

Active Expressions in Python

Arne Boockmeyer

arne.boockmeyer@student.hpi.de

03.07.2018

Reactive Programming

Summer term 2018

Software Architecture Group

Hasso Plattner Institute

Demo

Implementation

Active Expression

```
aexpr(lambda: expr_to_monitor).onChange(lambda new_value: do_sth())
```



Difficult part!

When did something changed?

Fairly easy!

If change detected: Execute lambda.

Two Parts: **What?** **When?**

Example (What)

```
class Example:  
    def __init__(self):  
        self.f = 5  
        self.g = 10  
  
    def method(self):  
        t = self.f + 2  
        return t + self.get_g()  
  
    def get_g(self):  
        return self.g  
  
tmp = Example()  
aexpr(lambda: tmp.method())
```

Static Byte-Code Interpretation (What)

- Using library `dis` to get operations:

Method:

```
def method(self):  
    t = self.f + 2  
    return t + self.get_g()
```

`method.__code__`

```
<code object method at  
0x7f9dacfcdb0, file  
"file.py", line 47>
```

`dis.dis()`

Operations:

LOAD_FAST (self)
LOAD_ATTR (f)
LOAD_CONST (2)
BINARY_ADD
STORE_FAST (t)

LOAD_FAST (t)
LOAD_FAST (self)
LOAD_ATTR (get_g)
CALL_FUNCTION
BINARY_ADD
RETURN_VALUE

callable's
compiled to

Op-Codes as Bytes (`co_code`):

```
b'|\x00\x00j\x00\x00d  
\x01\x00\x17}...'
```

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

Instruction Queue:

Object Stack:



Variable Mapping:

Name	Value

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

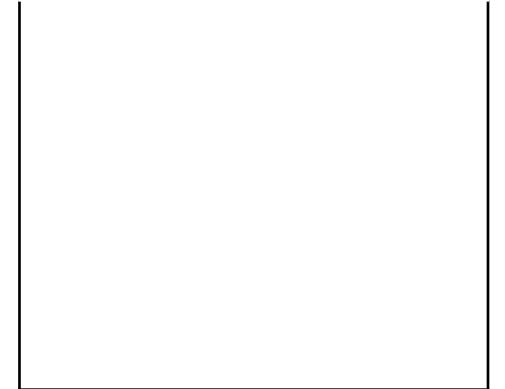
    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())
    
```

Instruction Queue:

Object Stack:



Variable Mapping:

Name	Value

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

Instruction Queue:

→ LOAD_GLOBAL (tmp)
 LOAD_ATTR (method)
 CALL_FUNCTION
 RETURN VALUE

Object Stack:

tmp from
 globals()



Variable Mapping:

Name	Value
self	None

Example (What)

```
class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())
```

Instruction Queue:

→ LOAD_GLOBAL (tmp)
 LOAD_ATTR (method)
 CALL_FUNCTION
 RETURN VALUE

Object Stack:

tmp from
 globals()



tmp

Variable Mapping:

Name	Value
self	None

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

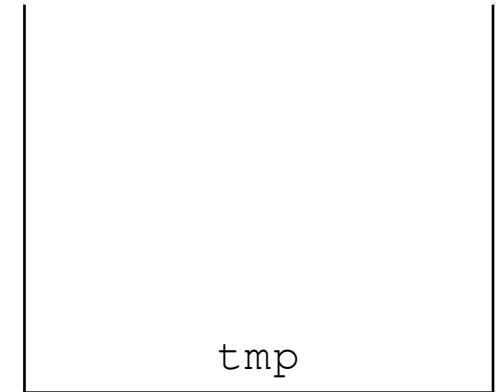
Instruction Queue:

 LOAD_GLOBAL (tmp)
 LOAD_ATTR (method)
 CALL_FUNCTION
 RETURN VALUE

Object Stack:

pull object from stack

 push attribute



Variable Mapping:

Name	Value
self	None

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

Instruction Queue:

 LOAD_GLOBAL (tmp)
 LOAD_ATTR (method)
 CALL_FUNCTION
 RETURN VALUE

Object Stack:

pull object from stack

 push attribute

method (tmp)

Variable Mapping:

Name	Value
self	None

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

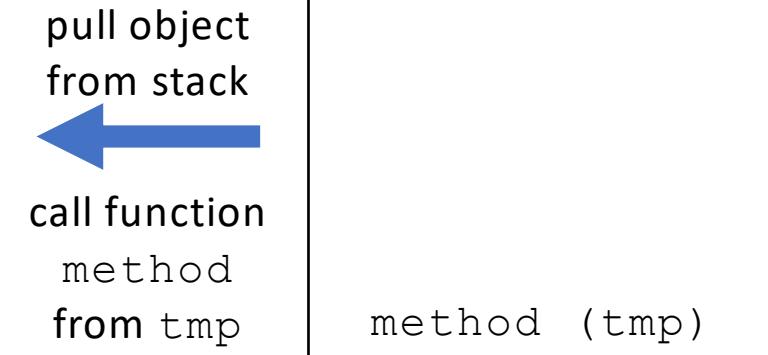
tmp = Example()
aexpr(lambda: tmp.method())

```

Instruction Queue:

 LOAD_GLOBAL (tmp)
 LOAD_ATTR (method)
CALL_FUNCTION
 RETURN VALUE

Object Stack:



Variable Mapping:

Name	Value
self	None

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

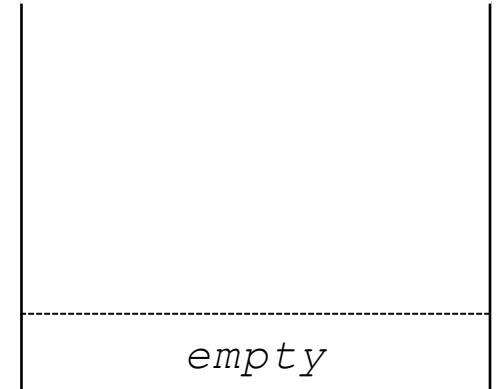
```

Instruction Queue:

→ LOAD_FAST (self)
 LOAD_ATTR (f)
 LOAD_CONST (2)
 BINARY_ADD
 STORE_FAST (t)
 LOAD_FAST (t)
 LOAD_FAST (self)
 LOAD_ATTR (get_g)
 CALL_FUNCTION
 BINARY_ADD
 RETURN_VALUE

pull self
from variable
mapping

Object Stack:



Variable Mapping:

Name	Value
self	tmp

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

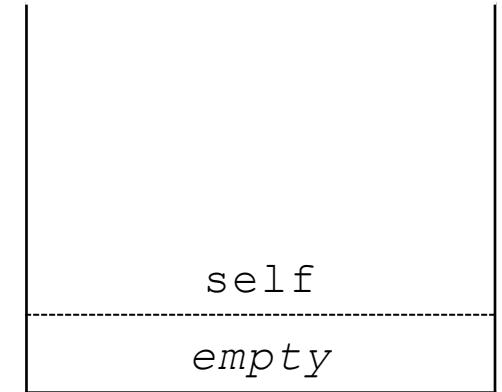
```

Instruction Queue:

→ LOAD_FAST (self)
 LOAD_ATTR (f)
 LOAD_CONST (2)
 BINARY_ADD
 STORE_FAST (t)
 LOAD_FAST (t)
 LOAD_FAST (self)
 LOAD_ATTR (get_g)
 CALL_FUNCTION
 BINARY_ADD
 RETURN_VALUE

pull self
from variable
mapping

Object Stack:



Variable Mapping:

Name	Value
self	tmp

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

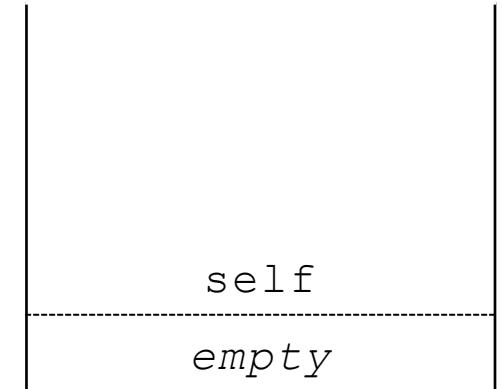
Instruction Queue:

 LOAD_FAST (self)
 LOAD_ATTR (f)
 LOAD_CONST (2)
 BINARY_ADD
 STORE_FAST (t)
 LOAD_FAST (t)
 LOAD_FAST (self)
 LOAD_ATTR (get_g)
 CALL_FUNCTION
 BINARY_ADD
 RETURN_VALUE

Object Stack:

pull object
 from stack

 push attribute
 to stack



Variable Mapping:

Name	Value
self	tmp

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

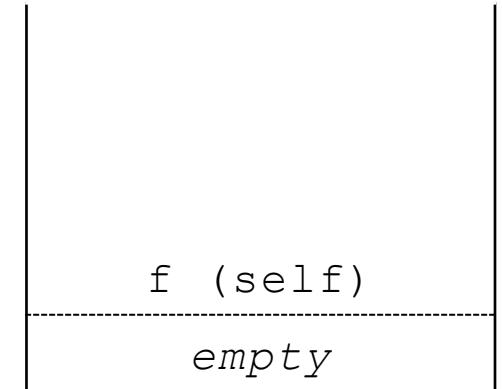
Instruction Queue:

 LOAD_FAST (self)
 LOAD_ATTR (f)
 LOAD_CONST (2)
 BINARY_ADD
 STORE_FAST (t)
 LOAD_FAST (t)
 LOAD_FAST (self)
 LOAD_ATTR (get_g)
 CALL_FUNCTION
 BINARY_ADD
 RETURN_VALUE

Object Stack:

pull object
 from stack

 push attribute
 to stack



Variable Mapping:

Name	Value
self	tmp

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

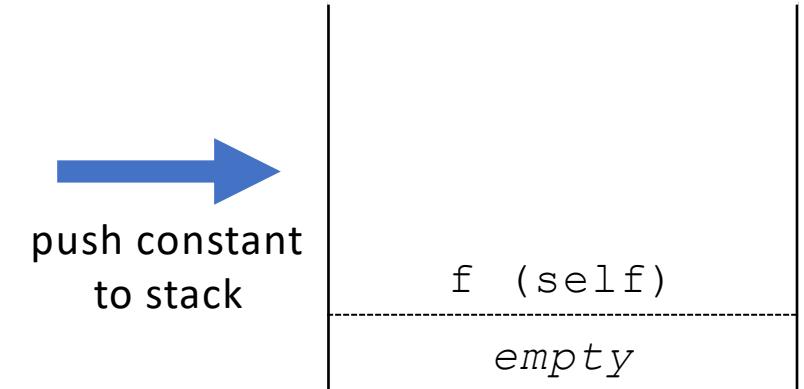
tmp = Example()
aexpr(lambda: tmp.method())

```

Instruction Queue:

 LOAD_FAST (self)
 LOAD_ATTR (f)
 LOAD_CONST (2)
 BINARY_ADD
 STORE_FAST (t)
 LOAD_FAST (t)
 LOAD_FAST (self)
 LOAD_ATTR (get_g)
 CALL_FUNCTION
 BINARY_ADD
 RETURN_VALUE

Object Stack:



Variable Mapping:

Name	Value
self	tmp

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

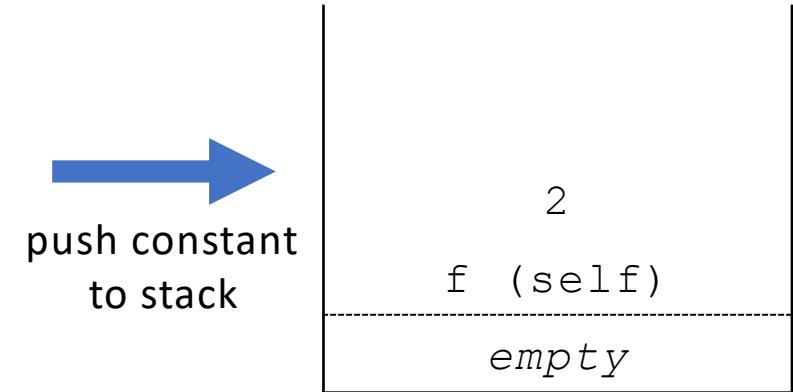
Instruction Queue:

```

LOAD_FAST (self)
LOAD_ATTR (f)
LOAD_CONST (2)
BINARY_ADD
STORE_FAST (t)
LOAD_FAST (t)
LOAD_FAST (self)
LOAD_ATTR (get_g)
CALL_FUNCTION
BINARY_ADD
RETURN_VALUE

```

Object Stack:



Variable Mapping:

Name	Value
self	tmp

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

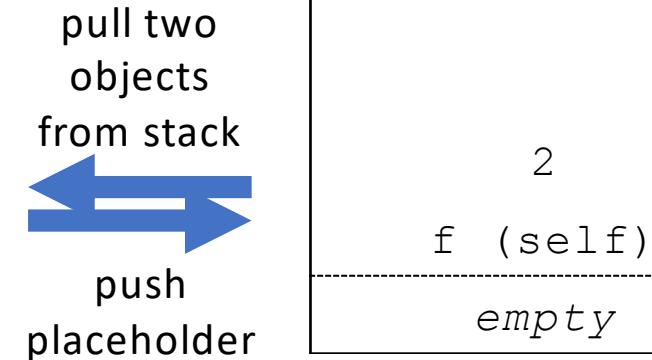
tmp = Example()
aexpr(lambda: tmp.method())

```

Instruction Queue:

 LOAD_FAST (self)
 LOAD_ATTR (f)
 LOAD_CONST (2)
 BINARY_ADD
 STORE_FAST (t)
 LOAD_FAST (t)
 LOAD_FAST (self)
 LOAD_ATTR (get_g)
 CALL_FUNCTION
 BINARY_ADD
 RETURN_VALUE

Object Stack:



 pull two
objects
from stack

 push
placeholder

Variable Mapping:

Name	Value
self	tmp

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

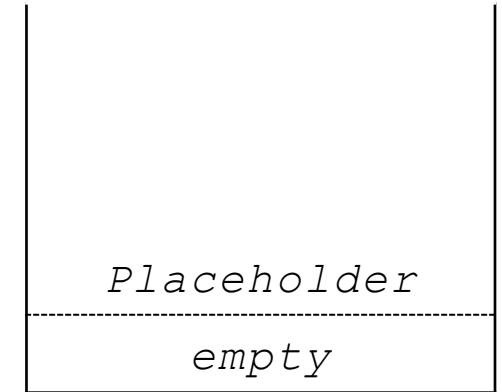
Instruction Queue:

 LOAD_FAST (self)
 LOAD_ATTR (f)
 LOAD_CONST (2)
 BINARY_ADD
 STORE_FAST (t)
 LOAD_FAST (t)
 LOAD_FAST (self)
 LOAD_ATTR (get_g)
 CALL_FUNCTION
 BINARY_ADD
 RETURN_VALUE

Object Stack:


 pull two objects from stack

 push placeholder



Variable Mapping:

Name	Value
self	tmp

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

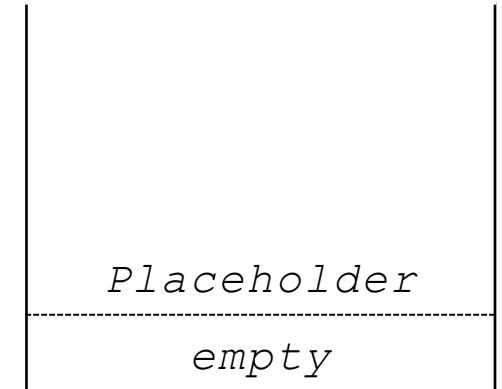
Instruction Queue:

```

LOAD_FAST (self)
LOAD_ATTR (f)
LOAD_CONST (2)
BINARY_ADD
STORE_FAST (t)
LOAD_FAST (t)
LOAD_FAST (self)
LOAD_ATTR (get_g)
CALL_FUNCTION
BINARY_ADD
RETURN_VALUE

```

Object Stack:



pull object
from stack

save in
mapping

Variable Mapping:

Name	Value
self	tmp

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

Instruction Queue:

```

LOAD_FAST (self)
LOAD_ATTR (f)
LOAD_CONST (2)
BINARY_ADD
STORE_FAST (t)
LOAD_FAST (t)
LOAD_FAST (self)
LOAD_ATTR (get_g)
CALL_FUNCTION
BINARY_ADD
RETURN_VALUE

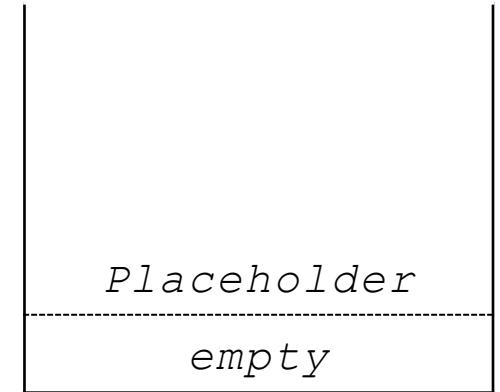
```



pull object
from stack

save in
mapping

Object Stack:



Variable Mapping:

Name	Value
self	tmp
t	Placeholder

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

Instruction Queue:

```

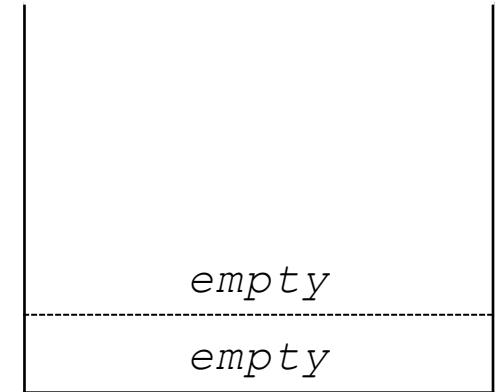
LOAD_FAST (self)
LOAD_ATTR (f)
LOAD_CONST (2)
BINARY_ADD
STORE_FAST (t)
LOAD_FAST (t)
LOAD_FAST (self)
LOAD_ATTR (get_g)
CALL_FUNCTION
BINARY_ADD
RETURN_VALUE

```



load t
from variable
mapping

Object Stack:



Variable Mapping:

Name	Value
self	tmp
t	Placeholder

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

Instruction Queue:

```

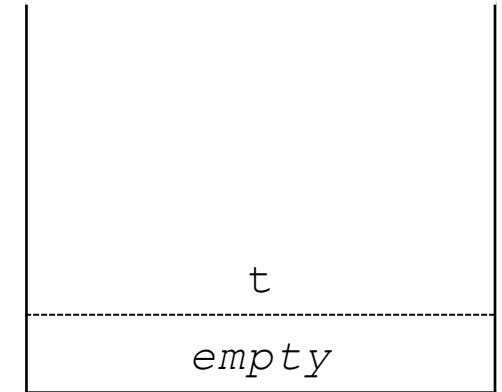
LOAD_FAST (self)
LOAD_ATTR (f)
LOAD_CONST (2)
BINARY_ADD
STORE_FAST (t)
LOAD_FAST (t)
LOAD_FAST (self)
LOAD_ATTR (get_g)
CALL_FUNCTION
BINARY_ADD
RETURN_VALUE

```



Object Stack:

load t
from variable
mapping

Variable Mapping:

Name	Value
self	tmp
t	Placeholder

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

Instruction Queue:

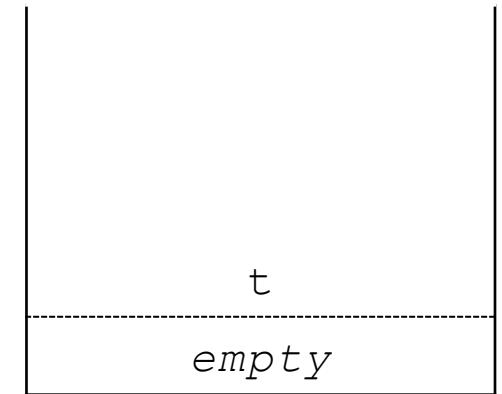
```

LOAD_FAST (self)
LOAD_ATTR (f)
LOAD_CONST (2)
BINARY_ADD
STORE_FAST (t)
LOAD_FAST (t)
LOAD_FAST (self)
LOAD_ATTR (get_g)
CALL_FUNCTION
BINARY_ADD
RETURN_VALUE

```



Object Stack:



load self
from variable
mapping



Variable Mapping:

Name	Value
self	tmp
t	Placeholder

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

Instruction Queue:

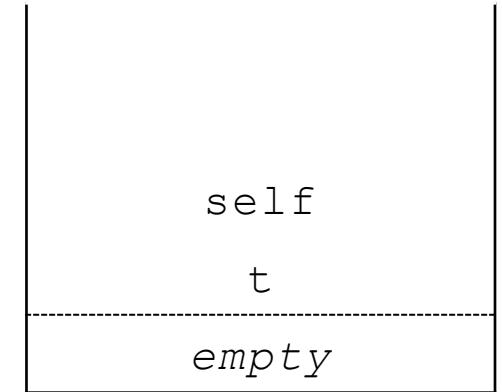
```

LOAD_FAST (self)
LOAD_ATTR (f)
LOAD_CONST (2)
BINARY_ADD
STORE_FAST (t)
LOAD_FAST (t)
LOAD_FAST (self)
LOAD_ATTR (get_g)
CALL_FUNCTION
BINARY_ADD
RETURN_VALUE

```



Object Stack:



load self
from variable
mapping



Variable Mapping:

Name	Value
self	tmp
t	Placeholder

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

Instruction Queue:

```

LOAD_FAST (self)
LOAD_ATTR (f)
LOAD_CONST (2)
BINARY_ADD
STORE_FAST (t)
LOAD_FAST (t)
LOAD_FAST (self)
LOAD_ATTR (get_g)
CALL_FUNCTION
BINARY_ADD
RETURN_VALUE

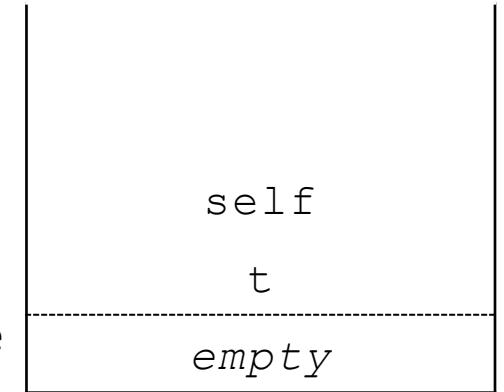
```



Object Stack:

pull object
from stack

push attribute



Variable Mapping:

Name	Value
self	tmp
t	Placeholder

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

Instruction Queue:

```

LOAD_FAST (self)
LOAD_ATTR (f)
LOAD_CONST (2)
BINARY_ADD
STORE_FAST (t)
LOAD_FAST (t)
LOAD_FAST (self)
LOAD_ATTR (get_g)
CALL_FUNCTION
BINARY_ADD
RETURN_VALUE

```



Object Stack:

pull object
 from stack

 push attribute

get_g (self)
t
empty

Variable Mapping:

Name	Value
self	tmp
t	Placeholder

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

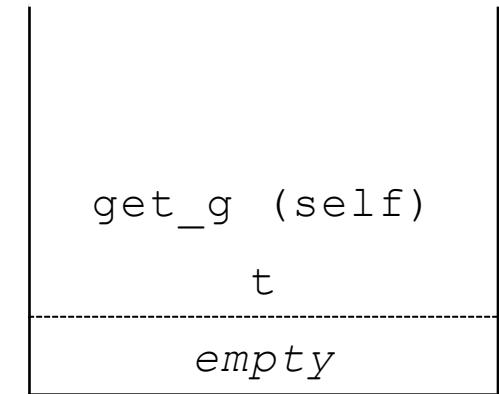
Instruction Queue:

```

LOAD_FAST (self)
LOAD_ATTR (f)
LOAD_CONST (2)
BINARY_ADD
STORE_FAST (t)
LOAD_FAST (t)
LOAD_FAST (self)
LOAD_ATTR (get_g)
CALL_FUNCTION
BINARY_ADD
RETURN_VALUE

```

Object Stack:



pull object from stack

call function
`get_g from self`

Variable Mapping:

Name	Value
<code>self</code>	<code>tmp</code>
<code>t</code>	Placeholder

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

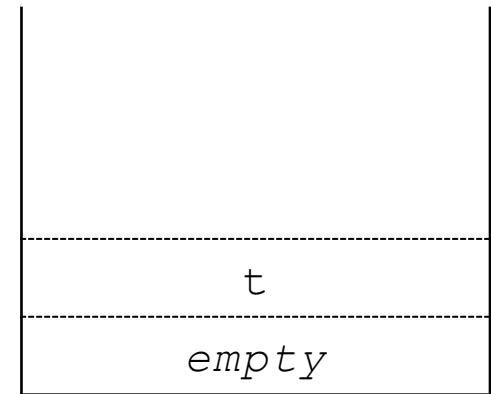
Instruction Queue:

→ LOAD_FAST (self)
 LOAD_ATTR (g)
 RETURN_VALUE

Object Stack:

load self
 from variable
 mapping

→



Variable Mapping:

Name	Value
self	tmp

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

def get_g(self):
    return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

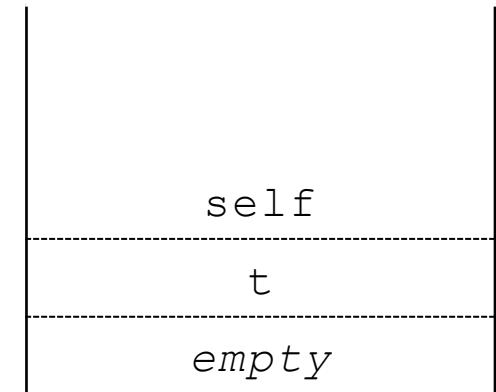
Instruction Queue:

→ LOAD_FAST (self)
 LOAD_ATTR (g)
 RETURN_VALUE

Object Stack:

load self
 from variable
 mapping

→



Variable Mapping:

Name	Value
self	tmp

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

def get_g(self):
    return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

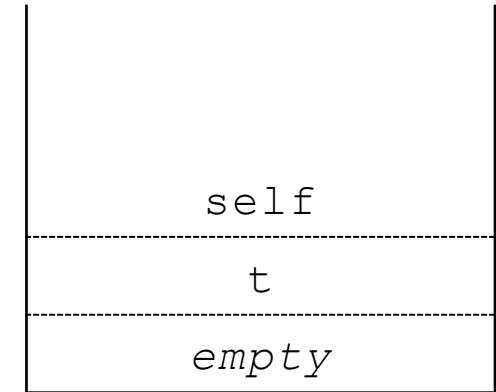
Instruction Queue:

 LOAD_FAST (self)
 LOAD_ATTR (g)
 RETURN_VALUE

Object Stack:

pull object from stack

push attribute



Variable Mapping:

Name	Value
self	tmp

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

def get_g(self):
    return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

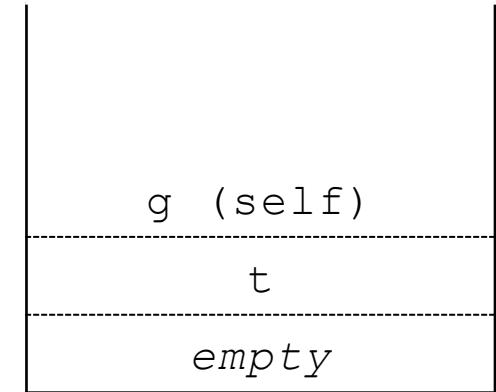
Instruction Queue:

 LOAD_FAST (self)
 LOAD_ATTR (g)
 RETURN_VALUE

Object Stack:

pull object from stack

push attribute



Variable Mapping:

Name	Value
self	tmp

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

def get_g(self):
    return self.g

tmp = Example()
aexpr(lambda: tmp.method())

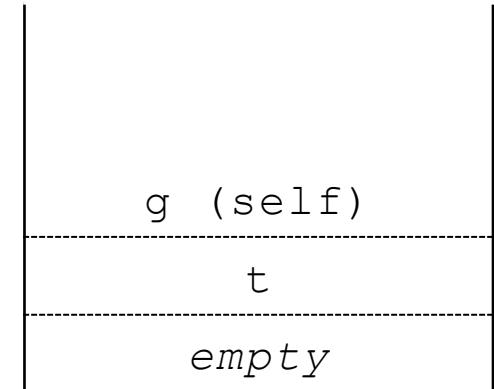
```

Instruction Queue:

LOAD_FAST (self)
 LOAD_ATTR (g)
 RETURN_VALUE

return
 top of stack

Object Stack:



Variable Mapping:

Name	Value
self	tmp

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

Instruction Queue:

```

LOAD_FAST (self)
LOAD_ATTR (f)
LOAD_CONST (2)
BINARY_ADD
STORE_FAST (t)
LOAD_FAST (t)
LOAD_FAST (self)
LOAD_ATTR (get_g)
CALL_FUNCTION
BINARY_ADD
RETURN_VALUE

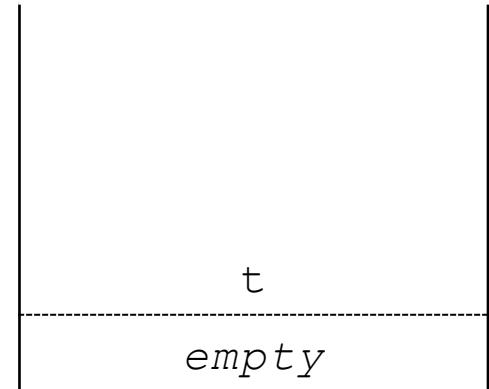
```



pull object
 from stack
 ←
 call function
 get_g from
 self

push return
 value to
 stack

Object Stack:



Variable Mapping:

Name	Value
self	tmp
t	Placeholder

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

Instruction Queue:

```

LOAD_FAST (self)
LOAD_ATTR (f)
LOAD_CONST (2)
BINARY_ADD
STORE_FAST (t)
LOAD_FAST (t)
LOAD_FAST (self)
LOAD_ATTR (get_g)
CALL_FUNCTION
BINARY_ADD
RETURN_VALUE

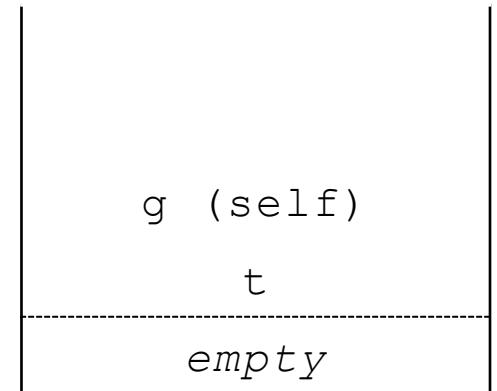
```



pull object
 from stack
 ←
 call function
 get_g from
 self

push return
 value to
 stack

Object Stack:



Variable Mapping:

Name	Value
self	tmp
t	Placeholder

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

Instruction Queue:

```

LOAD_FAST (self)
LOAD_ATTR (f)
LOAD_CONST (2)
BINARY_ADD
STORE_FAST (t)
LOAD_FAST (t)
LOAD_FAST (self)
LOAD_ATTR (get_g)
CALL_FUNCTION
BINARY_ADD
RETURN_VALUE

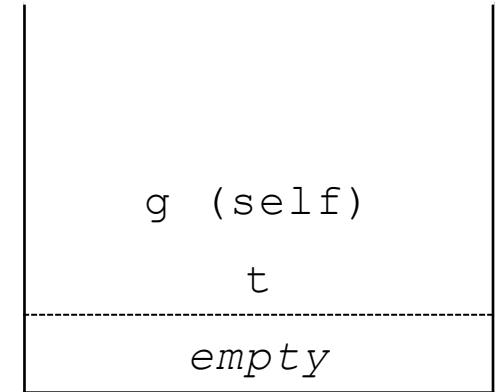
```



pull two
 objects
 from stack

 push
 placeholder
 to stack

Object Stack:



Variable Mapping:

Name	Value
self	tmp
t	Placeholder

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

Instruction Queue:

```

LOAD_FAST (self)
LOAD_ATTR (f)
LOAD_CONST (2)
BINARY_ADD
STORE_FAST (t)
LOAD_FAST (t)
LOAD_FAST (self)
LOAD_ATTR (get_g)
CALL_FUNCTION
BINARY_ADD
RETURN_VALUE

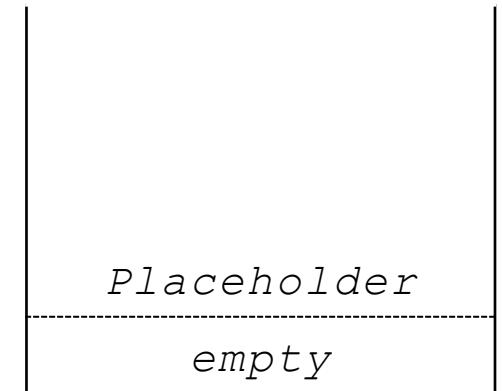
```



Object Stack:

pull two
 objects
 from stack

 push
 placeholder
 to stack



Variable Mapping:

Name	Value
self	tmp
t	Placeholder

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

Instruction Queue:

```

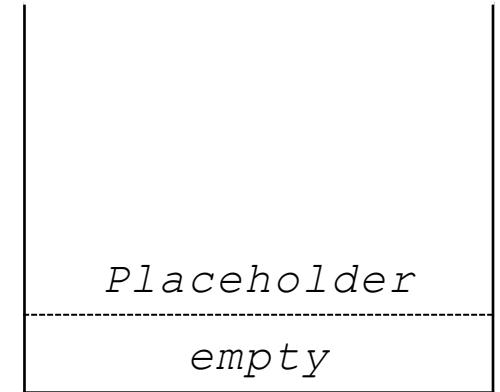
LOAD_FAST (self)
LOAD_ATTR (f)
LOAD_CONST (2)
BINARY_ADD
STORE_FAST (t)
LOAD_FAST (t)
LOAD_FAST (self)
LOAD_ATTR (get_g)
CALL_FUNCTION
BINARY_ADD
RETURN_VALUE

```



return top of
stack

Object Stack:



Variable Mapping:

Name	Value
self	tmp
t	Placeholder

Example (What)

```
class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

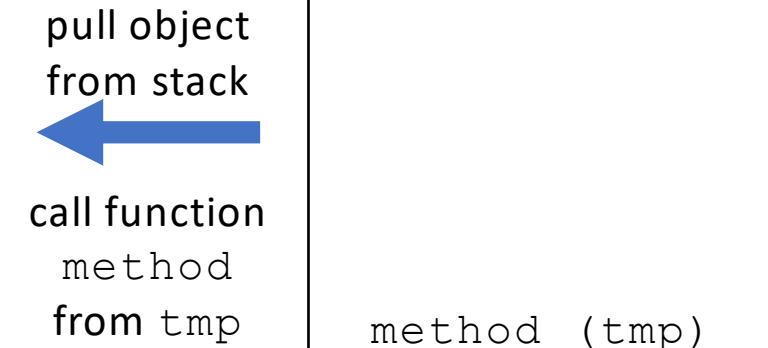
tmp = Example()
aexpr(lambda: tmp.method())
```

Instruction Queue:

LOAD_GLOBAL (tmp)
 LOAD_ATTR (method)
 CALL_FUNCTION
 RETURN VALUE



Object Stack:



Variable Mapping:

Name	Value
self	None

Example (What)

```

class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())

```

Instruction Queue:

 LOAD_GLOBAL (tmp)
 LOAD_ATTR (method)
 CALL_FUNCTION
 RETURN VALUE

Object Stack:

pull object
 from stack

 call function
 method
 from tmp
 push return
 value

Placeholder
 method (tmp)

Variable Mapping:

Name	Value
self	None

Example (What)

```
class Example:
    def __init__(self):
        self.f = 5
        self.g = 10

    def method(self):
        t = self.f + 2
        return t + self.get_g()

    def get_g(self):
        return self.g

tmp = Example()
aexpr(lambda: tmp.method())
```

Instruction Queue:

LOAD_GLOBAL (tmp)
 LOAD_ATTR (method)
 CALL_FUNCTION
 RETURN VALUE

Object Stack:

return
 top of stack
Placeholder
 method (tmp)

Variable Mapping:

Name	Value
self	None

Placing the hook (When)

- Imagine: Placing a hook on the attribute `attr` of object `obj`

`obj (Instance-side)`

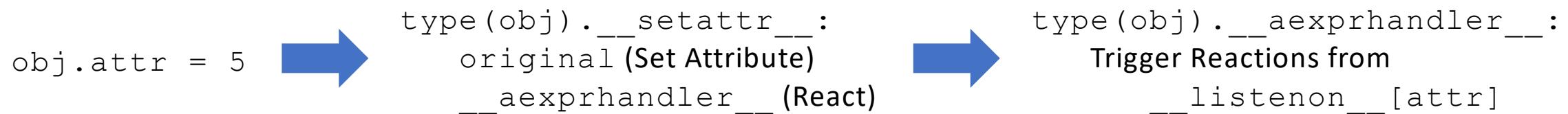
- `obj` gets a field `__listenon__`:
- ```
__listenon__ =
{
 attr:
 set(<ExpressionReactions>)
}
```

## `type(obj) (Class-side)`

- Class gets a method `__aexprhandler__`:  
calls all `ExpressionReactions` in  
`__listenon__[attr]`
- Class gets a new `__setattr__` method:  
calls original one first, afterwards  
`__aexprhandler__`

# Placing the hook (When)

- Imagine: Placing a hook on the attribute `attr` of object `obj`
- Control flow:



# Limitations

# Limitations

- There are at least these limitations:

## Lists:

```
def listExample(self):
 a = []
 a.append(self)
 return a[0].f
```

- `a = []` could add a placeholder
- Access to element would be access to placeholder – not supported so far
- Type Inference could solve the problem
- Same for sets, maps, ...

## Local Variables:

```
def localVarExample(self):
 a = 5
 aexpr(lambda: a).onChange()
 a = 6
```

- Local variables are not instrumentable
- No `__setattr__` or `__set__` method

# Limitations

- There are at least these limitations:

## External Resources:

```
aexpr(lambda: server.reachable())
```

- There is no repeated check if a external resource is available
- Could be implemented with a field storing a Boolean flag and the user updates that flag
- Same for local files

## Transactions:

```
aexpr(lambda: rect.aspect_ratio())
```

```
def update_bottomleft(point):
 rect.point1.x = point.x
 rect.point1.y = point.y
```

- Triggers twice, first one is obsolete

# Limitations

- There are at least these limitations:

Other language features:

- Exceptions
- Closures
- Meta-Programming
- Concurrency
- Asynchronous

# Discussion: Implementation Ideas

# Discussion: Implementation Ideas

- We discussed the following approaches:

## exec

- Use build-in `exec()` to execute expression
  - Use self-implemented maps for `globals()` and `locals()` to monitor, which objects / variables are necessary
- + Able to detect local variables  
- Loose tool support (like error tracing)  
- Dynamic Analysis – visiting only one branch

## trace

- Insert own trace hook with `sys.settrace()`
- + Able to detect local variables  
+ Really easy to implement  
- Really slow!

# Discussion: Implementation Ideas

- We discussed the following approaches:

## “Pybelsberg”-Way

- Wrap all objects with an object-wrapper
  - Override getters and setters
- + Easy implementation
- Not able to instrument local variables
- Dynamic Analysis – visiting only one branch
- Side-Effects possible

## Static Code Analysis

- Byte-Code Analysis as presented before
- + Static-Analysis – identify all dependencies
- + Skip Side-Effects
- + Can be really detailed
- Start-Overhead
- Not able to instrument local variables
- Lots of different (complex) operations

# Next Steps & Future Work

# Next Steps / Future Work

- Think about how to handle lists / sets / maps / ...
  - Type inference could be a solution for this
- Think about Transactions

# Discussion

About: Demo? Implementation? Other Variants? Next Steps / Future Work?