

日本語にあった述語

大村伸一

平成 31 年 2 月 10 日

1 概要

例えば次の文を考える。

例文 1 僕は君が好き

これは次のように述語化できる。

述語化 1.1 好き (僕, 君)

しかし、日本語では語順を変えることができるので、

例文 1' 君が僕は好き

と修正したとき、これを述語化 1.1 に変換するのは面倒くさい。

日本語で語順が変えやすいのは、助詞がその文法的な意味を肩代わりしていて、英語では語順で文法の意味を表しているということだと思う。

そこで、助詞の「は」や「が」をキーワードとしたこのような書き方をしてはどうだろうか。

述語化 1.2 好き (は=僕, が=君)

こうすると、例文 1' は次のようになる。

述語化 1.2' 好き (が=君, は=僕)

このような述語に対して、unify を、キーワード「は」「が」での一致と考えると、語順が問題にならなくなる。

キーワードを「助詞」と考えるのは、英文では文法のカテゴリーだと考えることに相当するので

例文 1e I love you.

は次のように変換できる。

述語化 1e Love(I, you)

述語化 1.1e Love(S=I, O=you)

こうなると、述語記号が動詞であるというのも、たまたまのような気がするので次のようにもかける。

述語化 1.2e $\text{Sent}(I, \text{love}, \text{you})$

述語化 1.3e $\text{Sent}(S=I, V=\text{love}, O=\text{you})$

ここでは、述語は文を表していると考えているが、それは、キーワードが文法概念だからで自然だと思う。

ここまできると、データベースと似てきていて、列名がここのキーワードになる。

DB の場合は、格納されたデータと Query のマッチングを行うので、unification とは異なるがマッチングするときは SQL を使うのであり、キーワードの順番は問題にならなくなっている。

1.1 論理式

次に論理式の場合を考える。

例文 2 君が僕にりんごを渡したので、僕はりんごを持っている

これは「渡す」「持つ」という述語を考えると次のような論理式になる。

述語化 2 渡す (が=君, に=僕, を=りんご) 持つ (は=僕, を=りんご)

これを一般化するとこうなる。

述語化 2 $\text{who1}, \text{who2}, \text{obj} . \text{渡す} (\text{が}=\text{who1}, \text{に}=\text{who2}, \text{を}=\text{obj}) \quad \text{持つ} (\text{は}=\text{who2}, \text{を}=\text{obj})$

日本語の parse の手を抜くため、助詞を”ha”, ”ga”, ”ni”と書くとよさそう

1.2 文法と叙述

このように、文法だけでは、叙述はできない。

2 実装

実装する場合、述語の引数を $\text{Dict}(\text{Julia の})$ として持てばよい。

代入文も Dict とする方法を先に考えたので、何もかも Dict にできないのかなと思う。

2.1 存在しない引数

今度は、「私はリンゴを食べる」という文からはじめる。

文 1 文 ($ha=私, wo=リンゴ, ru=食べ$)

文 2 [X]. 文 ($ha=私, wo=X, ru=食べ$)

文 3 文 ($ha=私, ru=食べ$)

文 1 は、「私はリンゴを食べる」という文だが、その「リンゴ」の部分を変数にすると文 2 になる。文 1 と文 2 を nify すると $\{リンゴ \rightarrow X\}$ という代入が得られる。

これを適用すると

$$\begin{aligned} \text{文}(ha = 私, wo = \text{リンゴ}, ru = \text{食べ}) \bullet \{リンゴ \rightarrow X\} = \\ \forall X \text{ 文}(ha = 私, wo = X, ru = \text{食べ}) \bullet \{リンゴ \rightarrow X\} \end{aligned}$$

では「文 3」のように「私は食べる」を述語化するとすると、unification はどうなるべきか。

「食べる」という述語には、 $ha-wo-ru$ の 3 つの要素が必要だと考えるならば、 $who=X$ を補った「文 2」が正しい述語化のように見える。文 1 と文 3 の unification で、文に書かれていない変数 X が登場してよいのだろうか。

あえて $wo=?$ を補うのだとすると、 $\{wo = \text{リンゴ}\}$ が代入に含まれるということか。これは $\{リンゴ \rightarrow wo\}$ ではない。

代入は、文 1 と文 3 の両方に適用して、同じ文がでてくるものでなくてはならないので

$$\begin{aligned} \text{文}(ha = 私, wo = \text{リンゴ}, ru = \text{食べ}) \bullet \{wo = \text{リンゴ}\} = \\ \text{文}(ha = 私, ru = \text{食べ}) \bullet \{wo = \text{リンゴ}\} \end{aligned}$$

存在しない引数のリストに $\{wo = \text{リンゴ}\}$ を適用すると、追加されるという規則になりそう。

3 はたしてどうか

日本語の述語化を考えたとき、語順を考慮して変換できるかどうか。

「孤独な文法」で考えたように、それぞれが独自のルールを使うようなシステムでは parse の段階で、日本語文法を仮定したくなかったので、このようなことを考え始めた。

助詞に注目するのは、助詞が頻繁に出現するので、それを parse のキーワードとすることに大きな違和感はなかったので、こちらのほうがよいと思えた。

すなくとも、parse の段階で日本語文法を考慮し、その意味に基づいて引数順序を決めるのは好まない。

この方式では、語順ではなく書かれた順番で述語の引数を書いていけば日本語の特殊なルールを考慮しなくてもよくなる。

英文も同じようにキーワード付きにできるが、その場合、文法を考慮して順番を決めるだけなのであまりメリットはないと思う。

また、DB の SQL 文は、このような考えを持っていると思うが、先に書いたように unification がないのでこの話とは様相が違おうだろう。

DB は、Fact と Query に別れていて、論理式は Query にしか書けない。

DB の中に Query を置くのは Stored Program だろうか。

Prolog ではないので、プログラムに相当する論理式はあるが、論理式はプログラムではない。