

## 人工言語 Talkable Programming Language から述語論理への翻訳

## 7Q-8

松田 和之

駒宮 安男

明治大学

明治大学

## 1. はじめに

Talkable Programming Language (以下略して TPL と呼ぶ) とは, 分野を科学技術に関するものに限定して, 多くの国の人々の間での討論や論文の記述ができ, 必要なら計算機にもかけられることを目標とした人工言語で, 1979年に著者の一人である駒宮安男により提案され, 研究が進められている. 今回の発表では, TPL のこれまでの研究内容と本研究内容である構文解析された TPL 文から述語論理式への翻訳について述べる.

## 2. TPL の設計

以下に TPL の主な設計条件をあげる.

① 多くの国の人々にとって発音が容易

② 一字一音, 一音一字

文字, 単語は普及率の高い英語を参考に  
にする. 単語の構造は次の通りである.

(接頭辞) + 語幹 + 品詞語尾 + (派生辞)

語幹に単語の意味が含まれ, 品詞語尾には品詞情報を持たせる. 派生辞は品詞語尾で決めた品詞を別の品詞に変える情報を持たせる. 接頭辞は品詞を変えずにその単語に関連した意味の単語にする働きを持たせる.

③ 多義語を排除

多義語は曖昧な表現を生じる.

④ 不特定話者, 連続音声の容易な認識

TPL は計算機入力をするのに, 音声入力をを用いる. 音節は原則的に CV 音節 (C: 子音, V: 母音) とする.

⑤ 述語論理が TPL のメタ言語

科学技術上の討論においては, その根底に論理があるので述語論理を基盤にする.

## 3. TPL から述語論理への翻訳

TPL 文を述語論理式に翻訳する利点は次の二つの理由からである. TPL 文から述語論理式への意味表現が曖昧さを含まないこと, 述語論理は TPL を計算機用言語に翻訳する第一段階となることである.

## # 述語論理へ翻訳する概念及び技法

## &lt; 構成性の原理 &gt;

TPL を述語論理へ翻訳する際に, 構成性の原理という考え方を基盤としている. 構成性の原理とは, 全体の意味はその構成部分の意味から一意的に決定されるという考え方で, TPL 文の意味は構文解析された文の構成部分の要素の意味によって決定される.

<  $\lambda$  (ラムダ) 演算 >

構成性の原理に基づいて TPL 文を述語論理式に翻訳する技法に, A. Church によって導入された  $\lambda$  演算を用いる. ここでは技法だけが必要なので, 特に細かな概念は省略する.

$$f(x) = \text{walk}(x)$$

なる関数  $f(x)$  において  $f$  自身を直接表現し, 変数が  $x$  であることを明示するために  $\lambda$  演算子を用いて

$$f = \lambda x [\text{walk}(x)]$$

と書く. この式の右辺は, 「"walk" という性質を持つ集合」を表し, これに個体

定項である "John" を代入すると、次のようになる。

$$\begin{aligned} f(\text{John}) &= \lambda x[\text{walk}(x)](\text{John}) \\ &= \text{walk}(\text{John}) \end{aligned}$$

第二式から第三式への変換は、式 walk(x) 内の変数 x のすべての現れを "John" に置き換える操作をしており、この操作を λ 演算という。

では、実際に T P L 文の例をあげて、述語論理式への翻訳について述べる。翻訳に関しては R. Montague の PTQ 文法を参考にしている。

T P L 文: Remo Rarunogera sutadoke.

日本語訳: 「すべての学生は勉強する。」

remo ... DET (限定詞:  $\forall$ )  
Rarunogera ... N (名詞: 学生)  
sutadoke ... V (動詞: 勉強する)

この式の構文解析木は図 1 の様になる。

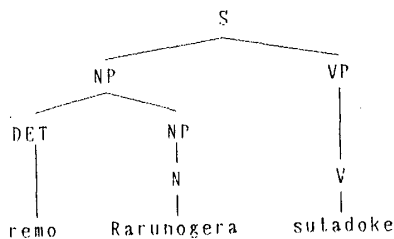


図 1 構文解析木

翻訳に先立って T P L の単語一つ一つに λ 演算子を用いた基本表現 (基本論理式) を与える。

$$\begin{aligned} \text{remo} &\Rightarrow \lambda P \lambda Q [\forall x (P(x) \rightarrow Q(x))] \\ \text{Rarunogera} &\Rightarrow \text{Rarunogera} \\ \text{sutadoke} &\Rightarrow \lambda a [\text{sutadoke}(a)] \end{aligned}$$

そして、構成性の原理に基づき、T P L の構文規則と一対一に対応する翻訳規則を設ける。X と Y から構文規則によって Z が構成されたことを

$$Z \rightarrow X Y$$

また、翻訳規則によって A の論理表現 [A] が、B の論理表現 [B] から翻訳されること

を

$$[A] \Rightarrow [B]$$

と表わすとすると、例文に適用される規則及び翻訳は次のようになる。

$$\begin{aligned} \text{NP} &\rightarrow \text{DET NP}, [\text{NP}] \Rightarrow [\text{DET}][[\text{NP}]] \\ \text{remo rarunogera} & \\ &\Rightarrow \lambda P \lambda Q [\forall x (P(x) \rightarrow Q(x))](\text{Rarunogera}) \\ &\Rightarrow \lambda Q [\forall x (\text{Rarunogera}(x) \rightarrow Q(x))] \\ \text{S} &\rightarrow \text{NP VP}, [\text{S}] \Rightarrow [\text{NP}][[\text{VP}]] \\ \text{remo Rarunogera sutadoke} & \\ &\Rightarrow \lambda Q [\forall x (\text{Rarunogera}(x) \rightarrow Q(x))] \\ &\quad (\lambda a [\text{sutadoke}(a)]) \\ &\Rightarrow \forall x (\text{Rarunogera}(x) \\ &\quad \rightarrow \lambda a [\text{sutadoke}(a)](x)) \\ &\Rightarrow \forall x (\text{Rarunogera}(x) \rightarrow \text{sutadoke}(x)) \end{aligned}$$

#### 4. まとめ

T P L は疑問や時制を含む文も作成されているが、述語論理の枠内では表現できない。しかし、T P L の意味表現のため、T P L 専用の論理体系を構築し、前述の通り、T P L がプログラム言語として使用できることが必要である。

#### 5. 参考文献

- (1) 駒宮: "「しゃべれるプログラム言語」の提案", 信学誌 vol.62, No.12, pp.1402-1403(1979)
- (2) 駒宮: "「しゃべれるプログラム言語」の提案その2", 信学誌 vol.66, No.6, pp.609-611(1983)
- (3) D.R.Dowty 他著, 井口他訳: "モック'ユ-意味論入門", 三修社, 1987年10月15日, 第1版発行