

Stack graphs

Name resolution at scale

Douglas A. Creager
Hendrik van Antwerpen



Eelco Visser Commemorative Symposium
April 5, 2023 – TU Delft

Code Navigation



Code Navigation

```
stove.py
def bake():
    pass

def broil():
    pass
    ^

def saute():
    pass

broil()
```

The diagram illustrates code navigation within a Python file named 'stove.py'. It shows four function definitions: 'bake()', 'broil()', 'saute()', and 'bake()'. A red arrow originates from the word 'broil' in the fourth line and points back to the start of the 'broil()' function definition. The file name 'stove.py' is positioned above the first function definition.



Index time



Query time



c1fa878547d8a3fb38689ed0fdad438a0ddb4329 pytest.ini
9a899df67fb619d871db8b62eea9c1ed436c3a79 requests/__init__.py
9844f740abea47edbe57cb01d181b341d021ffb1 requests/__version__.py
759d9a56ba0102bb3b7b3abfcfae6731a2ecc243 requests/_internal_utils.py
fa4d9b3cc9a2cf6dc2465958387f32193b431410 requests/adapters.py
88cfcdcebbf51d7bfb7b35579085bf985c6926818 requests/api.py
34e7c8b8fc2daa3229423bc9c6ad6e39c3b3d41a requests/auth.py
d1a378d787d5a220ef2df694c5cbcd6b4015cbea requests/certs.py
c44b35efb984650802c40efbc5281b875bae4adc requests/compat.py
56fccd9c2570d2a31365ed11278fd1b6ecc2aa54 requests/cookies.py
a80cad80f14411ddf0ba86070d1a65c78dbf7878 requests/exceptions.py
e53d35ef6dec122aa4aba941b9c555997a35d143 requests/help.py
7a51f212c8a08fc6a04e509e89d8dcc85185df77 requests/hooks.py
62dcd0b7c8c64399ad0d317fcfa926e40388d968 requests/models.py
7232fe0ff7e85083d47d624532f29badd7c854d0 requests/packages.py
d73d700fa6baa975ab9b10fdbdb186bccdb9d3610 requests/sessions.py
813e8c4e62fdccbfc73e2f52e187e0c3c5b84d7 requests/status_codes.py
da930e2852049feb2b6e262d442b9cc25ecb2d94 requests/structures.py
9170a9d2a45d9dbb44abc4d5ea4d7e548463d requests/utils.py

3e13cc7883138d7c9311dd23c13831ed71331a3f13x0, requests_1.log07.svg
c1fa878547d8a3fb38689ed0fdad438a0ddb4329 **pytest.ini**
9a899df67fb619d871db8b62eea9c1ed436c3a79 requests/_init_.py
9844f740abea47edbe57cb01d181b341d021ffb1 requests/_version_.py
759d9a56ba0102bb3b7b3abfcfae6731a2ecc243 requests/_internal_utils.py
fa4d9b3cc9a2cf6dc2465958387f32193b431410 requests/adapters.py
88cfcdcebbf51d7bfb7b35579085bf985c6926818 requests/api.py
eeface39ae62c3975ff535e6b1f79f2c28fbf888 requests/auth.py
d1a378d787d5a220ef2df694c5cbcd6b4015cbea requests/certs.py
c44b35efb984650802c40efbc5281b875bae4adc requests/compat.py
56fccd9c2570d2a31365ed11278fd1b6ecc2aa54 requests/cookies.py
a80cad80f14411ddf0ba86070d1a65c78dbf7878 requests/exceptions.py
e53d35ef6dec122aa4aba941b9c555997a35d143 requests/help.py
7a51f212c8a08fc6a04e509e89d8dcc85185df77 requests/hooks.py
457b8412d7cabde41125AAF5ed81fbc86e53eacf requests/models.py
7232fe0ff7e85083d47d624532f29badd7c854d0 requests/packages.py
759ecbe03cd52eb7b3ab966c91ad7921a52620eb requests/sessions.py
813e8c4e62fdccbfc73e2f52e187e0c3c5b84d7 requests/status_codes.py
da930e2852049feb2b6e262d442b9cc25ecb2d94 requests/structures.py
28e936e2f79bc627e5fcc95bddc3697e11befc7 requests/utils.py

Incremental processing





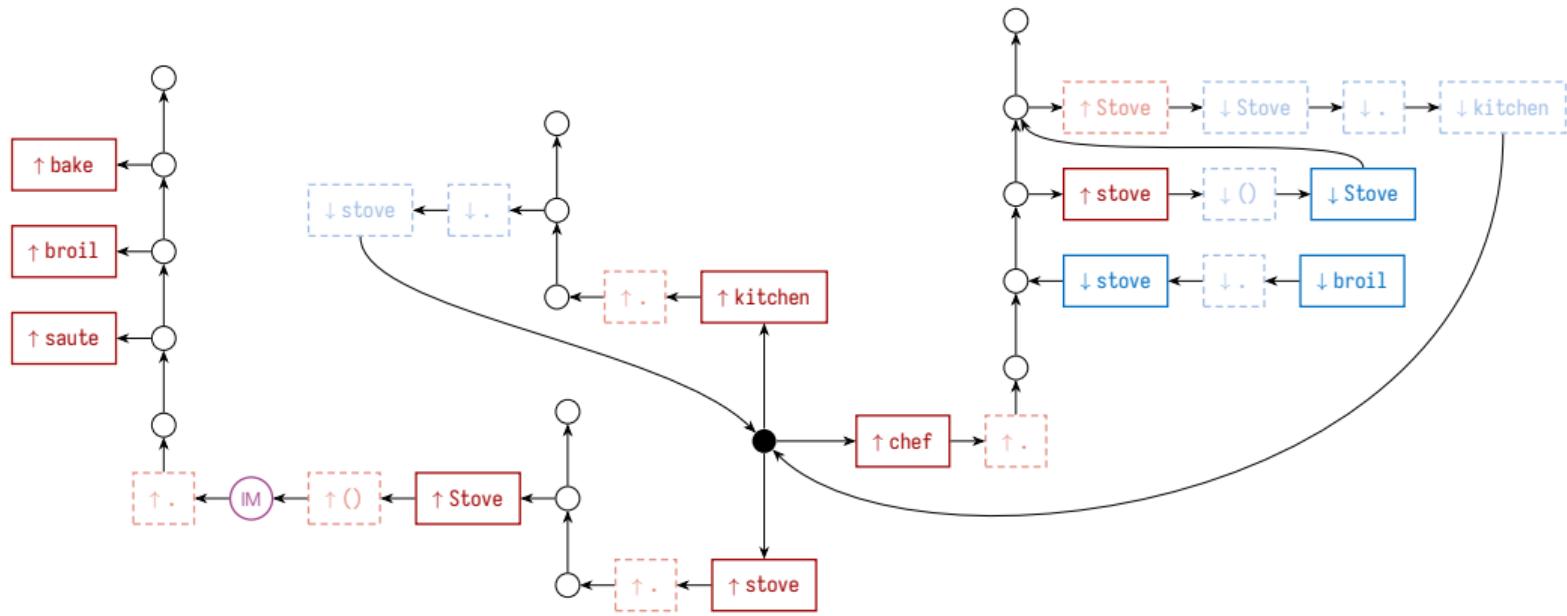
```
stove.py  
class Stove(object):  
    def bake(self):  
        pass  
  
    def broil(self):  
        pass  
  
    def saute(self):  
        pass
```

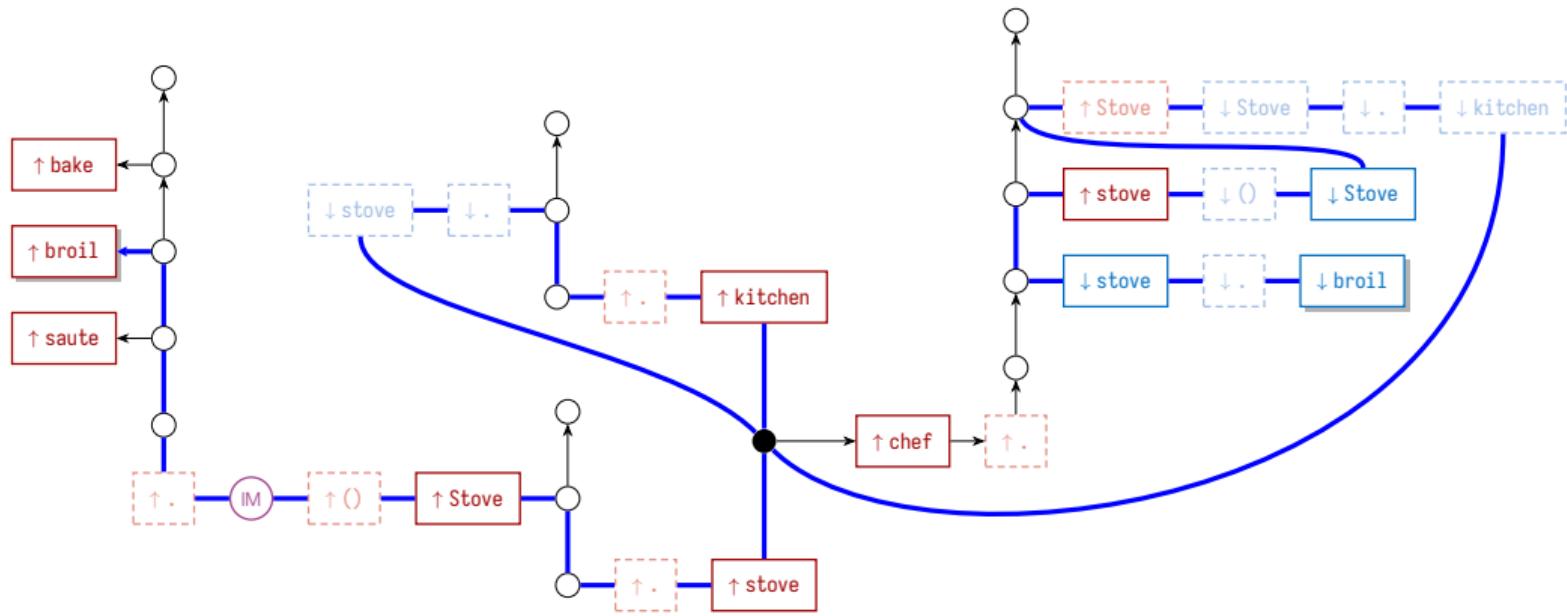


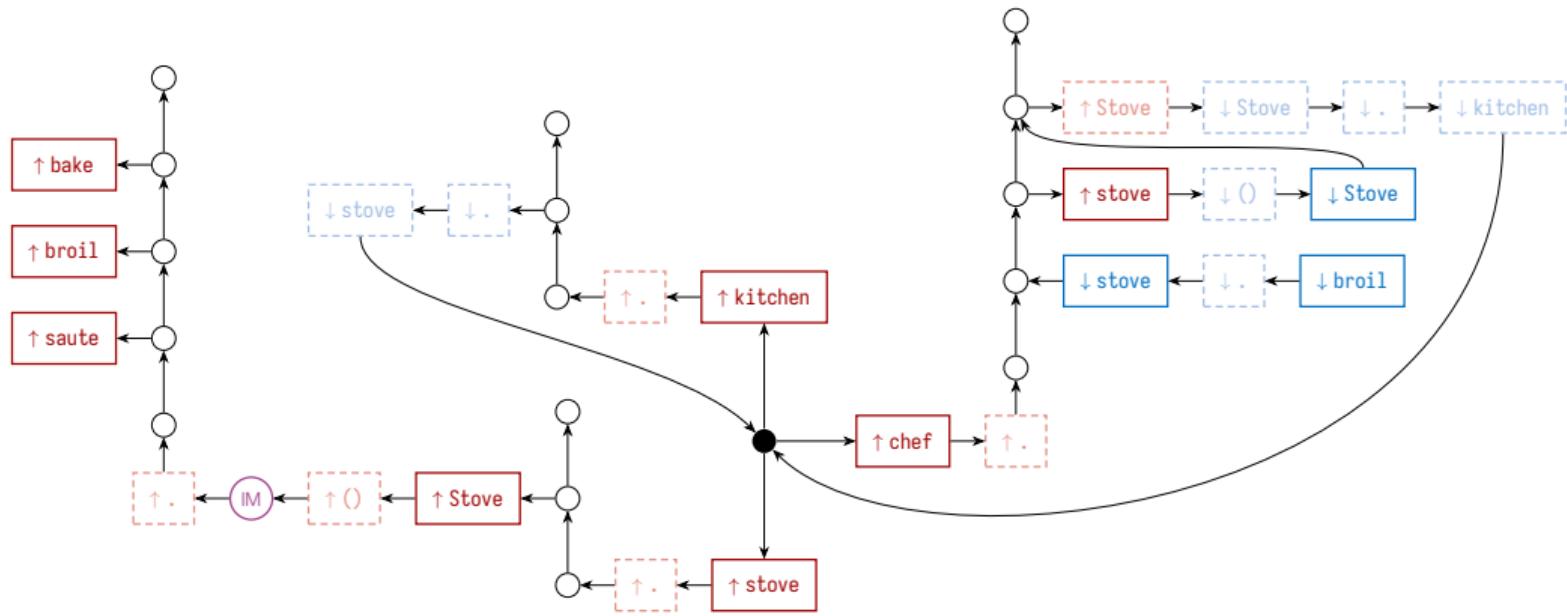
```
kitchen.py  
from stove import *
```



```
chef.py  
from kitchen import Stove  
  
stove = Stove()  
stove.broil()
```

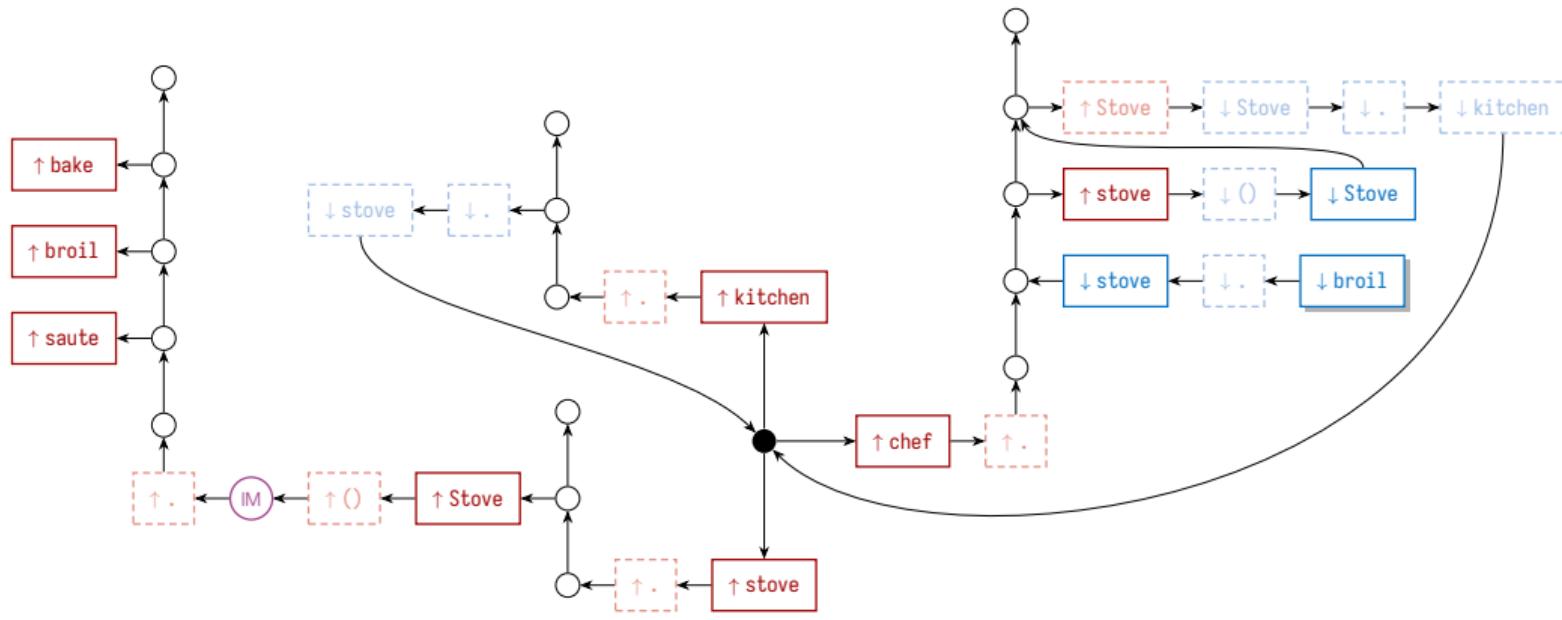




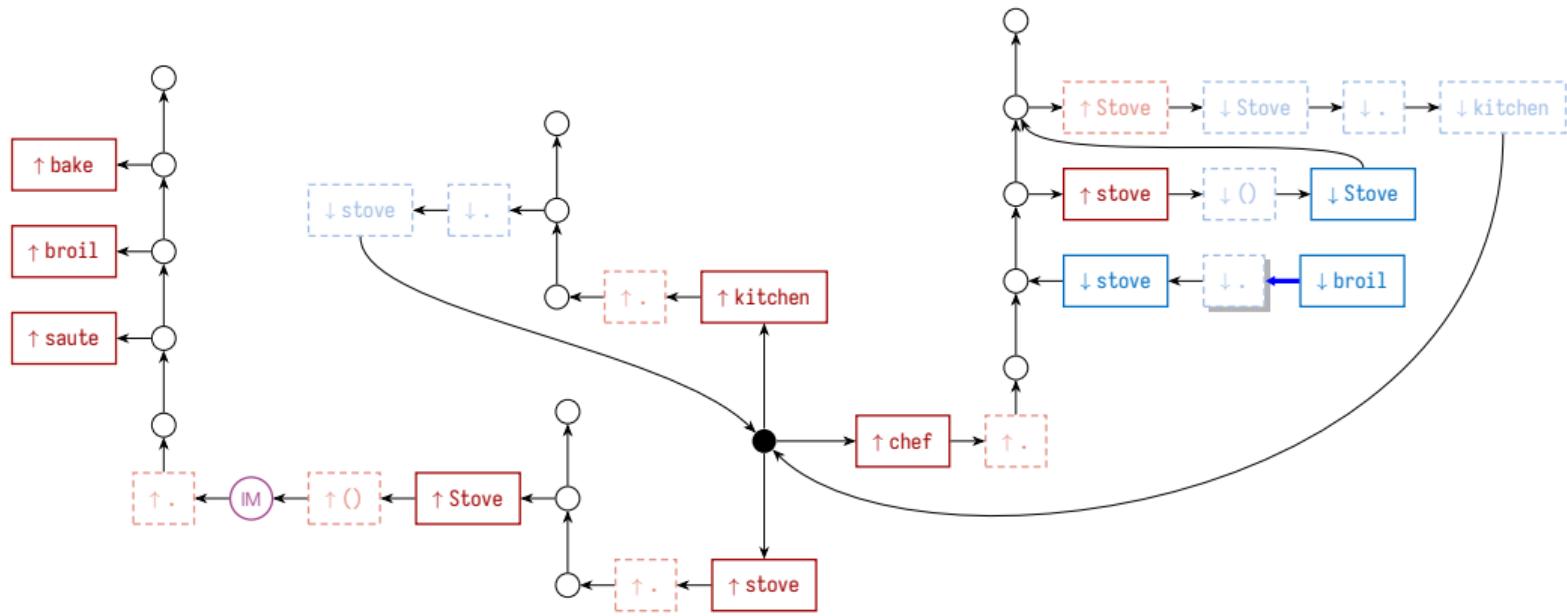


Symbol stack:

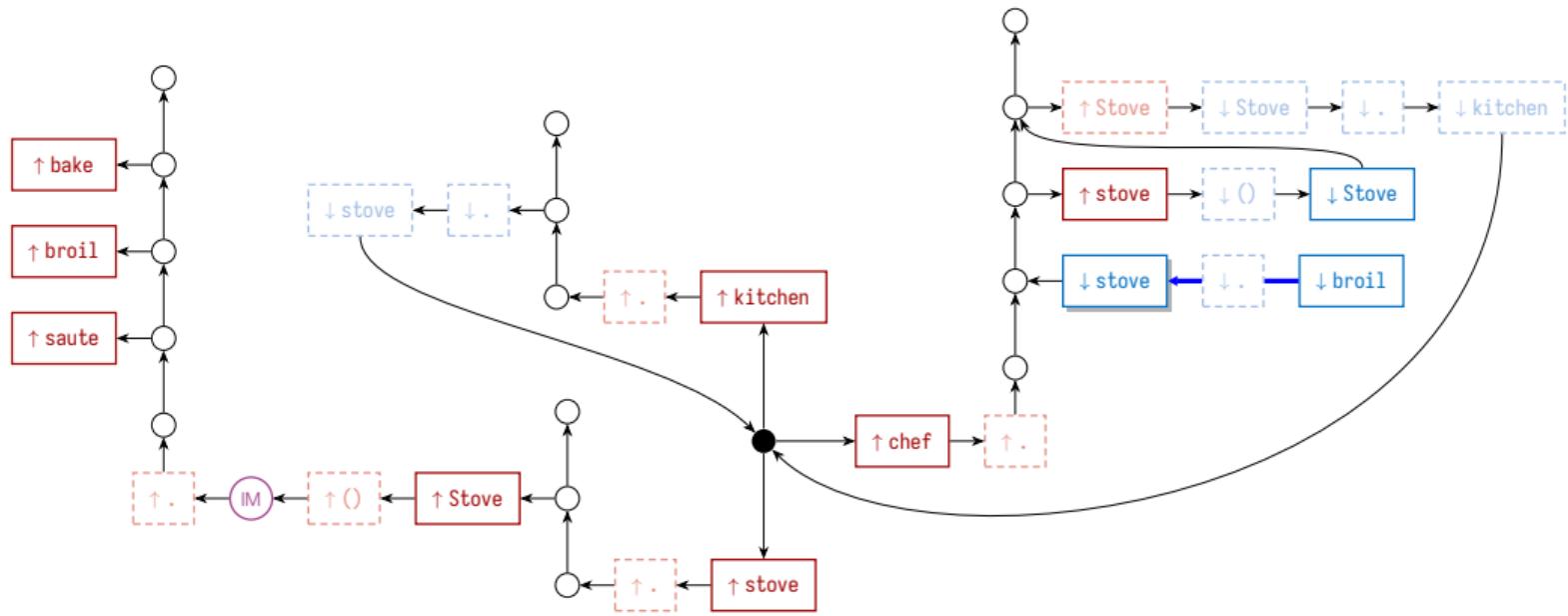




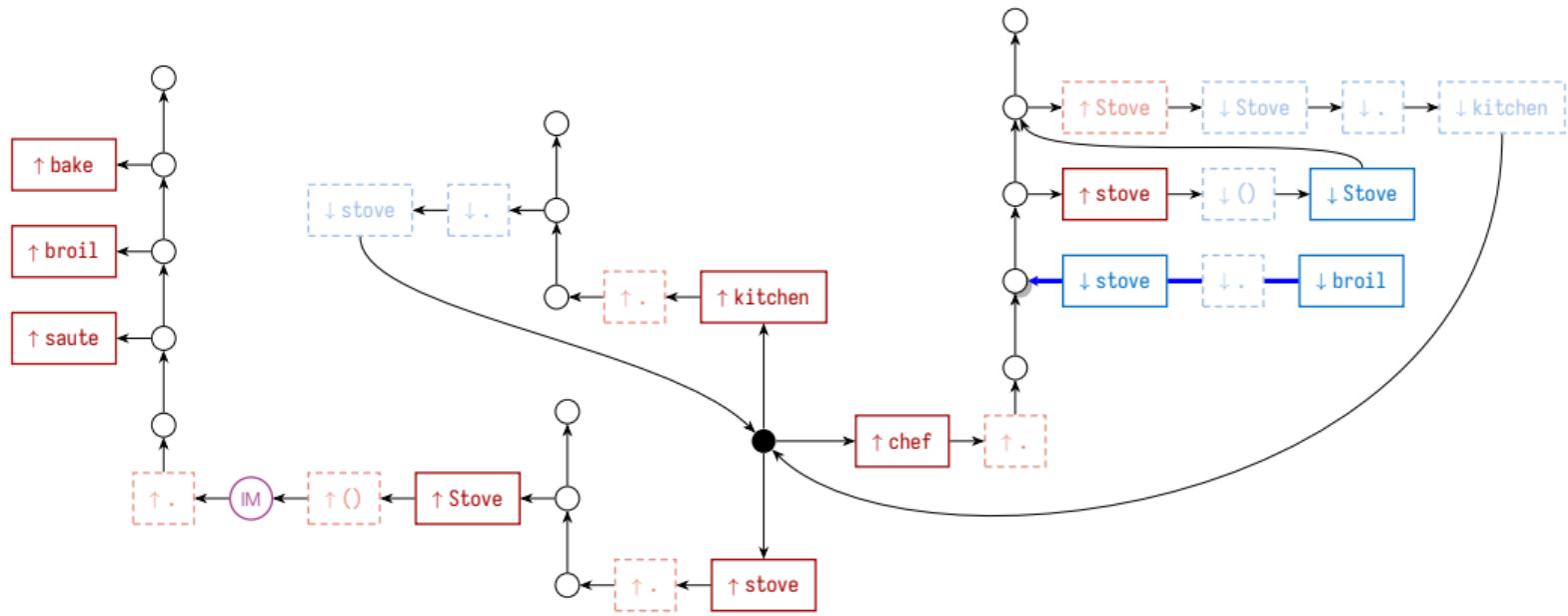
Symbol stack: $\langle \text{broil} \rangle$



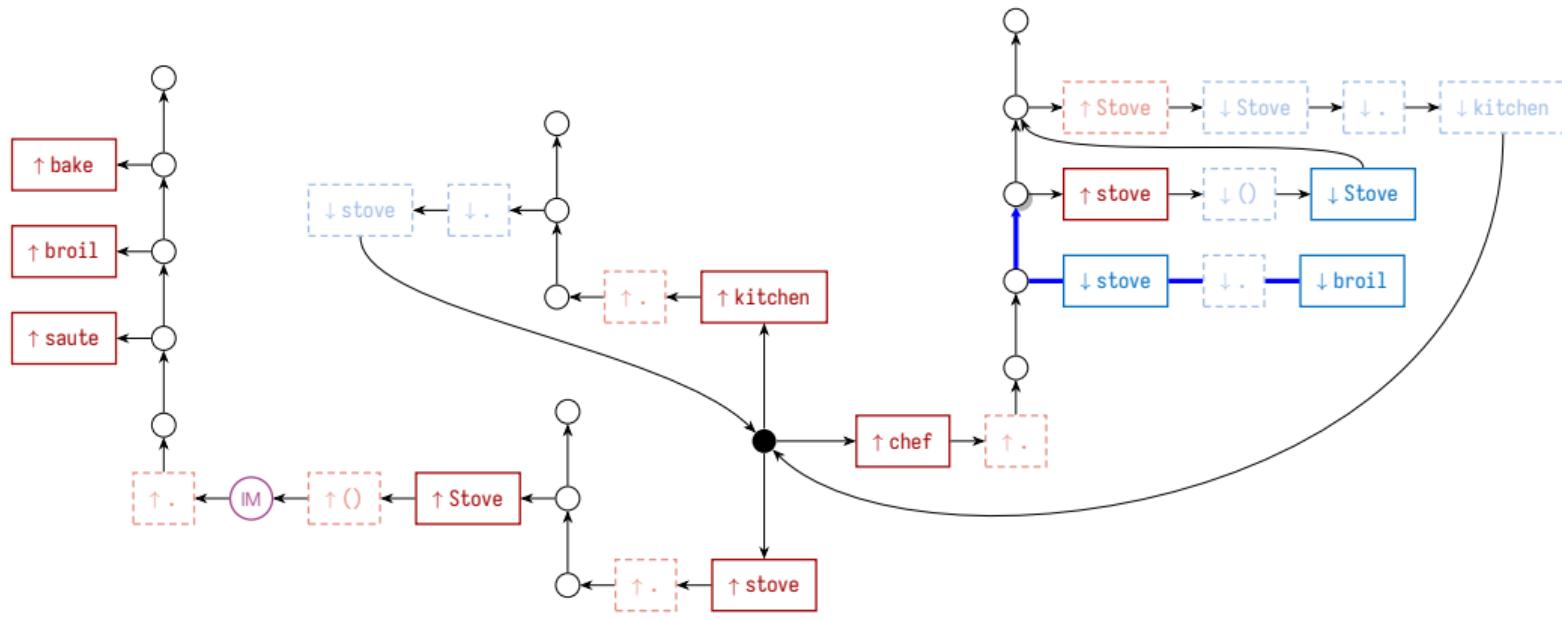
Symbol stack: $\langle \cdot \text{broil} \rangle$



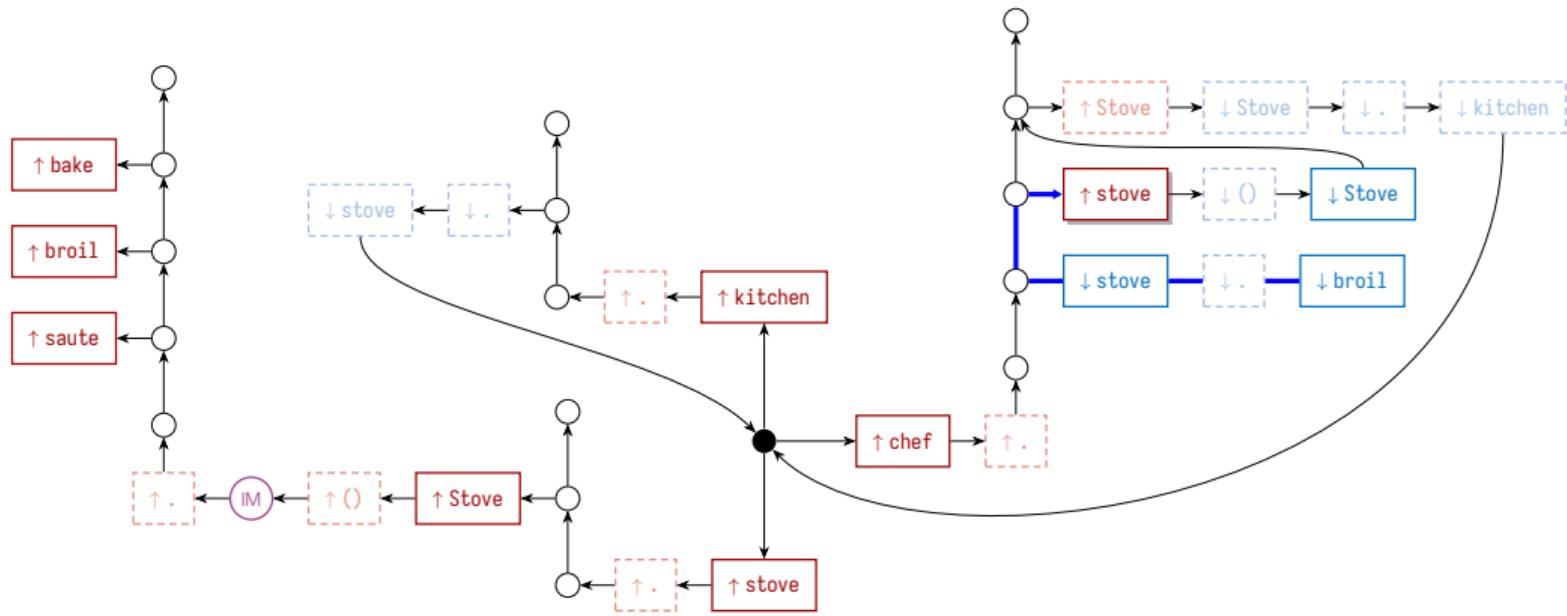
Symbol stack: $\langle \text{stove.broil} \rangle$



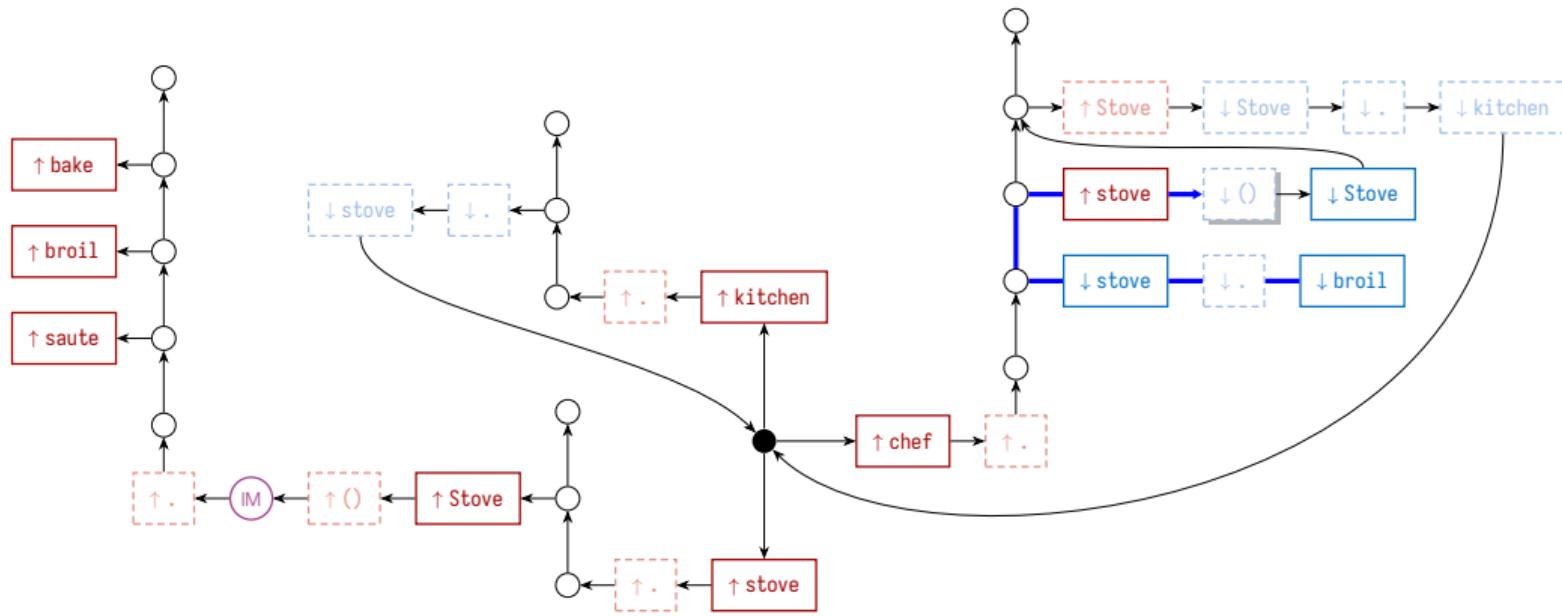
Symbol stack: $\langle \text{stove.broil} \rangle$



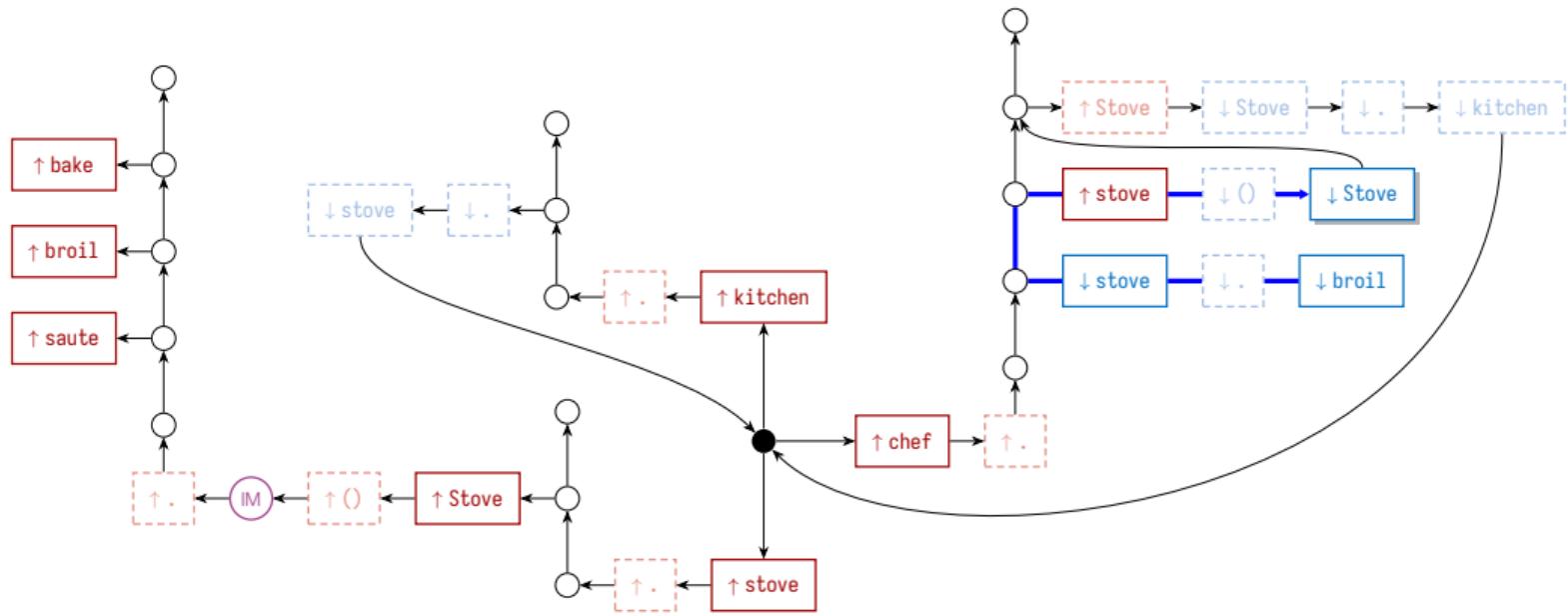
Symbol stack: <stove.broil>



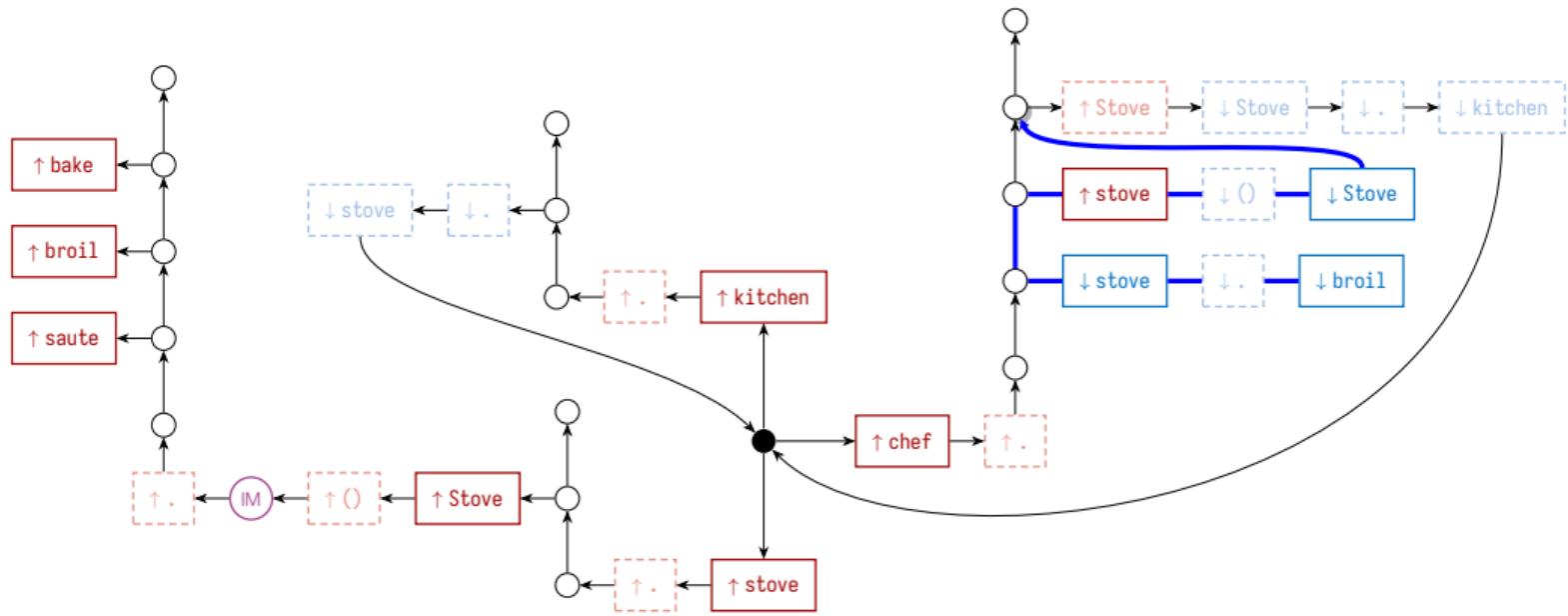
Symbol stack: $\langle \cdot \text{broil} \rangle$



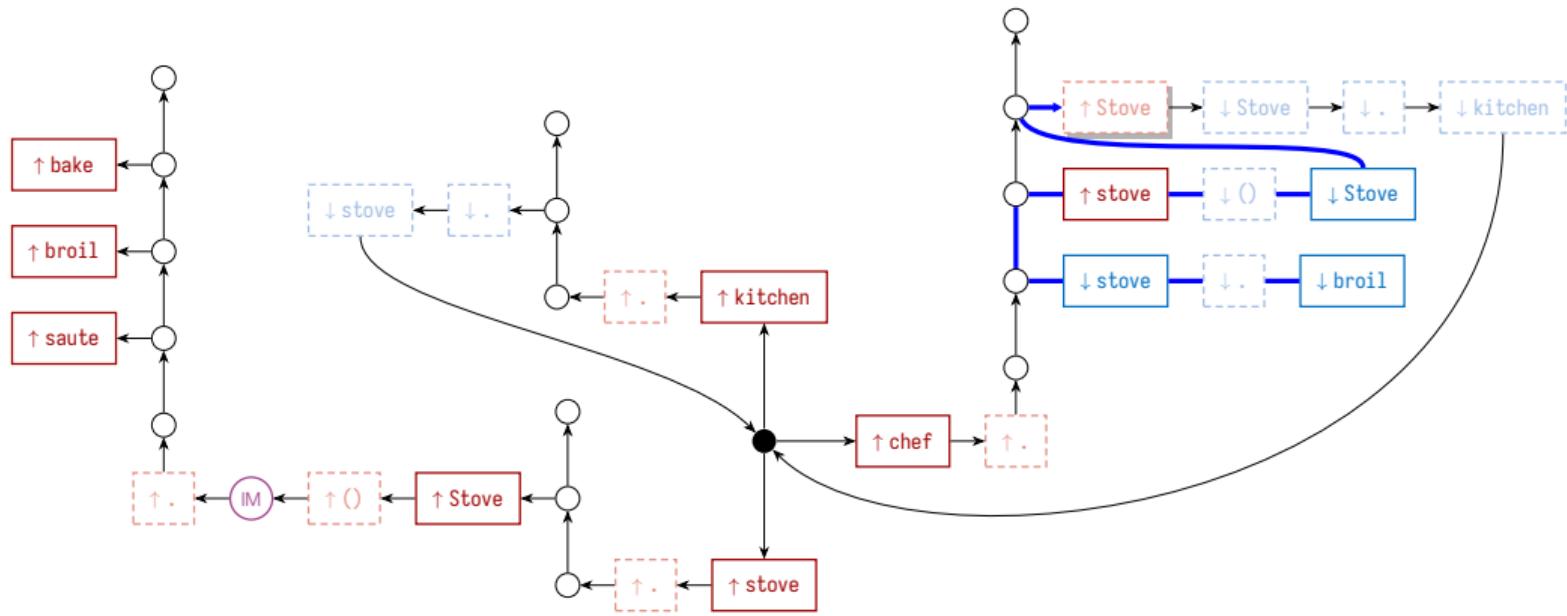
Symbol stack: $\langle () . \text{broil} \rangle$



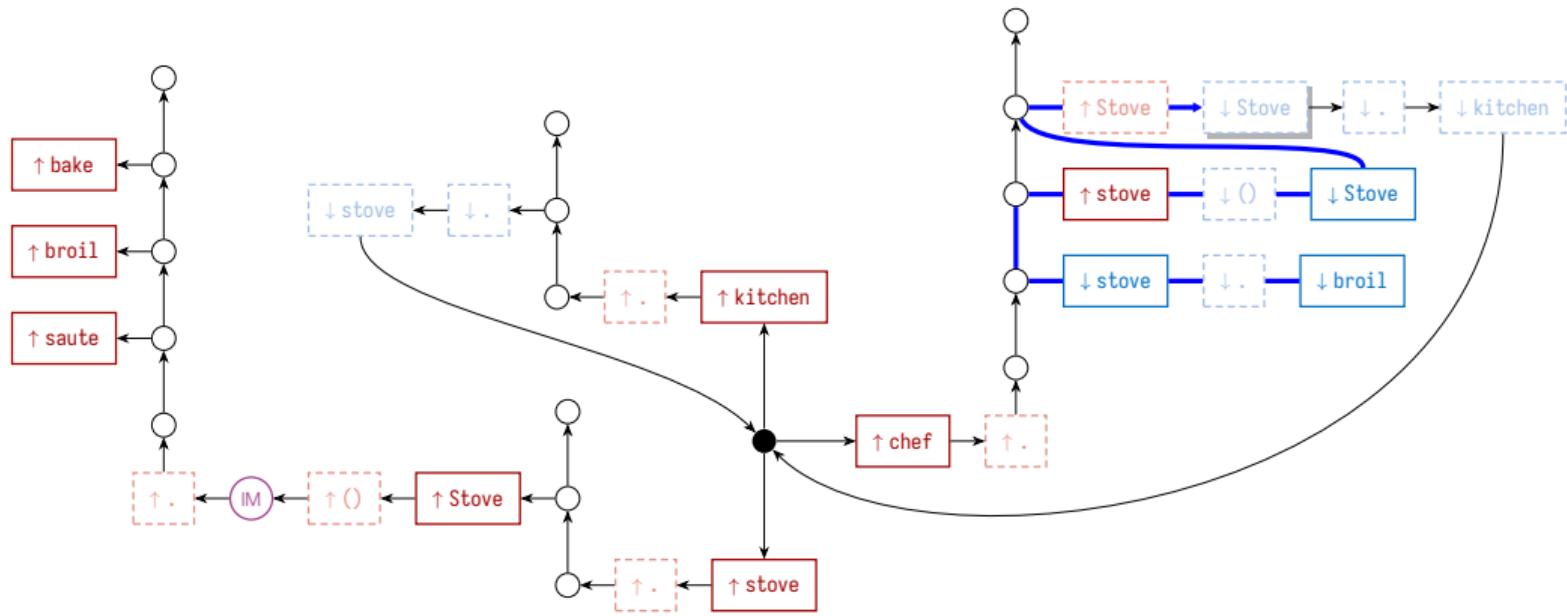
Symbol stack: $\langle \text{Stove}().\text{broil} \rangle$



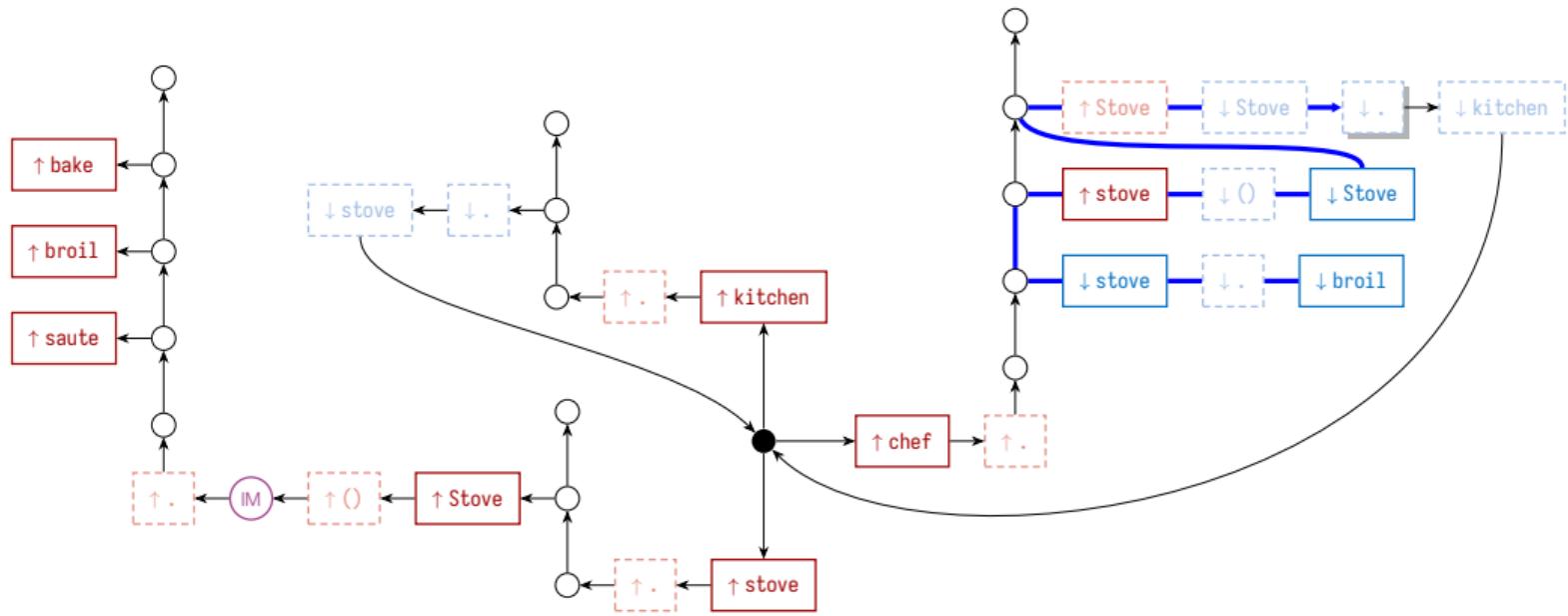
Symbol stack: $\langle \text{Stove}().\text{broil} \rangle$



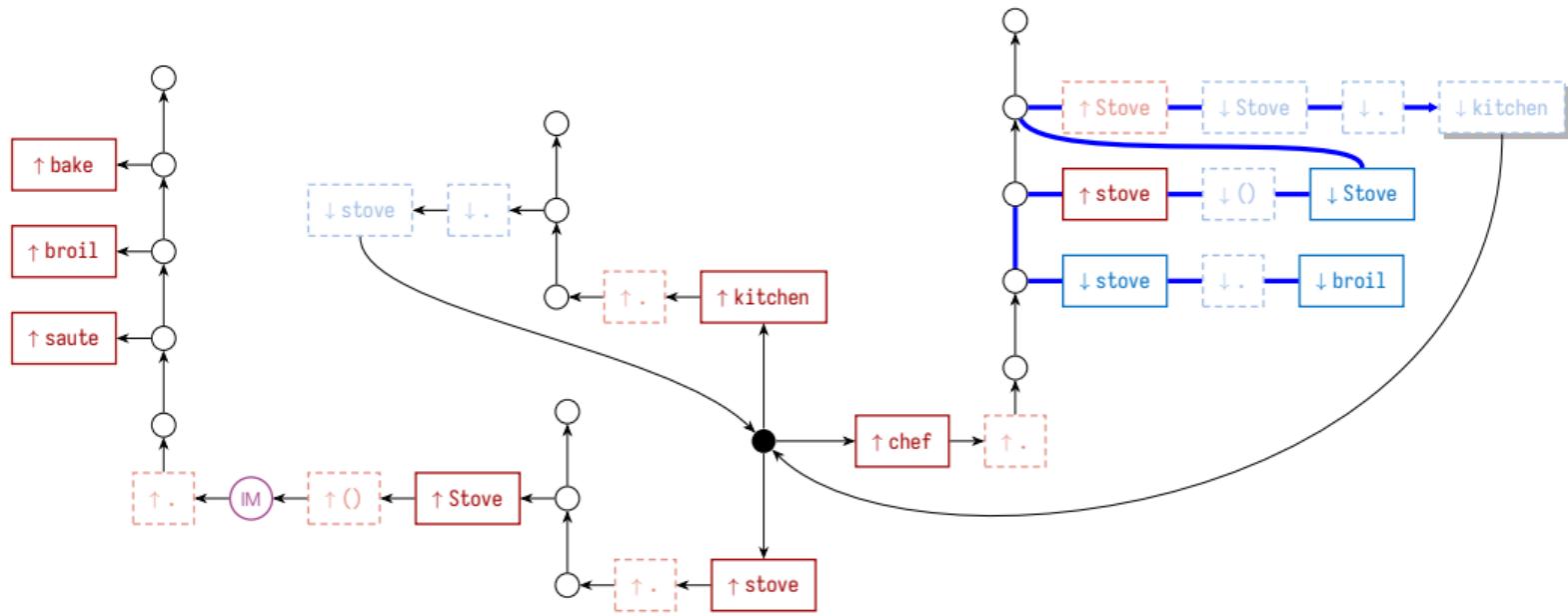
Symbol stack: $\langle () . \text{broil} \rangle$



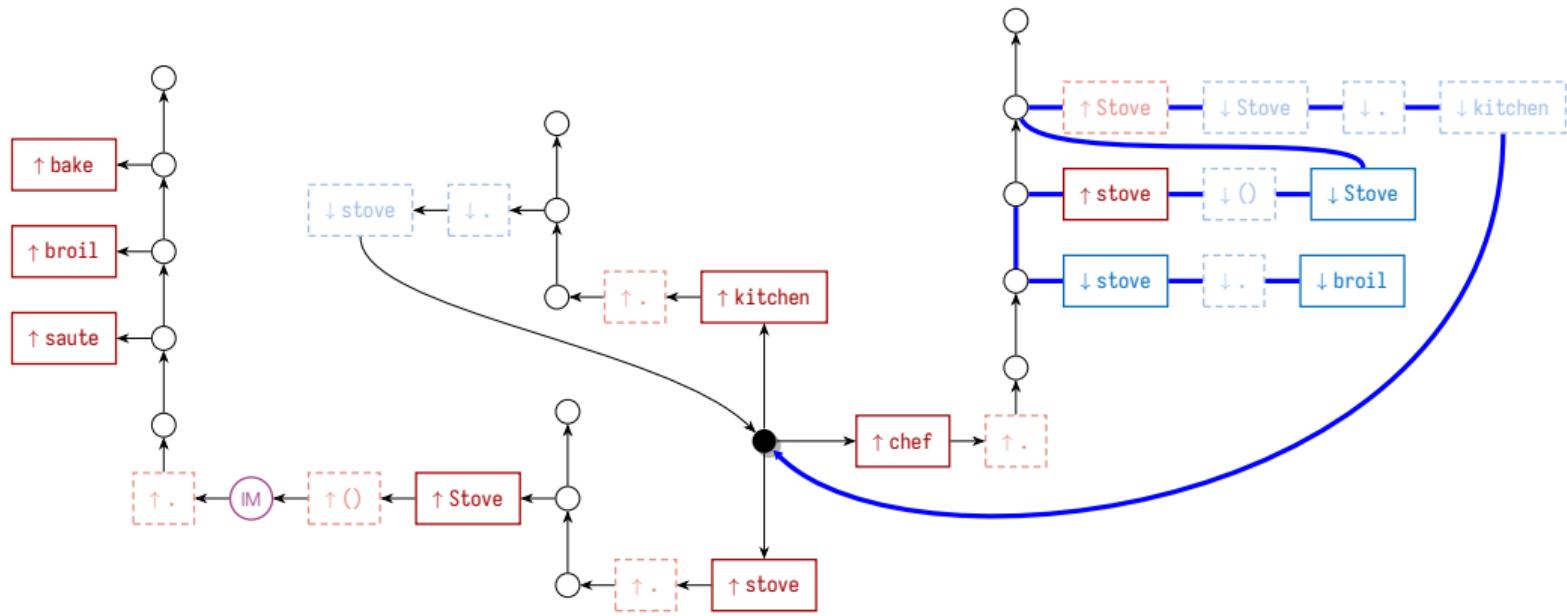
Symbol stack: $\langle \text{Stove}().\text{broil} \rangle$



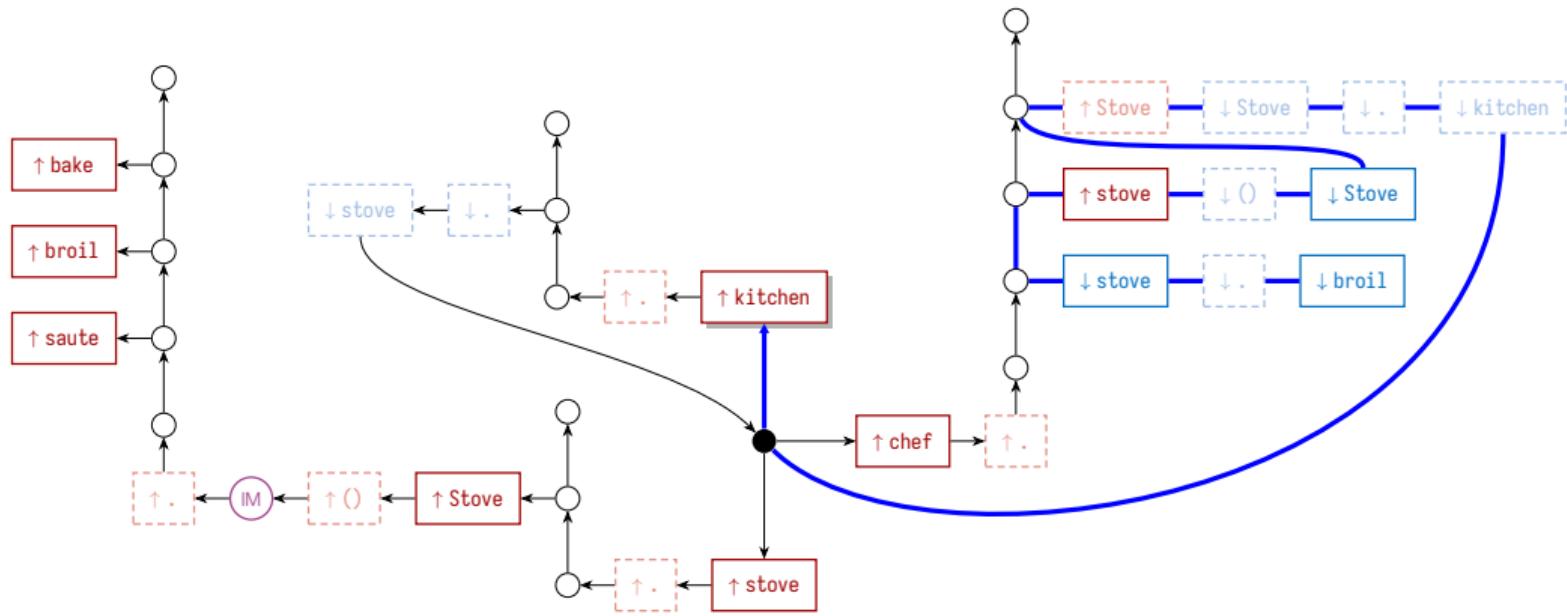
Symbol stack: $\langle .\text{Stove}().\text{broil} \rangle$



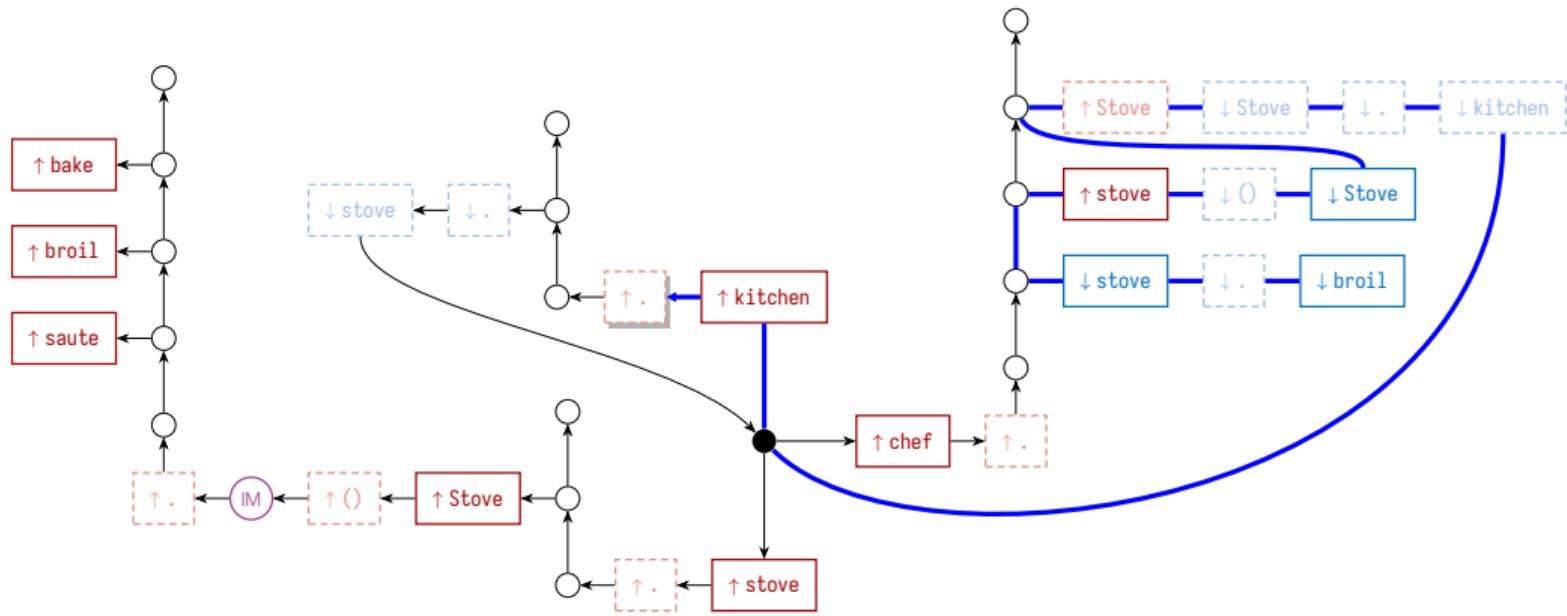
Symbol stack: $\langle \text{kitchen.Stove()}.broil \rangle$



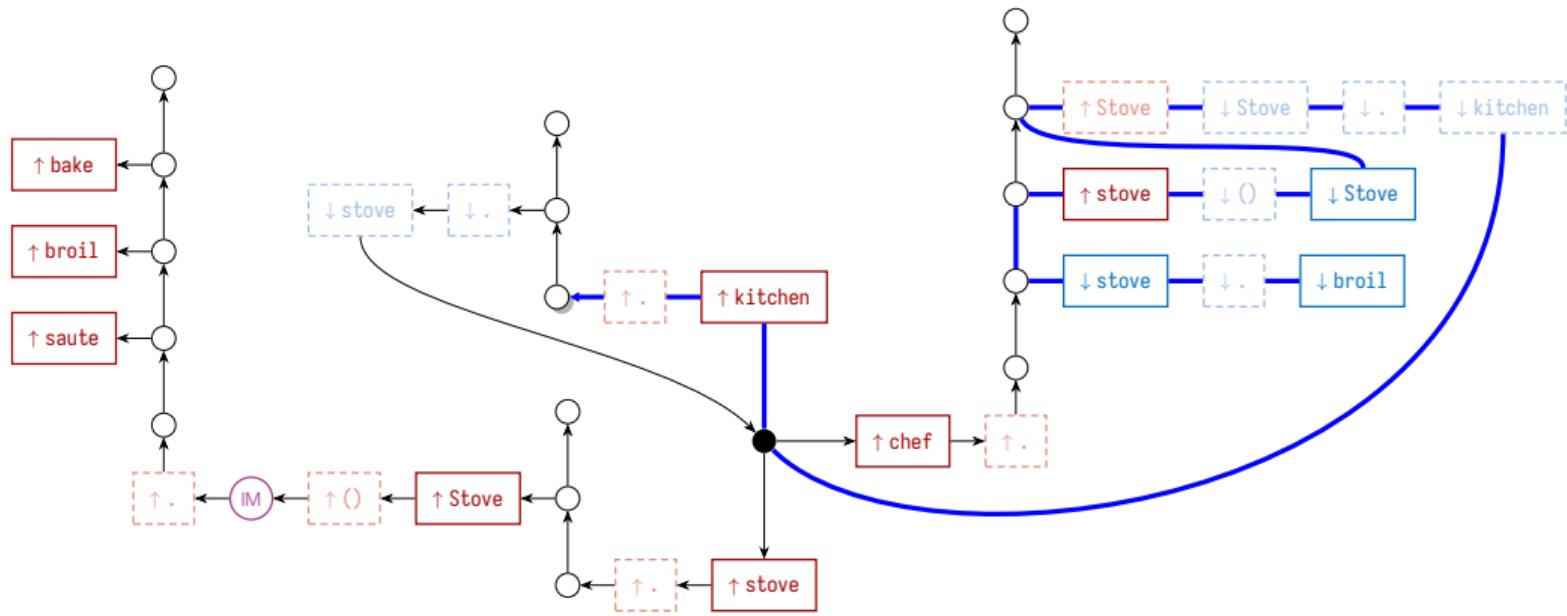
Symbol stack: **<kitchen.Stove().broil>**



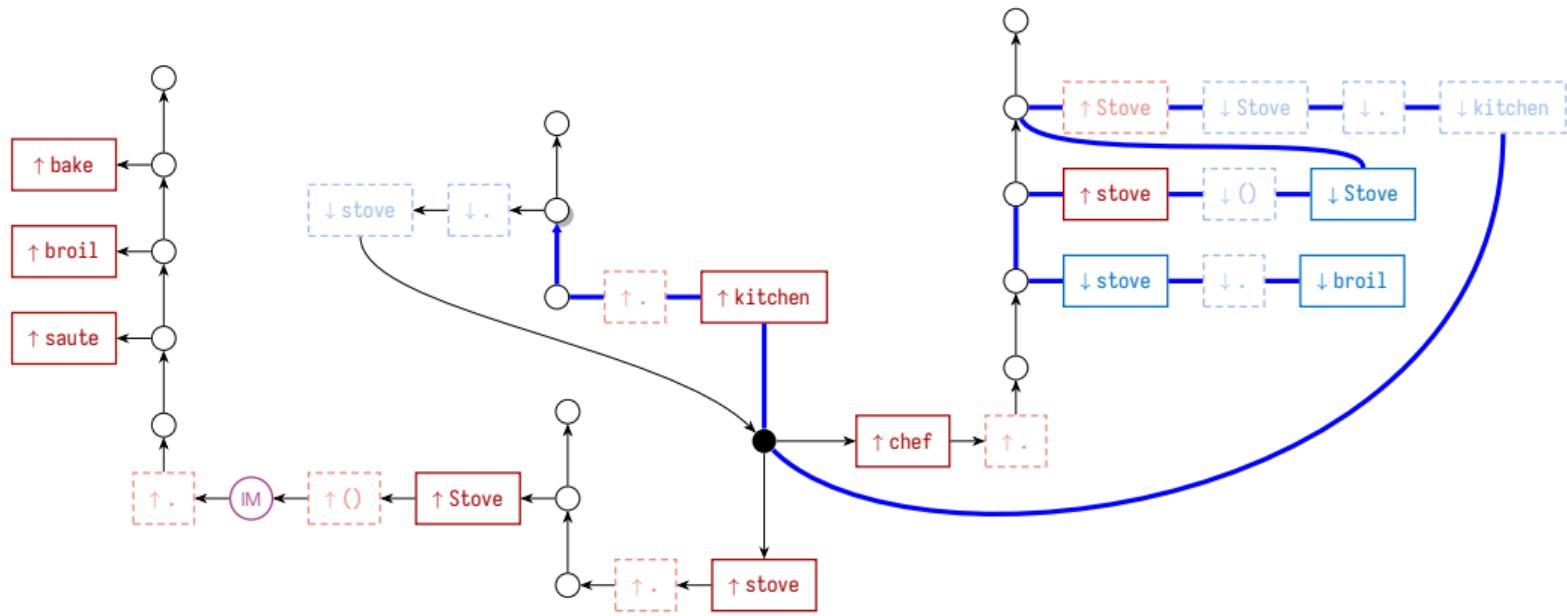
Symbol stack: $\langle .\text{Stove}().\text{broil} \rangle$



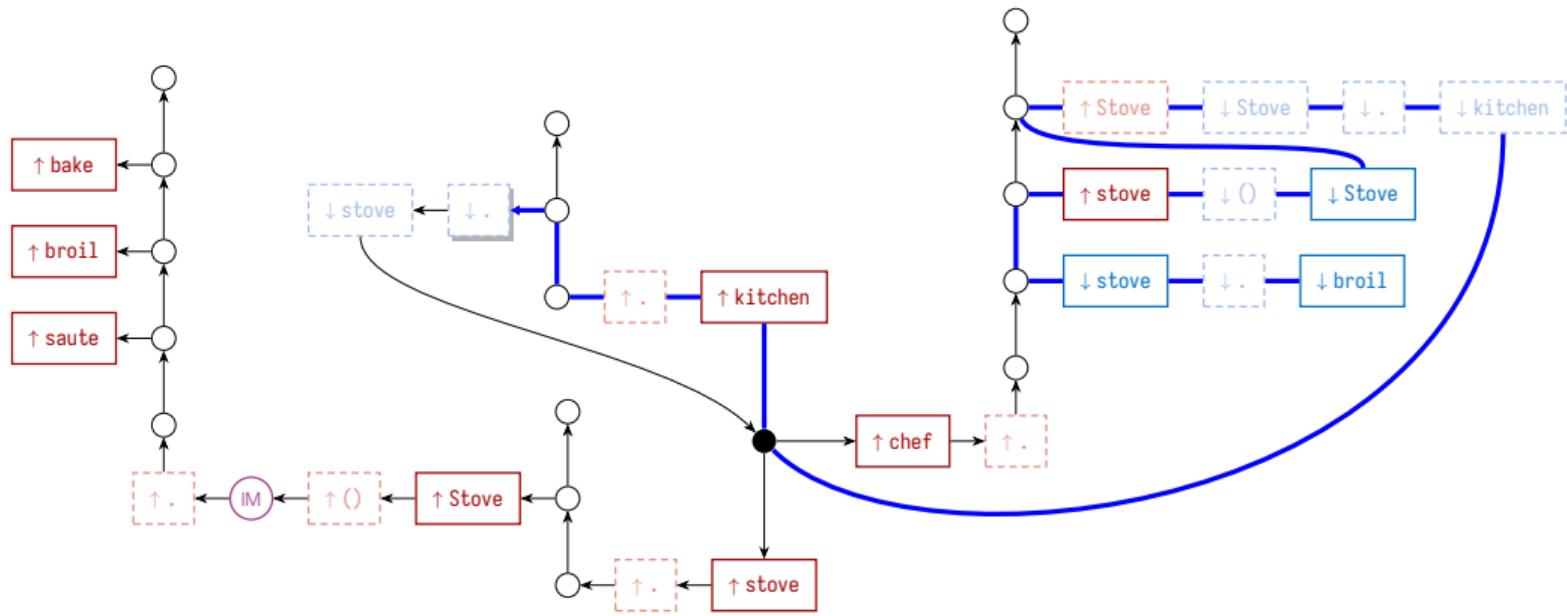
Symbol stack: $\langle \text{Stove}().\text{broil} \rangle$



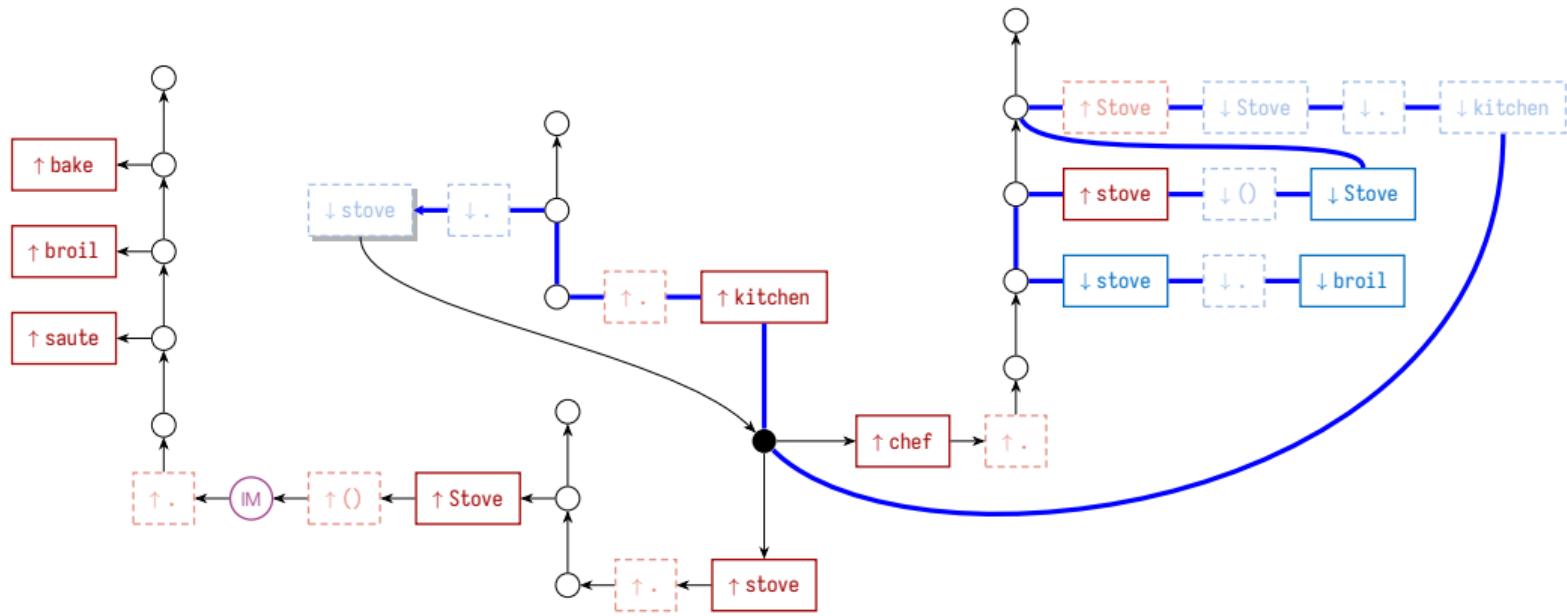
Symbol stack: $\langle \text{Stove}().\text{broil} \rangle$



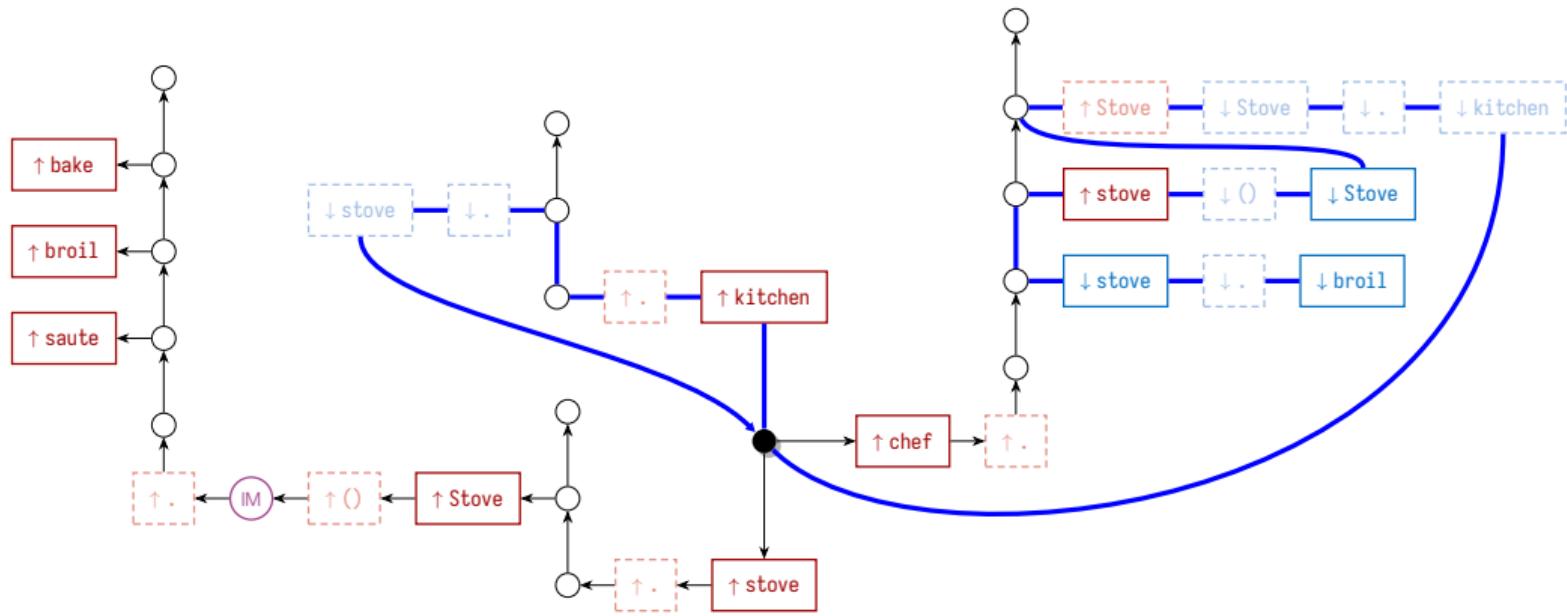
Symbol stack: $\langle \text{Stove}().\text{broil} \rangle$



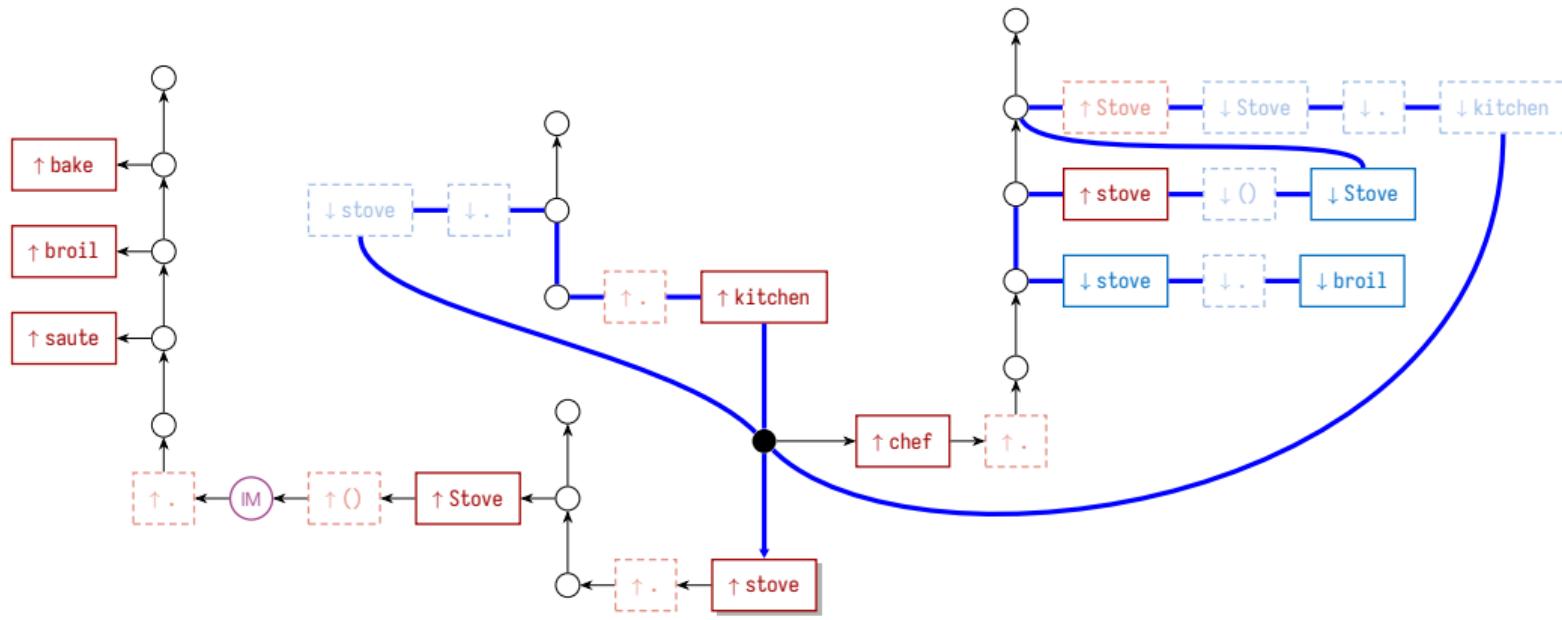
Symbol stack: $\langle \cdot \text{Stove}().\text{broil} \rangle$



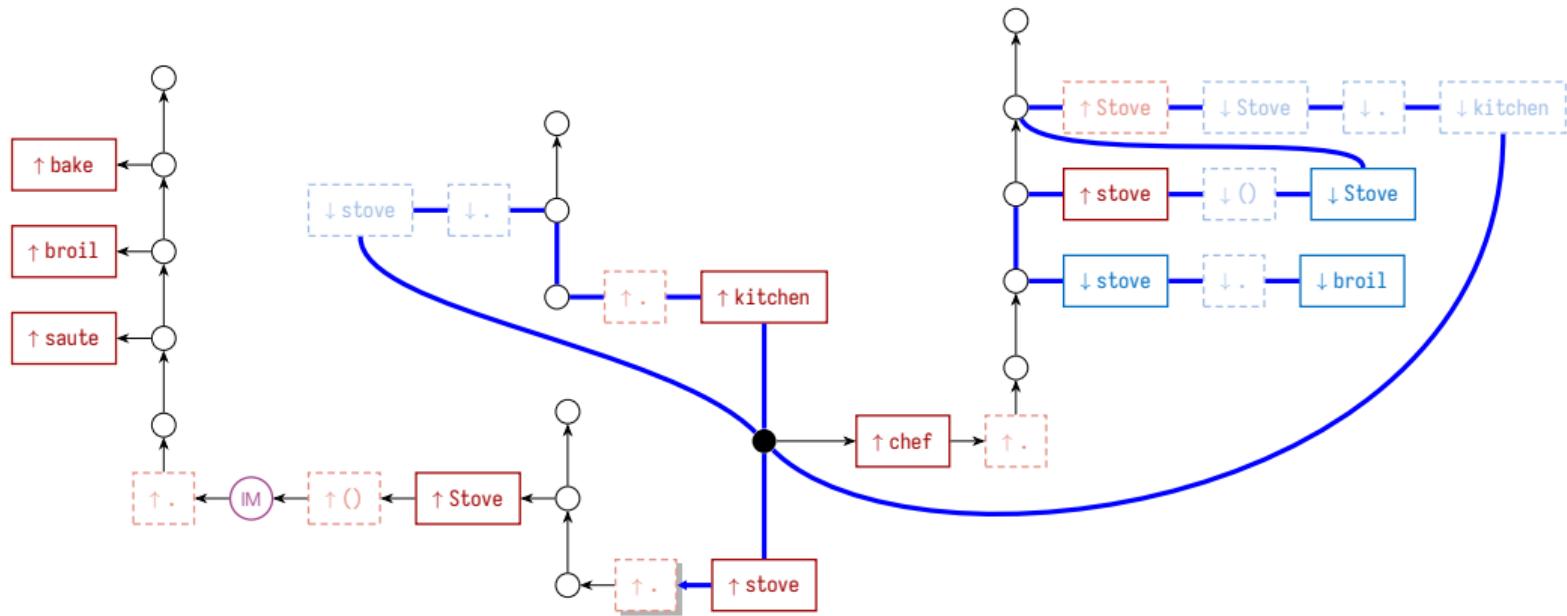
Symbol stack: $\langle \text{stove}.\text{Stove}().\text{broil} \rangle$



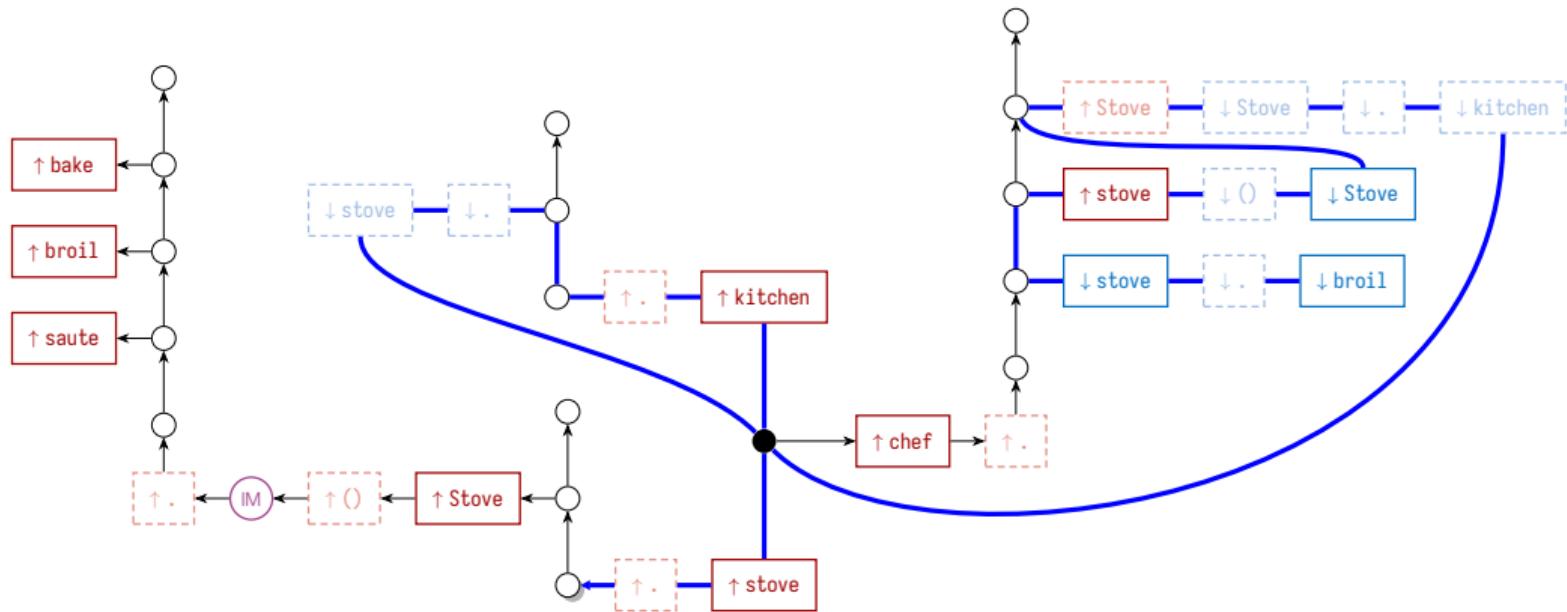
Symbol stack: $\langle \text{stove}.\text{Stove}().\text{broil} \rangle$



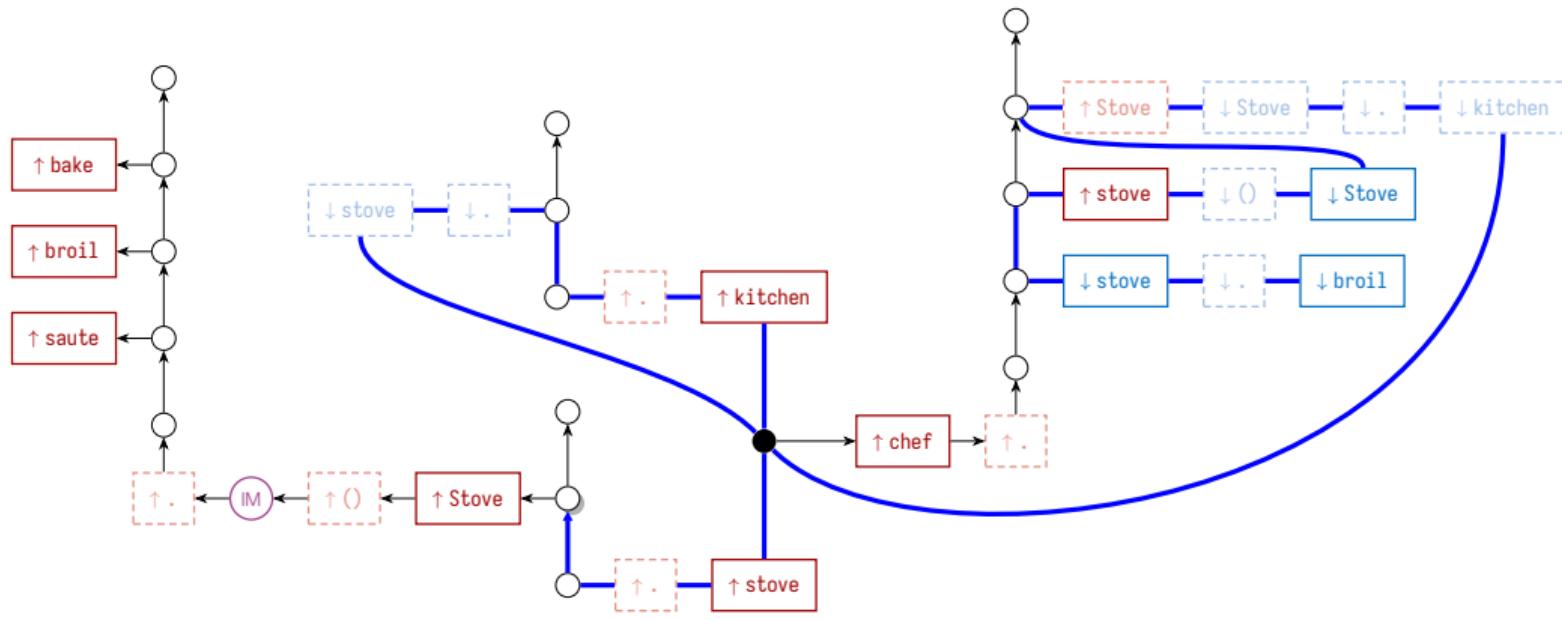
Symbol stack: <.Stove().broil>



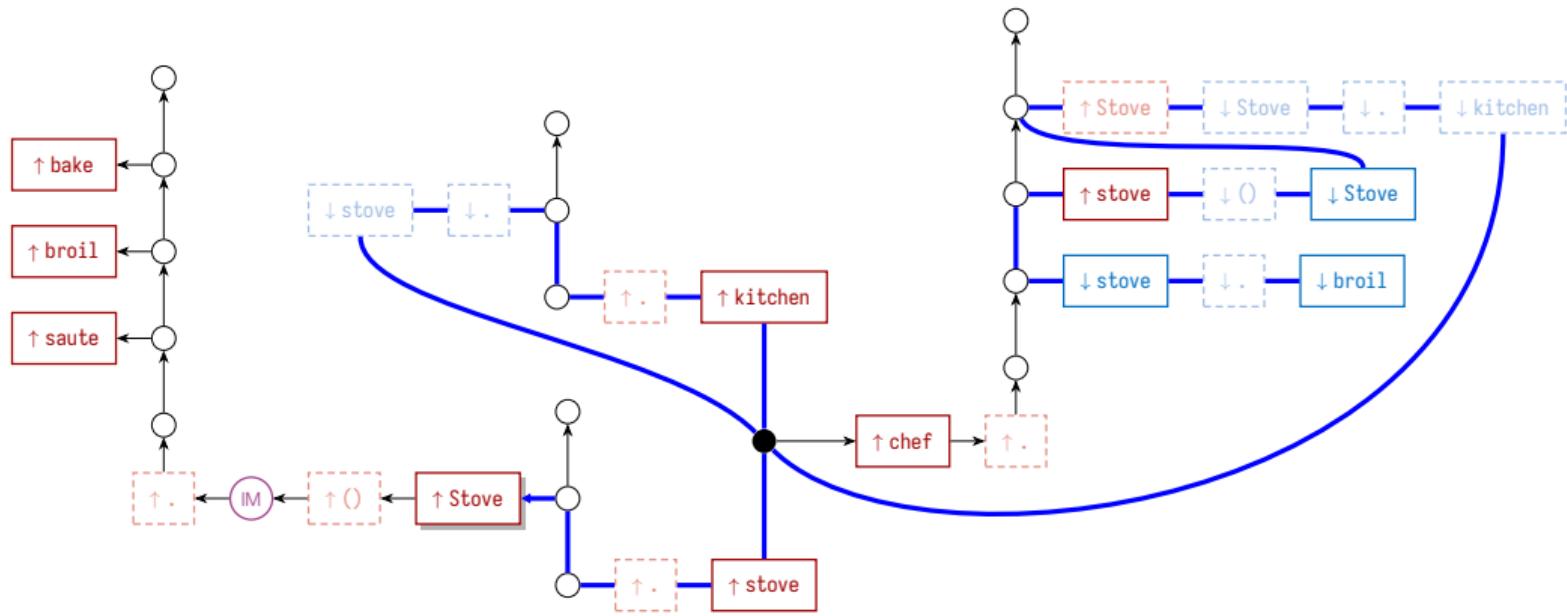
Symbol stack: $\langle \text{Stove}().\text{broil} \rangle$



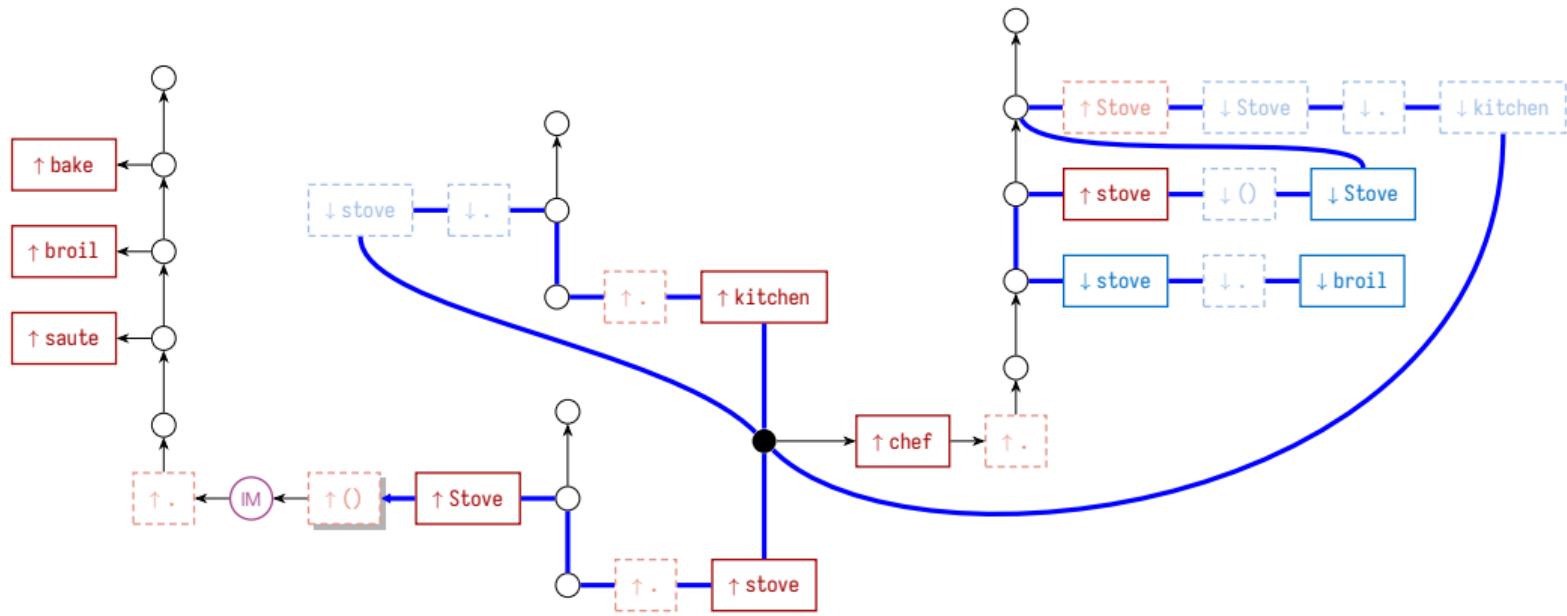
Symbol stack: $\langle \text{Stove}().\text{broil} \rangle$



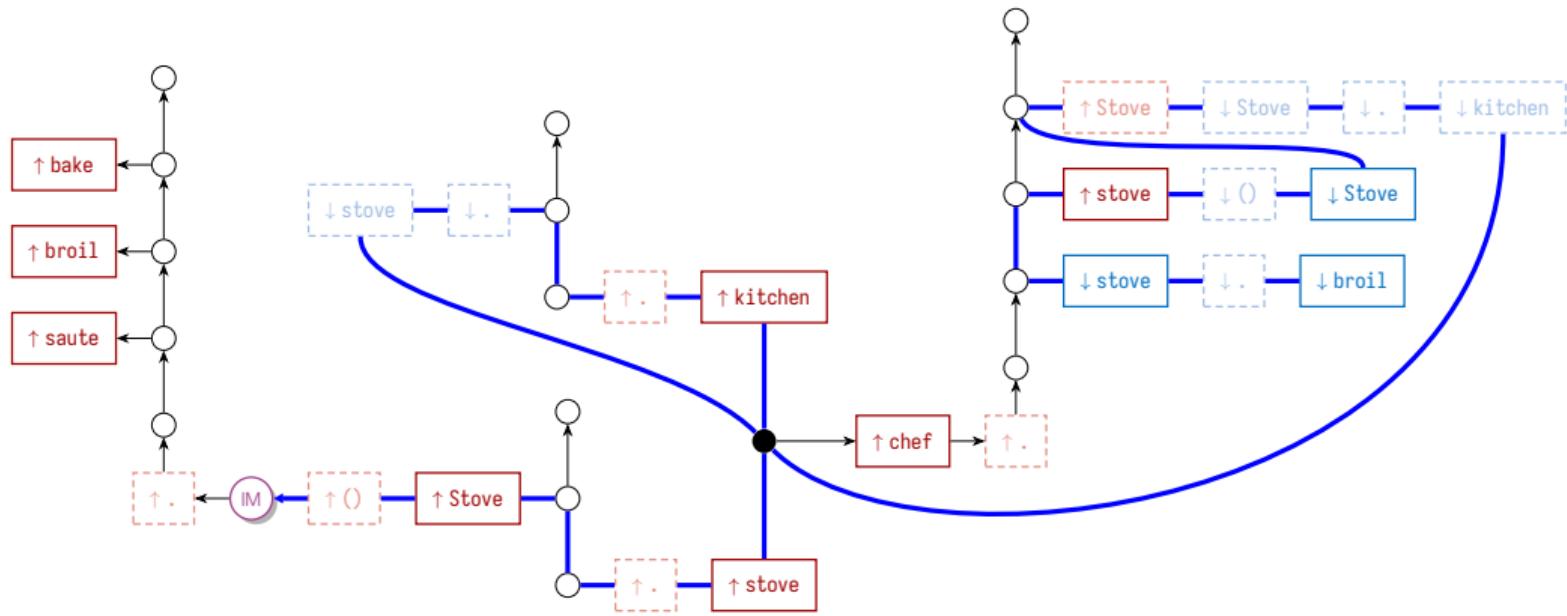
Symbol stack: <Stove().broil>



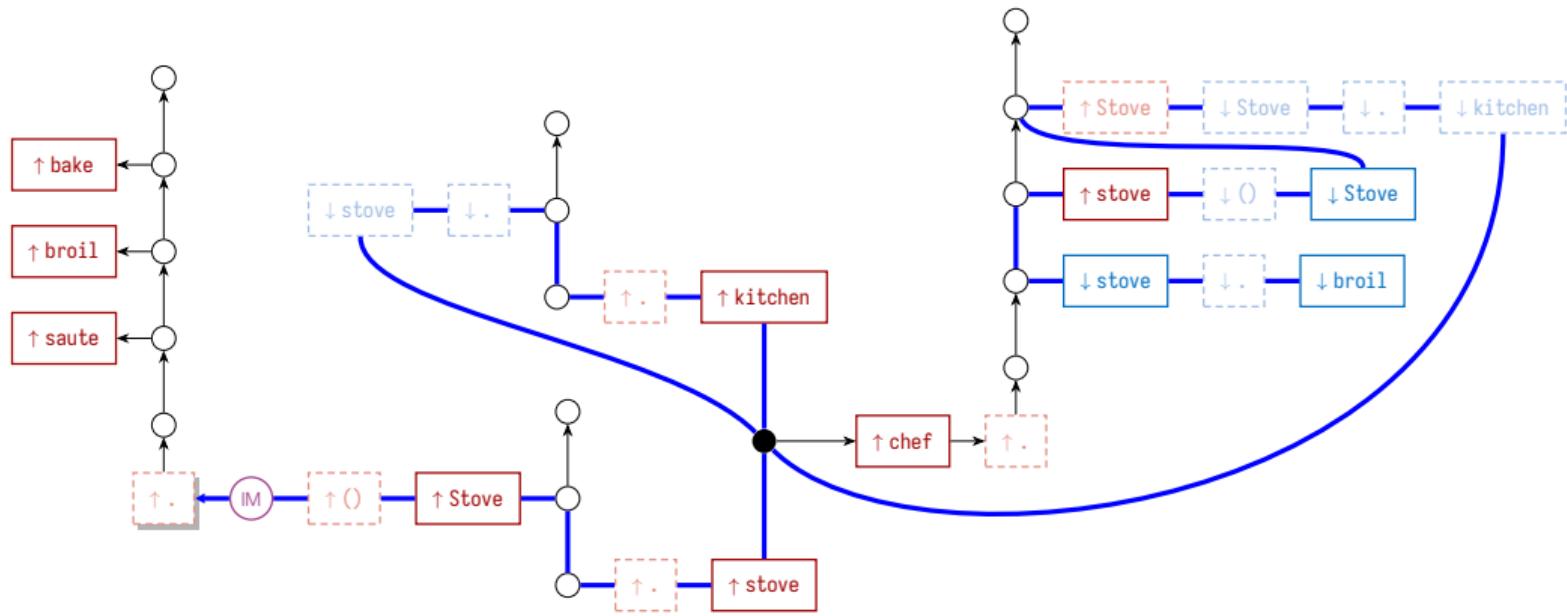
Symbol stack: $\langle () . \text{broil} \rangle$



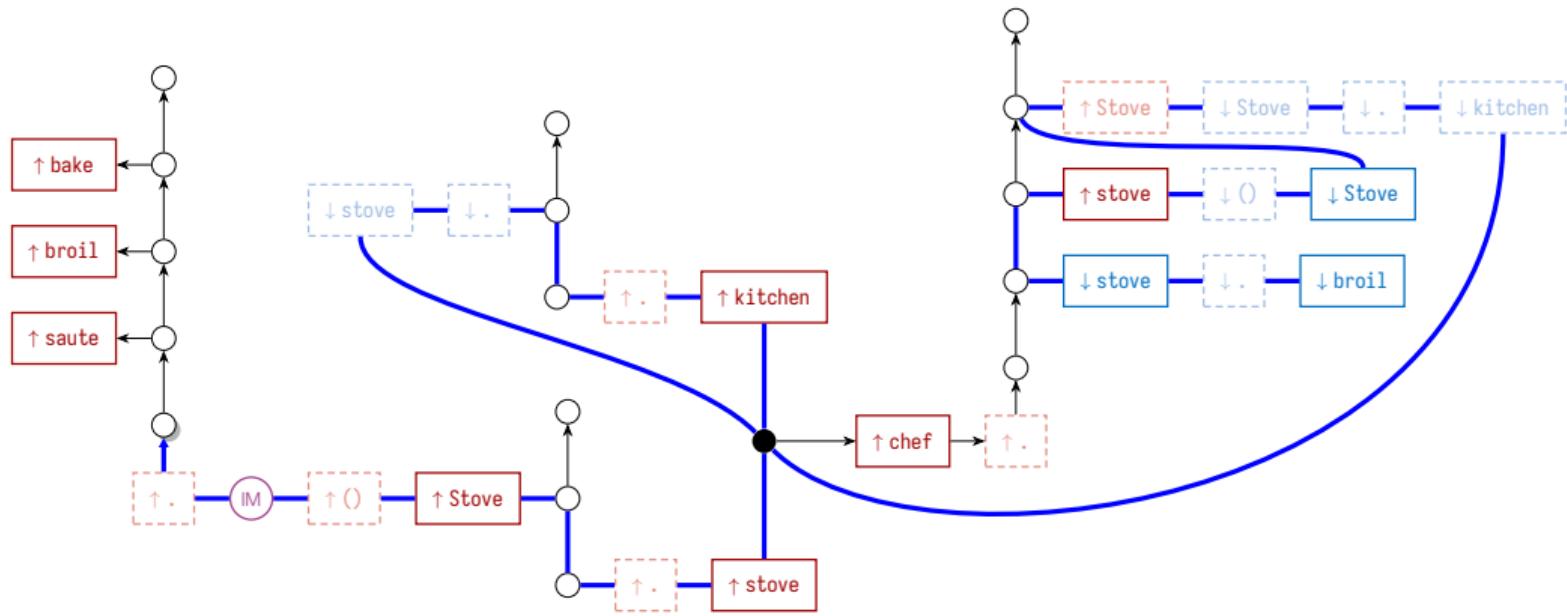
Symbol stack: $\langle \text{.broil} \rangle$



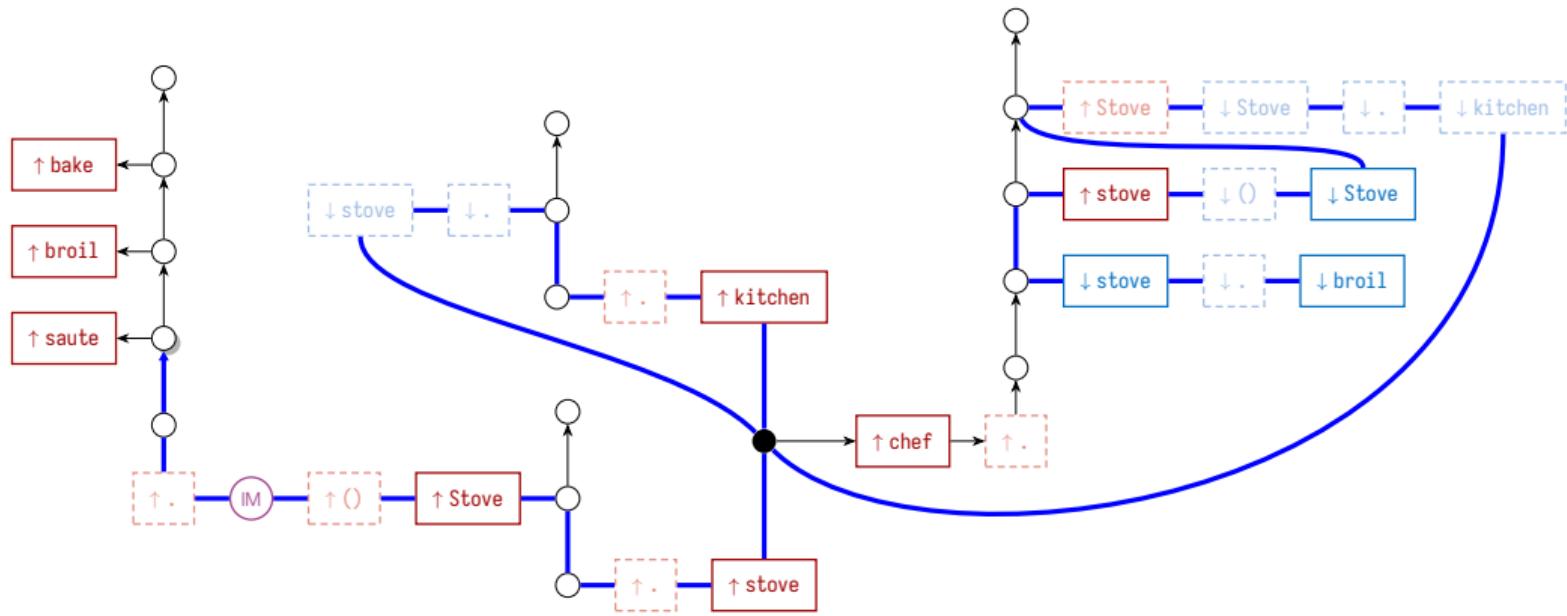
Symbol stack: ⟨ .broil ⟩



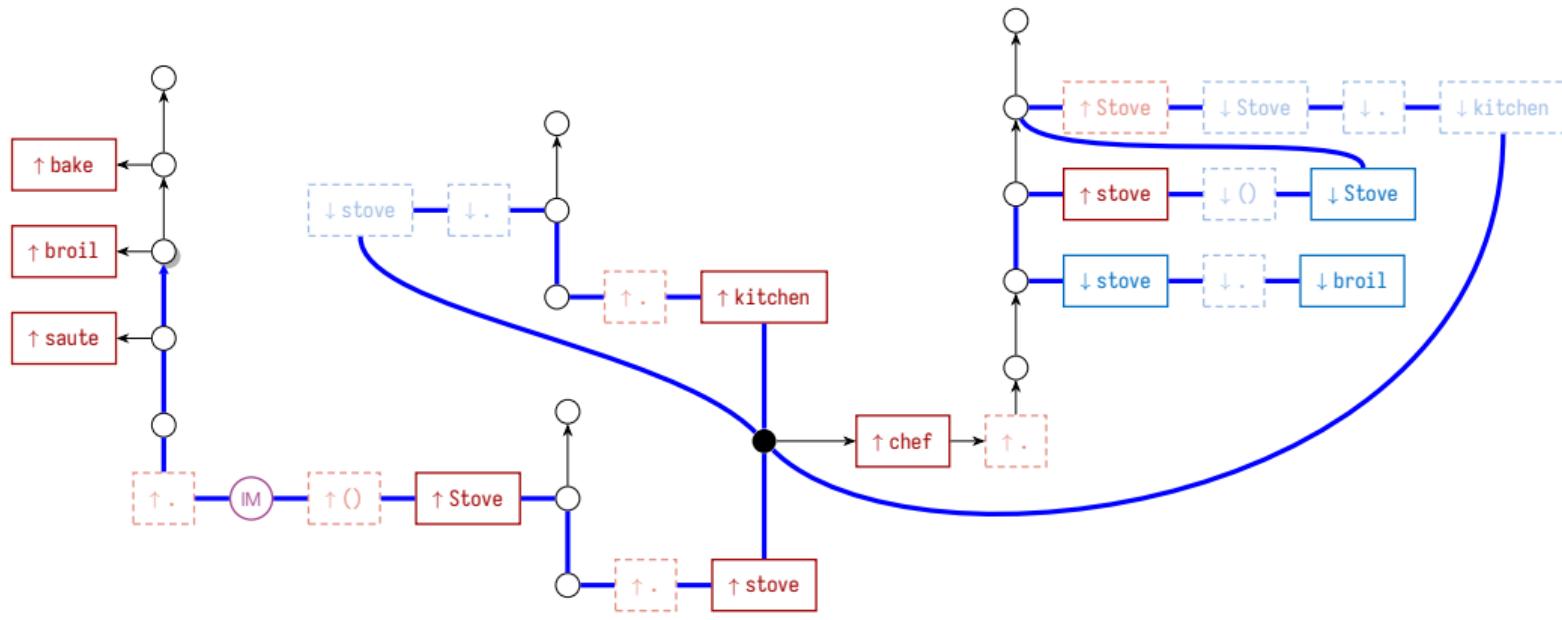
Symbol stack: $\langle \text{broil} \rangle$



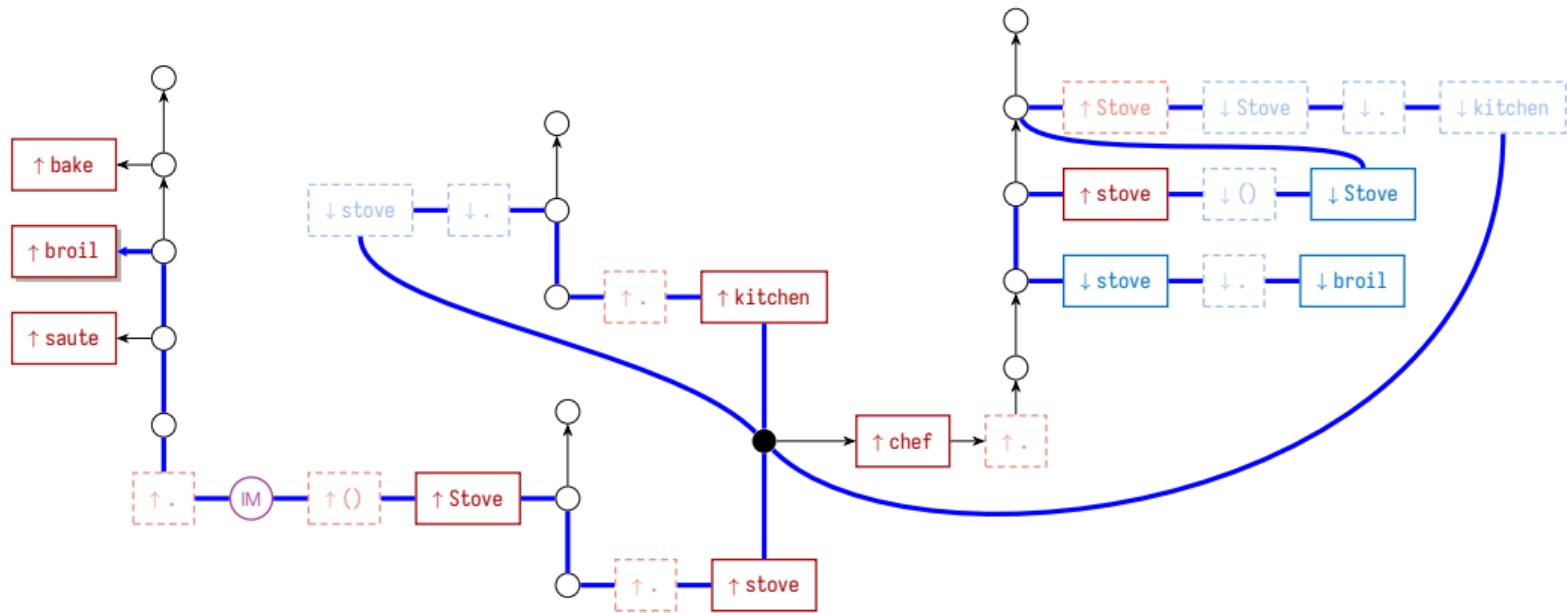
Symbol stack: $\langle \text{broil} \rangle$



Symbol stack: $\langle \text{broil} \rangle$



Symbol stack: $\langle \text{broil} \rangle$



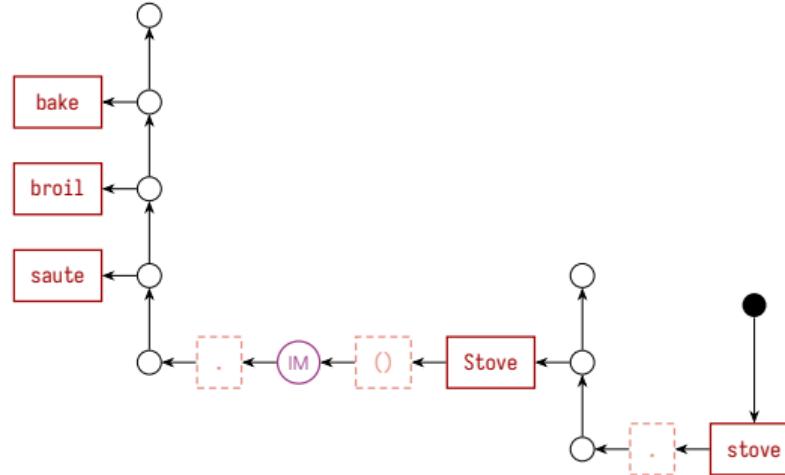
Symbol stack: \diamond

A photograph showing a large-scale bridge construction project. In the foreground, a white barge with the name "BENJAMIN FOSS" is carrying a massive, dark grey concrete bridge girder. The girder is suspended from a red lattice boom crane standing on the barge. To the left, a green truss bridge spans a river, with its internal structure visible. In the background, another bridge is under construction, with long concrete piers and a partially completed span. The sky is clear and blue.

Partial paths

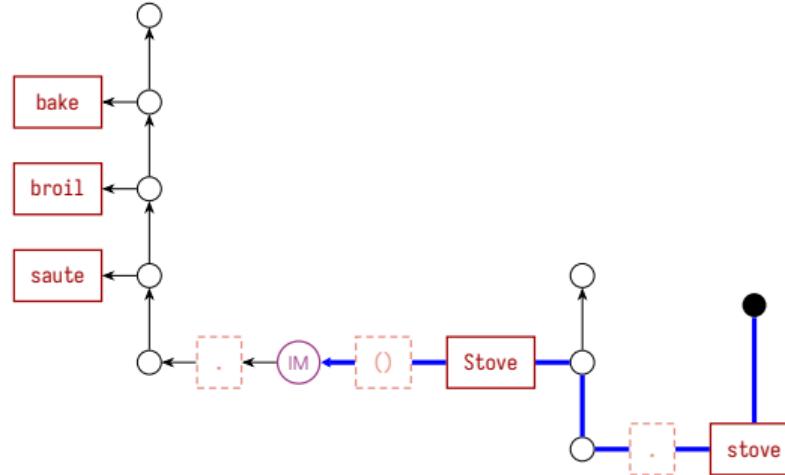
Per-file stack graphs

```
stove.py  
-----  
class Stove(object):  
    def bake(self):  
        pass  
  
    def broil(self):  
        pass  
  
    def saute(self):  
        pass
```



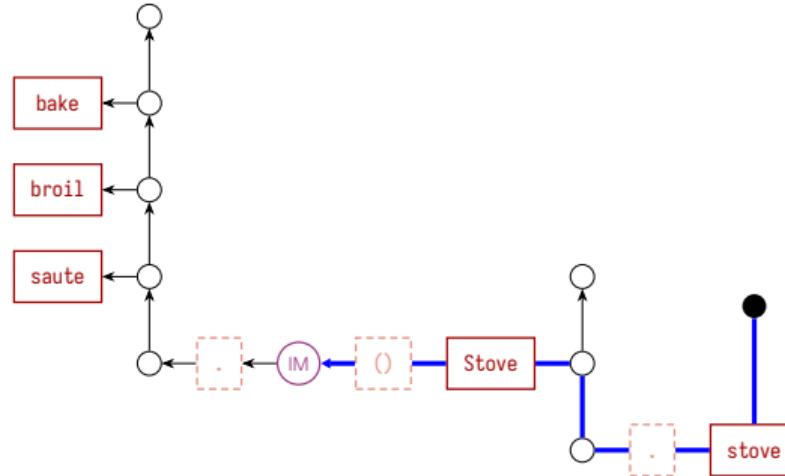
Per-file stack graphs

```
stove.py  
-----  
class Stove(object):  
    def bake(self):  
        pass  
  
    def broil(self):  
        pass  
  
    def saute(self):  
        pass
```



Per-file stack graphs

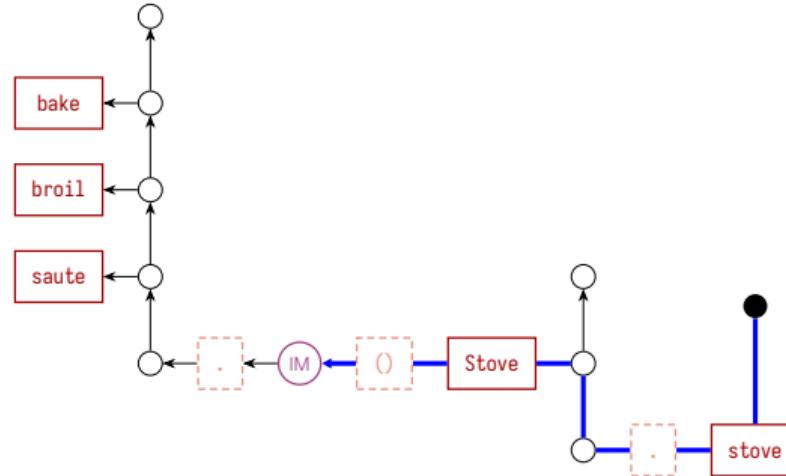
```
stove.py  
-----  
class Stove(object):  
    def bake(self):  
        pass  
  
    def broil(self):  
        pass  
  
    def saute(self):  
        pass
```



$$\{\langle \text{stove.Stove}() \rangle \cdot \psi\} \quad \bullet \rightsquigarrow \textcircled{IM} \quad \{\psi\}$$

Per-file stack graphs

```
stove.py  
-----  
class Stove(object):  
    def bake(self):  
        pass  
  
    def broil(self):  
        pass  
  
    def saute(self):  
        pass
```

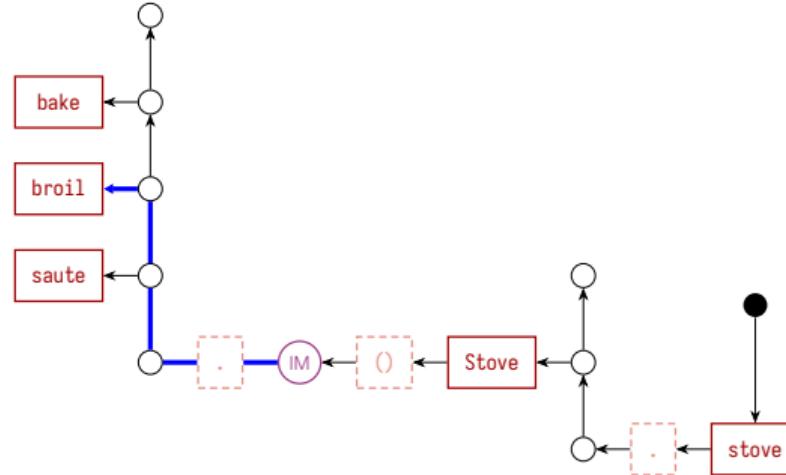


$$\{\langle \text{stove.Stove}() \rangle \cdot \psi\} \quad \bullet \rightsquigarrow \textcircled{IM} \quad \{\psi\}$$

Invoking `stove.Stove`
gives you an instance of the `Stove` class.

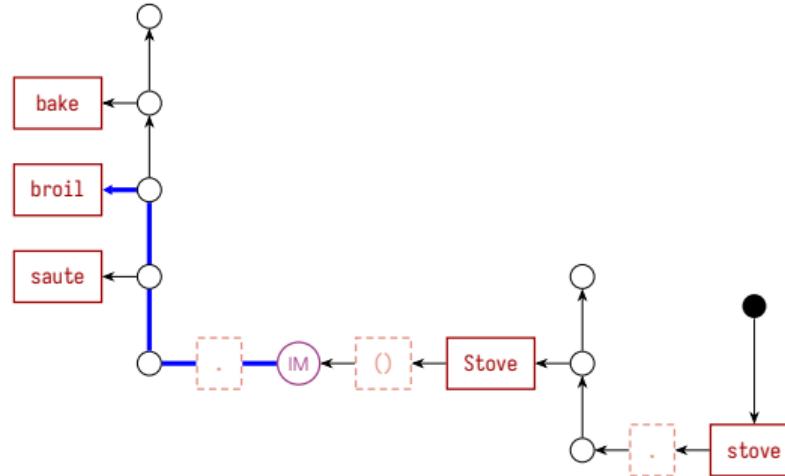
Per-file stack graphs

```
stove.py  
-----  
class Stove(object):  
    def bake(self):  
        pass  
  
    def broil(self):  
        pass  
  
    def saute(self):  
        pass
```



Per-file stack graphs

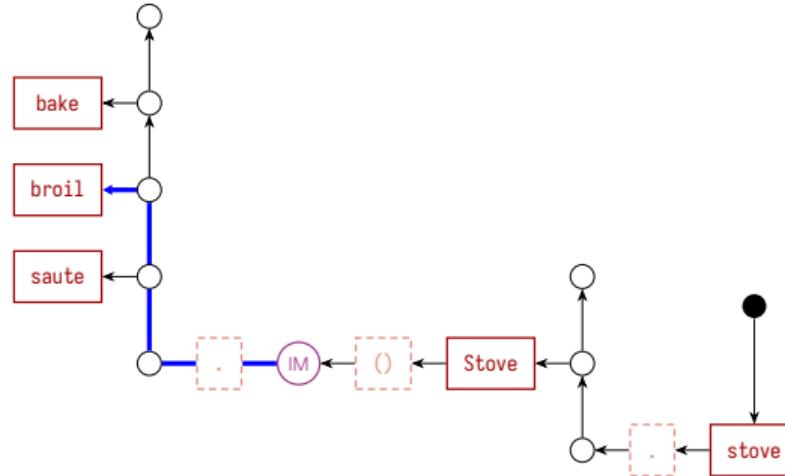
```
stove.py  
-----  
class Stove(object):  
    def bake(self):  
        pass  
  
    def broil(self):  
        pass  
  
    def saute(self):  
        pass
```



$$\{\langle \cdot.\text{broil} \rangle \cdot \psi\} \xrightarrow{\text{IM}} \uparrow \text{broil} \quad \{\psi\}$$

Per-file stack graphs

```
stove.py  
-----  
class Stove(object):  
    def bake(self):  
        pass  
  
    def broil(self):  
        pass  
  
    def saute(self):  
        pass
```

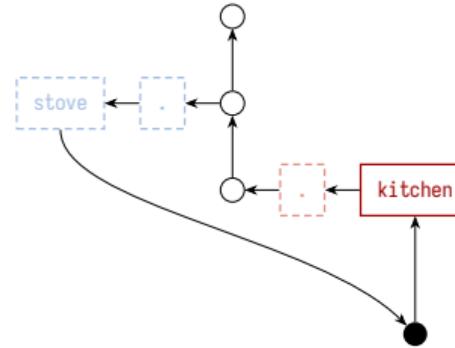


$$\{\langle \cdot.\text{broil} \rangle \cdot \psi\} \xrightarrow{\text{IM}} \uparrow \text{broil} \quad \{\psi\}$$

The `Stove` class
has an instance member named `broil`
defined at `stove.py:5:9`.

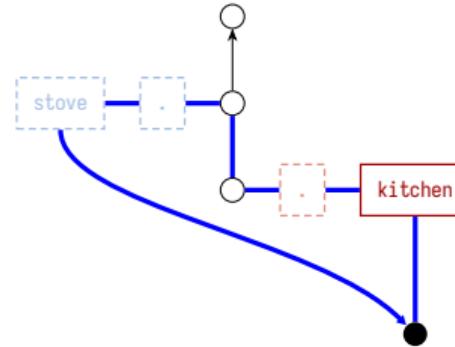
Per-file stack graphs

```
kitchen.py  
from stove import *
```



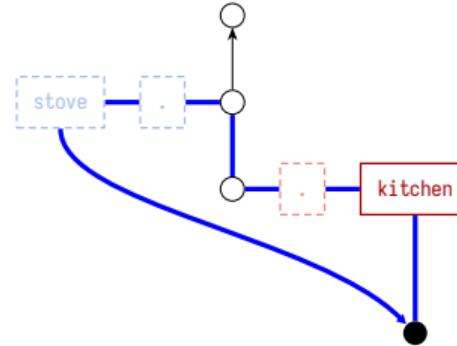
Per-file stack graphs

```
kitchen.py  
from stove import *
```



Per-file stack graphs

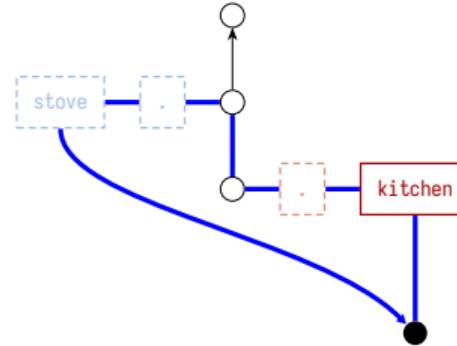
```
kitchen.py  
from stove import *
```



$$\{\langle \text{kitchen.} \rangle \cdot \psi\} \quad \bullet \rightsquigarrow \bullet \quad \{\langle \text{stove.} \rangle \cdot \psi\}$$

Per-file stack graphs

```
kitchen.py  
from stove import *
```



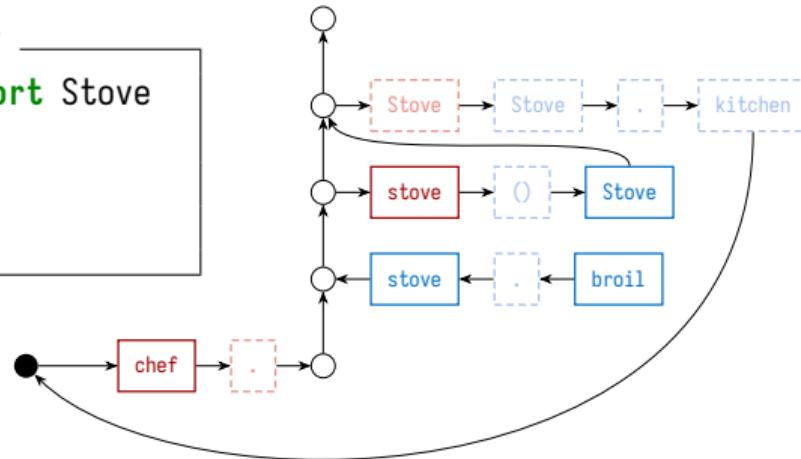
$$\{\langle \text{kitchen.} \rangle \cdot \psi\} \quad \bullet \rightsquigarrow \bullet \quad \{\langle \text{stove.} \rangle \cdot \psi\}$$

If you are looking for `kitchen.[anything]`
then you might find it at `stove.[anything]`.

Per-file stack graphs

```
chef.py
from kitchen import Stove

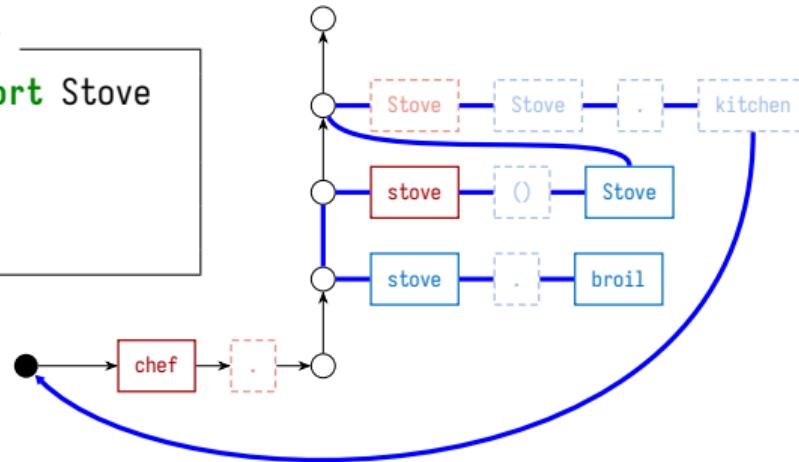
stove = Stove()
stove.broil()
```



Per-file stack graphs

```
chef.py
from kitchen import Stove

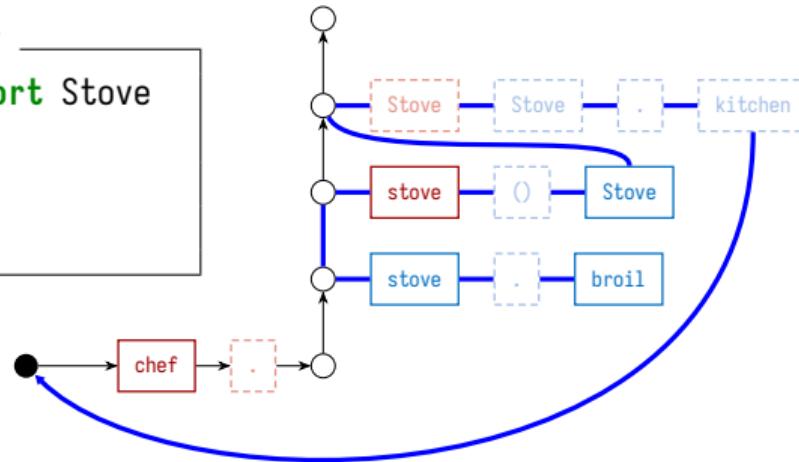
stove = Stove()
stove.broil()
```



Per-file stack graphs

```
chef.py
from kitchen import Stove

stove = Stove()
stove.broil()
```

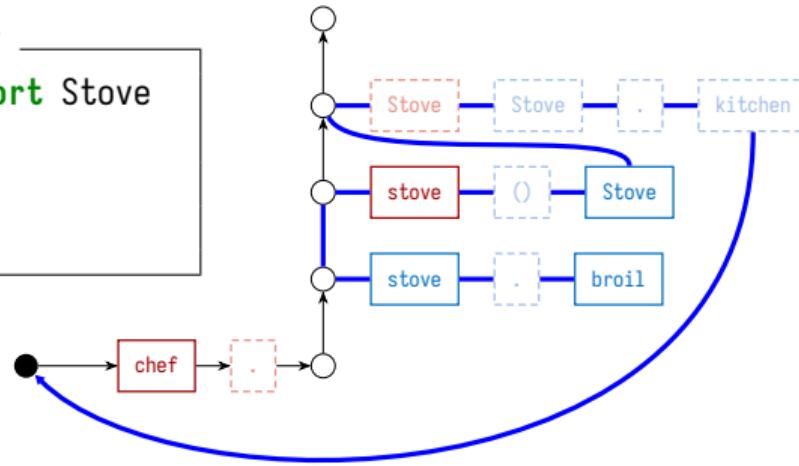


$\{\diamond\}$ $\downarrow \text{broil}$ $\rightsquigarrow \bullet \quad \{\langle \text{kitchen.Stove().broil} \rangle\}$

Per-file stack graphs

```
chef.py
from kitchen import Stove

stove = Stove()
stove.broil()
```



$\{◊\} \downarrow \text{broil} \rightsquigarrow \bullet \{(\text{kitchen}.Stove().broil)\}$

If you can find what `kitchen.Stove` resolves to and can call it
then the result should have a member named `broil`
which the reference at `chef.py:4:7` resolves to.

Concatenating partial paths

```
{◊} ↓ broil ↵ {⟨kitchen.Stove().broil⟩}
```

If you can find what `kitchen.Stove` resolves to and can call it, then the result should have a member named `broil` which the reference at `chef.py:4:7` resolves to.

Concatenating partial paths

$\{\diamond\} \xrightarrow{\downarrow \text{broil}} \bullet \{\langle \text{kitchen.Stove()}.broil \rangle\}$

+

$\{\langle \text{kitchen.} \rangle \cdot \psi\} \bullet \rightsquigarrow \bullet \{\langle \text{stove.} \rangle \cdot \psi\}$

If you can find what `kitchen.Stove` resolves to and can call it, then the result should have a member named `broil` which the reference at `chef.py:4:7` resolves to.

+

If you are looking for `kitchen.[anything]` then you might find it at `stove.[anything]`.

Concatenating partial paths

$$\{\diamond\} \xrightarrow{\downarrow \text{broil}} \bullet \{\langle \text{kitchen.Stove()}.broil \rangle\} + \{\langle \text{kitchen.} \rangle \cdot \psi\} \bullet \rightsquigarrow \bullet \{\langle \text{stove.} \rangle \cdot \psi\}$$
$$\psi = \langle \text{Stove()}.broil \rangle$$

If you can find what `kitchen.Stove` resolves to and can call it, then the result should have a member named `broil` which the reference at `chef.py:4:7` resolves to.

+

If you are looking for `kitchen.[anything]` then you might find it at `stove.[anything]`.

Concatenating partial paths

```
{◊} ↓ broil ↪ ● {⟨stove.Stove().broil⟩}
```

If you can find what `stove.Stove` resolves to and can call it, then the result should have a member named `broil` which the reference at *chef.py:4:7* resolves to.

Concatenating partial paths

$\{\diamond\}$  $\rightsquigarrow \bullet \{(\text{stove.Stove()}.broil)\}$

+

$\{(\text{stove.Stove()}) \cdot \psi\}$ $\bullet \rightsquigarrow \text{(IM)} \{\psi\}$

If you can find what `stove.Stove` resolves to and can call it, then the result should have a member named `broil` which the reference at `chef.py:4:7` resolves to.

+

Invoking `stove.Stove` gives you an instance of the `Stove` class.

Concatenating partial paths

$$\{\diamond\} \xrightarrow{\downarrow \text{broil}} \bullet \{\langle \text{stove.Stove()}.broil \rangle\} + \{\langle \text{stove.Stove()} \cdot \psi \rangle\} \bullet \xrightarrow{\text{IM}} \circledcirc \{\psi\}$$
$$\psi = \langle .\text{broil} \rangle$$

If you can find what `stove.Stove` resolves to and can call it, then the result should have a member named `broil` which the reference at `chef.py:4:7` resolves to.

+

Invoking `stove.Stove` gives you an instance of the `Stove` class.

Concatenating partial paths

```
{diamond} ↓ broil ~~> IM {⟨.broil⟩}
```

The `Stove` class
should have a member
named `broil` which the reference at
`chef.py:4:7` resolves to.

Concatenating partial paths

$\{\diamond\}$  \rightsquigarrow  $\{(\cdot.\text{broil})\}$

+

$\{(\cdot.\text{broil}) \cdot \psi\}$  \rightsquigarrow  $\{\psi\}$

The `Stove` class
should have a member
named `broil` which the reference at
chef.py:4:7 resolves to.

+

The `Stove` class has an
instance member named `broil`
defined at *stove.py:5:9*.

Concatenating partial paths

$$\{\diamond\} \boxed{\downarrow \text{broil}} \rightsquigarrow \textcolor{pink}{(\text{IM})} \{\langle .\text{broil} \rangle\} + \{\langle .\text{broil} \rangle \cdot \psi\} \textcolor{pink}{(\text{IM})} \rightsquigarrow \boxed{\uparrow \text{broil}} \{\psi\}$$
$$\psi = \diamond$$

The `Stove` class
should have a member
named `broil` which the reference at
`chef.py:4:7` resolves to.

+

The `Stove` class has an
instance member named `broil`
defined at `stove.py:5:9`.

Concatenating partial paths

```
{◊} ↓ broil ~~~> ↑ broil {◊}
```

The definition at *stove.py:5:9*
is what the reference at
chef.py:4:7 resolves to.

Picture credits

- Slide 2 Ivan Radic, "Close-up of a compass graffiti on the ground"
CC-BY-2.0, <https://flic.kr/p/2kGKmM>
- Slide 4 stantontcady, "Card catalog"
CC-BY-ND-2.0, <https://flic.kr/p/quiyyqg>
- Slide 4 neekoh.fi, "Forest Rays"
CC-BY-2.0, <https://flic.kr/p/7pFsmf>
- Slide 5 Mark Gunn, "This just tern'ed into a swarm!"
CC-BY-2.0, <https://flic.kr/p/P11JH1>
- Slide 8 Katja Schulz, "Inchworm"
CC-BY-2.0, <https://flic.kr/p/PJMP4w>
- Slide 11 Seattle Municipal Archives, "West Seattle Bridge under construction, circa 1983"
CC-BY-2.0, <https://flic.kr/p/7jKWYi>