コード生成

更新された CodeGeneration パッケージでは、新しく Python と Perl への変換をサポートし、MATLAB® のサポートが拡大され、また出力を制御するためのオプションも追加されます。

さらに、新しいコード生成アシスタントによって、便利なグラフィカルインターフェースを CodeGeneration パッケージに提供しています。

CodeGeneration[Python]

新しい Python 変換機能では、スタンドアロンで実行するためのコードを Python 3 言語で生成します。CodeGeneration のほかの変換機能と同様、CodeGeneration[Python] では、式をコードフラグメントに変換できます。

```
with(CodeGeneration):
```

$$Python\left(\sqrt{a^2+b^2+c^2}\right)$$

```
cg23 = math.sqrt(a ** 2 + b ** 2 + c ** 2)
```

プロシージャや、より大きなプログラムを変換することもできます。

$$Python(m \rightarrow add(ithprime(i), i = 1..m))$$

```
import sympy

def cg24 (m):
    r = 0
    for i in range(1, m):
        r = r + sympy.prime(i)
    return(r)
```

コア Python 3 言語のサポートに加え、CodeGeneration[Python] では多くの Maple データ構造および関数を変換することができます。これには、線形代数および特殊関数の多くのルーチンから、それに相当する科学技術計算の一般的な numpy および scipy ライブラリまでが含まれます。

Maple オブジェクト	生成された Python 出力

$\begin{bmatrix} -59 & -68 & 16 & -95 \\ 12 & -67 & 9 & -20 \\ -62 & 22 & 99 & -25 \\ -33 & 14 & 60 & 51 \end{bmatrix}$	cg25 = numpy.mat([[-59,-68,16,-95],[12,-67,9,-20],[-62,22,99,-25],[-33,14,60,51]])
$\begin{cases} -v^3+3v+1 & v<1 \\ 2v^3-9v^2+12v-2 & v<2 \\ -v^3+9v^2-24v+22 & otherwise \end{cases}$	cg26 = -v ** 3 + 3 * v + 1 if v < 1 else 2 * v ** 3 - 9 * v ** 2 + 12 * v - 2 if v < 2 else -v ** 3 + 9 * v ** 2 - 24 * v + 22
$(M, n) \rightarrow M - x\, LinearAlgebra{:-}IdentityMatrix(n)$	import numpy cg27 = lambda M,n: M - x * numpy.eye(n)
$\_C1\ \mathrm{BesselJ}(\nu, x) + \_C2\ \mathrm{BesselY}(\nu, x)$	cg28 = _C1 * scipy.special.jv(nu, x) + _C2 * scipy.special.yv(nu, x)
$\mathrm{LambertW}(100, \pi + e)$	cg29 = scipy.special.lambertw(math.pi + math.e, 100)

CodeGeneration[Perl]

新しい Perl 変換機能では、後で Perl 5 言語の既存のコードに含めることを想定し、Maple でコードをプロトタイプ化することができます。
Maple、Perl、および Python は、すべて明確にタイプ分けされていないインタープリタ型プログラミング言語であるため、Maple プログラムは多くの場合とても自然に Perl または Python で表されます。

with(*CodeGeneration*) :

$Perl\left(\sqrt{b^2-4\,a\,c}\right)$

```
$cg30 = sqrt(-4 * $a * $c + $b ** 2);
```

特に、Maple の文字列処理のための高度なツールの多くは、同等の Perl 言語に変換することができます。

replaceFoo := **proc**(*s*) **if** *StringTools*:-*RegMatch*("N[aeiou]*dl[ae]*", *s*) **then** "Match" **else** *cat*("No match: ", *s*) **end if**; **end proc**

proc(*s*) **if** *StringTools*:-*RegMatch*("N[aeiou]*dl[ae]*", *s*) **then** "Match" **else** *cat*("No match: ", *s*) **end if end proc** (2.1

replaceFoo("Needle"); *replaceFoo*("Haystack")

"Match"
"No match: Haystack"

(2.2

Perl(*replaceFoo*)

```
#!/usr/bin/perl

sub replaceFoo
{
  local($s) = @_;
  if (($s =~ m/N[aeiou]*dl[ae]*/)) {
    return("Match");
  } else {
    return("No match: " . $s);
  }
}
```

参照

CodeGeneration[Perl]、CodeGeneration[Python]、Maple 18 の言語とプログラミングの拡張