Polyhedral Calculator

This page provides examples of how to use the calculator feature in the AlphaZ system. The calculator is simply an interface to the underlying polyhedral libraries. There are a number of commands available in the script interface with prefix "calc_" which stands for calculator commands. (Eclipse has a very good context-sensitive assist feature; the default key binding is ctrl+space)

The calculator manipulates Domain and AffineFunction objects that are created by the following commands. Note that there is no notion of parameter domain in the calculator, so when you manipulate parameterized domains that come form an Alphabets program, and functions you need to add these indices explicitly rather than blind cut-and-paste.

ReadDomain

Parses a domain in alphabets syntax and returns a Domain object.

```
dom = calc_ReadDomain("{i,j,k|0<=i<10 && 0<=j<10}");
```

ReadFunction

Parses an affine function in alphabets syntax and returns a AffineFunction object.

```
func = calc_ReadFunction("(i,j,k->i,j+k)");
```

echo

These objects (and any other objects) can be printed out to the console by the echo command.

```
dom = calc_ReadDomain("{i,j,k|0<=i<10 && 0<=j<10}");
echo(dom);
```

Domain Operations

Intersection

Returns the intersection of two domains.

```
domA = calc_ReadDomain("{i,j,k|0<=i<10 && 0<=j<10}");
domB = calc_ReadDomain("{i,j,k|0<=i<10 && 0<=k<10}");
echo(calc_Intersection(domA,domB));
```

Union

Returns the union of two domains.

```
domA = calc_ReadDomain("{i,j,k|0<=i<10 && 0<=j<10}");
domB = calc_ReadDomain("{i,j,k|2<=i<12 && 5<j<20}");
echo(calc_Union(domA,domB));
```

Image

Returns the image of a domain by an affine function.

```
dom = calc_ReadDomain("{i,j,k|0<=i<10 && 0<=j<10}");
fun = calc_ReadFunction("(i,j,k->i-j,i,j)");
echo(calc_Image(fun,dom));
```

PreImage

Returns the pre-image of a domain by an affine function.

```
dom = calc_ReadDomain("{i,j,k|0<=i<10 && 0<=j<10}");
fun = calc_ReadFunction("(i,j,k->i+k,j,k)");
echo(calc_PreImage(fun, dom));
```

Difference

Returns the difference of two domains.

```
domA = calc_ReadDomain("{i,j,k|0<=i<10 && 0<=j<10}");
domB = calc_ReadDomain("{i,j,k|0<=i<10 && 0<=j<5}");
echo(calc_Difference(domA, domB));
```

IsEmpty

Returns true if the given domain is empty.

```
dom = calc_ReadDomain("{i|0<=i<10 && i>10}");
echo(calc_IsEmpty(dom));
```

IsEquivalent

Returns true if the given domains are equivalent.

```
dom = calc_ReadDomain("{i,j,k|0<=i<10 && 0<=j<10}");
fun = calc_ReadFunction("(i,j,k->i+k,i,j)");
domI = calc_Image(fun,dom);
domIPI = calc_PreImage(fun,domI);
echo(calc_IsEquivalent(dom,domIPI));
```

Function Operations

Compose

Returns f = f1 o f2. f(z) = f1(f2(z)).

```
f1 = calc_ReadFunction("(x,y->x,y,x+y)");
f2 = calc_ReadFunction("(i,j,k->i,j)");
echo(calc_Compose(f1, f2));
```

Inverse

Returns the inverse of an affine function if it exists.

```
f = calc_ReadFunction("(x,y->x,x+y)");
echo(calc_Inverse(f));
```

InverseInContext

Returns the inverse of an affine function in the context of a given domain if it exists.

```
f = calc_ReadFunction("(x,y->x)");
context = calc_ReadDomain("{i,j|i==j}");
echo(calc_InverseInContext(context, f));
```

IsEquivalent

Returns true if the given functions are equivalent.

```
f1 = calc_ReadFunction("(x,y->x,y,x+y)");
```

```
f2 = calc_ReadFunction("(i,j,k->i,j)");
echo(calc_IsEquivalent(calc_Compose(f1, f2), calc_Join(f2, f1)));
```

From:

<https://www.cs.colostate.edu/AlphaZ/wiki/> - **AlphaZ**

Permanent link:

https://www.cs.colostate.edu/AlphaZ/wiki/doku.php?id=calculator

Last update: **2017/04/19 13:31**