
Optimizer Model Documentation

Release dev

Alex Gaynor

December 19, 2016

1	Contents	3
1.1	Usage	3

`optimizer-model` is a Python package which implements the optimizer for a compiler. It's designed for use with tracing JITs and works over a linear sequence of operations.

1.1 Usage

class `optimizer.Optimizer` (*optimization_classes=[]*)

This is the core of `optimizer-model`. This performs “on-line” optimization, meaning that operations are optimized as they’re added, not all at once at the end. By default it actually performs no optimizations, it just records the operations which are added, but you can easily add them:

```
from optimizer import Optimizer

opt = Optimizer()
opt = Optimizer([optimization, classes, here])
```

add_input (*tp*)

This adds input variables to a trace.

```
from optimizer import Types

i0 = opt.add_input(Types.INT)
i1 = opt.add_input(Types.INT)
```

add_operation (*op, args, descr=None*)

Adds an operation to the sequence of operations, and runs it through all of the optimizations. Returns a representation of the result.

```
from optimizer import Operations

i2 = opt.add_operation(Operations.INT_ADD, [i0, i1])
opt.add_operation(Operations.FINISH, [i2])
```

build_operations ()

Returns a sequence of all of the operations, after optimizations:

```
ops = opt.build_operations()
assert len(ops) == 3
```

1.1.1 Optimizations

Out of the box, an `Optimizer` doesn’t actually run any optimizations, it just records the operations. However, `optimizer-model` includes many optimizations which can be plugged in.

The optimizations included with `optimizer-model` are:

class `optimizer.optimizations.IntBounds`

Keeps track of the possible bounds for an integer and propagates that data.

class `optimizer.optimizations.ConstantFold`

Performs constant folding on operations which do not have side-effects and which have all-constant arguments.

class `optimizer.optimizations.GuardPropagation`

Promotes values to be constant after they've been guarded against.

class `optimizer.optimizations.Virtualize`

Removes allocations which do not escape the trace, and removes `GETFIELD` and `SETFIELD` operations on objects whose allocation has been removed.

A

`add_input()` (`optimizer.Optimizer` method), 3

`add_operation()` (`optimizer.Optimizer` method), 3

B

`build_operations()` (`optimizer.Optimizer` method), 3

O

`optimizer.optimizations.ConstantFold` (built-in class), 4

`optimizer.optimizations.GuardPropagation` (built-in class), 4

`optimizer.optimizations.IntBounds` (built-in class), 3

`optimizer.optimizations.Virtualize` (built-in class), 4

`optimizer.Optimizer` (built-in class), 3