Liberal Syntax

Less pain in Tool

Thierry Bossy
Gregoire Hirt
Rafael Pizarro

Semicolon inference

```
def meaninglessGetter() : Matrix2 = { //before
  var ma : Matrix2; ma.init(1,0,0,1);
  do(this.print());
  return ma.times(2).plus(bias);
}
```

```
def meaninglessGetter() : Matrix2 = { //after
  var ma : Matrix2; ma.init(1,0,0,1)
  do(this.print())
  return ma.times(2).plus(bias)
}
```

Allow to omit semi-colon if they are at line end.

Surely easily doable by allowing end-of-line to replace them in the grammar.

Thierry Bossy
Gregoire Hirt
Rafael Pizarro

Methods as infix operators

```
def meaninglessGetter() : Matrix2 = { //before
  var ma : Matrix2; ma.init(1,0,0,1);
  do(this.print());
  return ma.times(2).plus(bias);
}
```

```
def meaninglessGetter() : Matrix2 = { //after
  var ma : Matrix2; ma.init(1,0,0,1);
  do(this.print());
  return ma * 2 + bias;
}
```

Allow to write arithmetics and other expressions for custom types in a much more intuitive way.

Probably the most important change, many work on the grammar.

Thierry Bossy
Gregoire Hirt
Rafael Pizarro

Parameterless calls

```
def meaninglessGetter() : Matrix2 = { //before
  var ma : Matrix2; ma.init(1,0,0,1);
  do(this.print());
  return ma.times(2).plus(bias);
}
```

```
def meaninglessGetter : Matrix2 = { //after
  var ma : Matrix2; ma.init(1,0,0,1);
  do(this.print);
  return ma.times(2).plus(bias);
}
```

Since functions are not first-class objects in Tool we allow function to be called just by writing them if they take no parameters.

Note there's no more way to discriminate a method call from a simple identifier.

Thierry Bossy
Gregoire Hirt
Rafael Pizarro

No need for this and do()

```
def meaninglessGetter() : Matrix2 = { //before
  var ma : Matrix2; ma.init(1,0,0,1);
  do(this.print());
  return ma.times(2).plus(bias);
}
```

```
def meaninglessGetter() : Matrix2 = { //after
  var ma : Matrix2; ma.init(1,0,0,1);
  print();
  return ma.times(2).plus(bias);
}
```

Don’t use “**this**” anymore to call a method of the current object.

Don’t use “**do**” anymore to evaluate an expression.

“**this**” is still usable to discriminate between members and parameters and to pass instance to other objects.

Thierry Bossy
Gregoire Hirt
Rafael Pizarro

All at once

```
def meaninglessGetter() : Matrix2 = { //before
  var ma : Matrix2; ma.init(1,0,0,1);
  do(this.print());
  return ma.times(2).plus(bias);
}
```

```
def meaninglessGetter : Matrix2 = { //after
  var ma : Matrix2; ma.init(1,0,0,1)
  print
  return ma * 2 + bias
}
```

All these changes make code less cluttered and more readable.

Thierry Bossy
Gregoire Hirt
Rafael Pizarro