

数学というか、大学入試レベルの理系問題ってそんな難しい操作が必要なことはないと思えるのですがどうでしょうか。群論と環論とか、対応とか、代入法とかそんなレベルの操作で実現できる数式処理ではないでしょうか。

数式をオブジェクトで表現し、意味オブジェクトのネットワークで数式の下部構造を表現できると思うのですね。例えば、意味オブジェクトは次のようなタプルです。

```
(name;コマンド,格,意味オブジェクト,格,意味オブジェクト,・・・);
```

ここに、name はこの式の参照名で、コマンドは積分とか足し算とかの演算で、格は、積分ですと積分範囲とかありますよね。意味オブジェクトは変数であったり、式であったりします。

```
( sum;"+", "object_case",X,"object_case",Y );
```

```
( ix;"  ", "object_case",X);
```

```
( iy;"  ", "object_case",Y);
```

積分の分配法則は次のように表現されます。

```
( integral;"  ", "object_case",sum ) :=( anonimus;"+", "object_case",ix,"object_case",iy);
```

ネットワークとして表現しようという理由は、式のパターンを動的に、いろいろの深さのレベルで発見して式オブジェクトとして表現したいからです。そんなこと、Prolog とか LISP では表現できないでしょう。

おわり