したアルゴリズムの評価を行った。評価用データの出典を以下に示す。

実験 1 市販の問題集 2 冊 [4, 5]

実験 2 センター試験 (2000 年から 2020 年のうち、偶数年の本試験と追試験 22 回分)

対象とする問題は、問題の記述に使われるオブジェクトがサイコロ、玉、カードのいずれかで、かつ、結果型が 3 節に示した 4 種類のいずれかに該当する問題である。それぞれの問題からは、

1. 試行におけるオブジェクトの個数と回数
2. 確率を求める事象の条件記述

を抜き出し、これを入力とした。アルゴリズムが出力する解釈—どの階層にどのような合成関数を定義したか—を目視にて確認し、解釈の成否を判定した。

まず、実験 1 を行い、アルゴリズムの不備を修正した後、実験 2 を行った。実験結果を表 4 に示す。この表の分数の分母は条件記述の数、分子は正しく解釈できた数を表す。

表 4 に示すように、両方の実験で 8 割以上の正解率が得られた。この結果より、本アルゴリズムは、確率問題に現れる条件記述の大半を正しく解釈できると考えられる。結果型は、LO 型、SO 型の順に多く、SLO 型は 1 例も存在しなかった[2)]。

解釈に失敗した原因として最も多い (13 件中 9 件) のは、基本関数の不足である。たとえば「u が整数になる (確率)」という条件記述は、「整数 (になる)」という関数 (述語) を用意していなかったため、解釈できなかった。

本アルゴリズムにおいて、基本関数を追加すること自身は容易である。しかしながら、必要となる基本関数を、前もってすべて予想することは、かなり難しい。

特に問題となるのは、6 種類の基本関数の中の述語である。今回は、オブジェクトの属性値として色と番号を扱うので、非数値 (色) を対象とする述語と整数 (数字) を対象とする述語を用意した。しかし、先にあげた述語「整数 (になる)」は、試行によって得られた２つの数字 a, b から、(結果に対する) 新たな属性として分数 $u = a/b$ を定義し、「u が整数となる」確率を求める問題として出現した。試行によって得られる結果の構造をそのまま使うのではなく、結果にある写像を施し、その出力の確率を問う場合、あらかじめ用意すべき述語が十分に限定できない。

評価用データの条件記述のうち、軽微な修正では対処できないのは、次の 2 件である。

(5) a. 8 回目が終わった時点で赤玉がすべて取り出されている
　b. 赤玉がちょうど 8 回目ですべて取り出される

前者は「n 回目までに個数制約が満たされた状態になっている」ことを、後者は「ちょうど n 回目で個数制約が満たされた状態となる」ことを意味するが、これらの解釈を現在の枠組みに組み込むのは、それほど単純ではない。

参考文献

[1]新井紀子, 東中竜一郎（編）. 人工知能プロジェクト「ロボットは東大に入れるか」. 東京大学出版会, 2018.

[2]神谷翼, 松崎拓也, 佐藤理史. 数学確率文章題の自動解答システムの開発. 言語処理学会第 21 回年次大会発表論文集, pp. 365–368, 2015.

[3]神谷翼, 松崎拓也, 佐藤理史. 題材となるオブジェクトの抽象化による確率文章題の自動解答. 人工知能学会第 30 回全国大会論文集, 4B1-2, 2016.

[4]森谷慎司. 数学 場合の和・確率 分野別 標準問題精講. 旺文社, 2016.

2) 開発用データには、1 例存在した。

[5]松田聡平. 単元攻略 場合の数・確率 解法のパターン 30.
技術評論社, 2015.