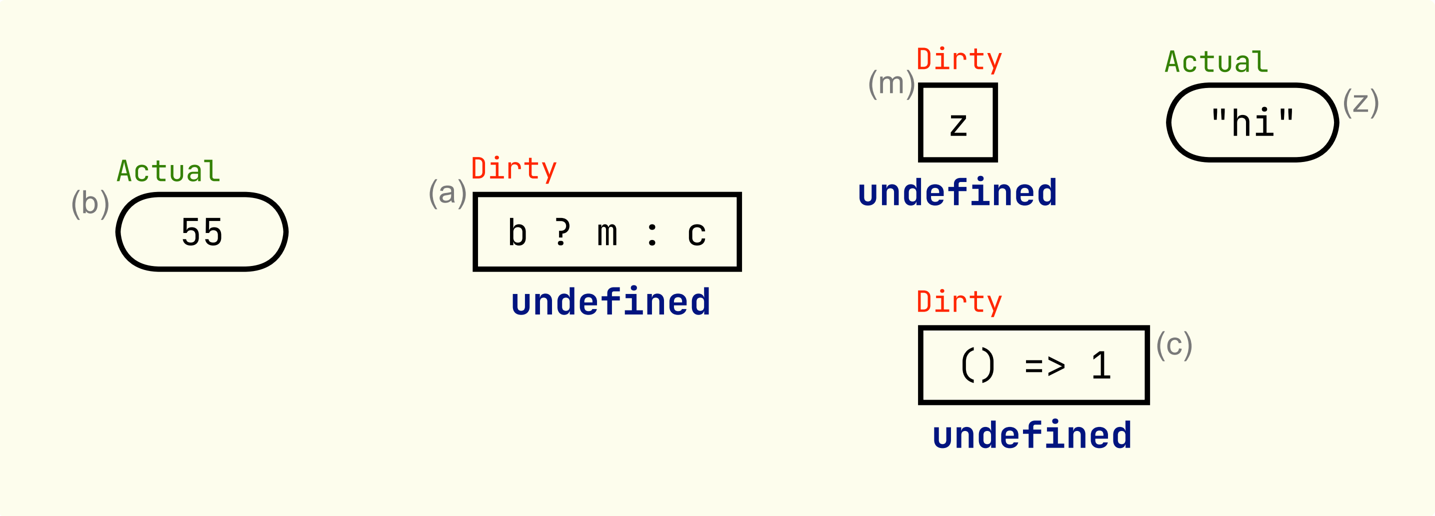
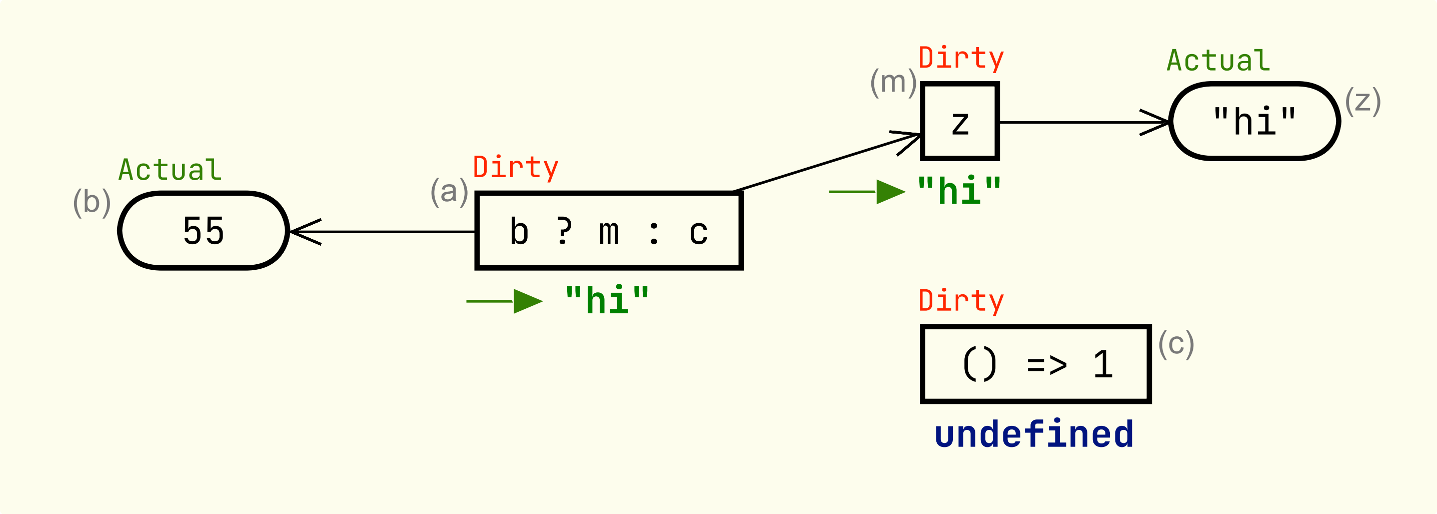
```
a = Cell(() => b.get() ? m.get() : c.get());
b = Cell(55);
m = Cell(() => z.get());
z = Cell('hi');
c = Cell(() => 1);
```



b changed
b actual
z changed
z actual

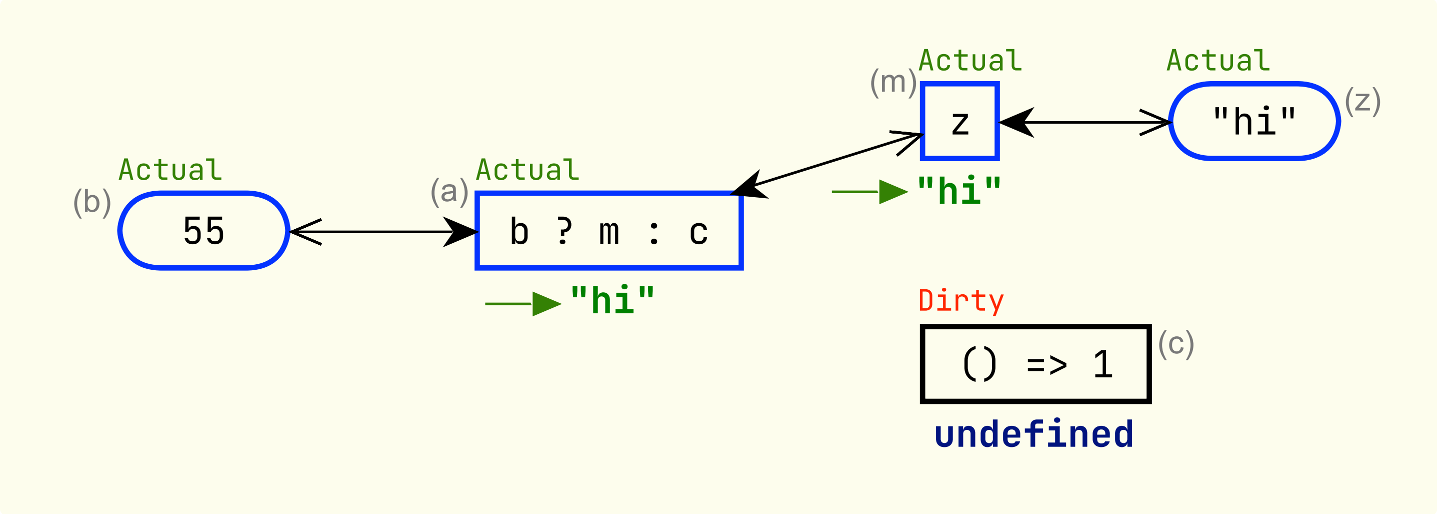
`a.get();`

actualize because a is Dirty



a pull
a +dep b
m pull
m +dep z
m changed
a +dep m
a changed

a observed
a pull
a +dep b
m pull
m +dep z
m changed
a +dep m
b +reaction a
b observed
m +reaction a
m observed
z +reaction m
z observed
m actual
a changed
a actual
a emit "change"



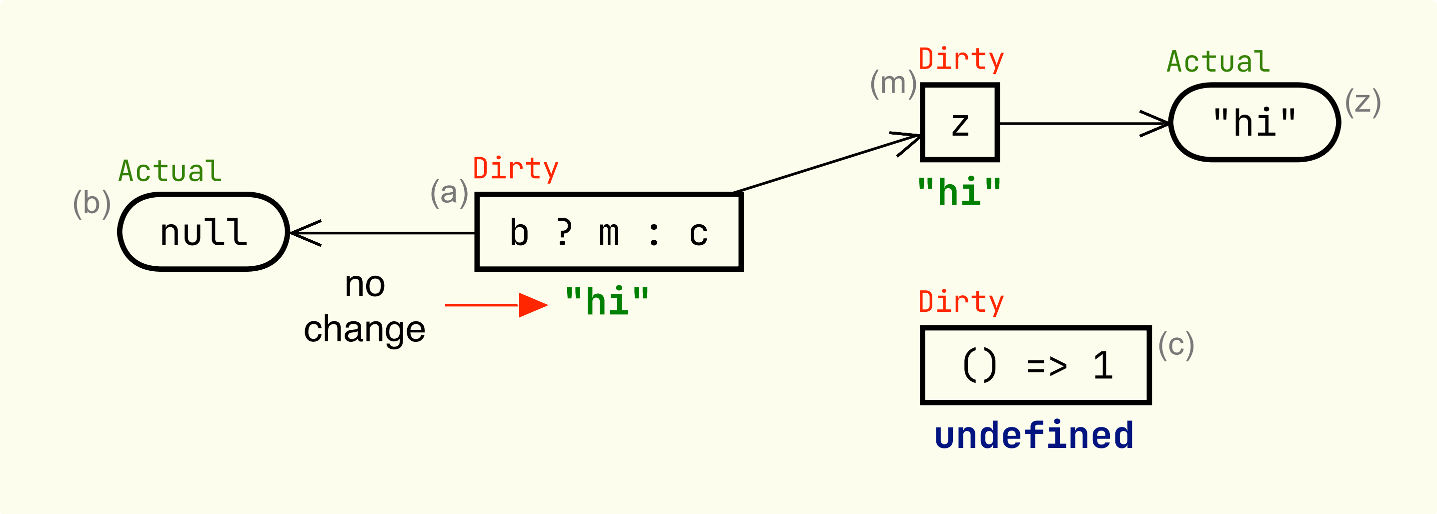
cell tree **activation**

`a.onChange(fn);`

subscribe to the "change" event for cell a

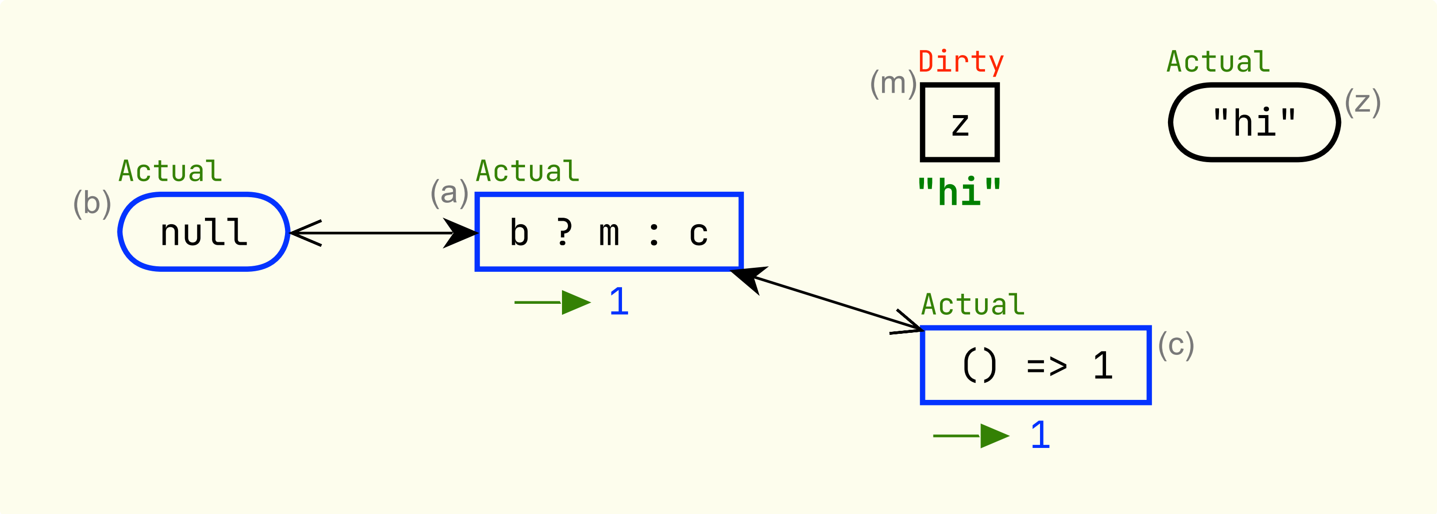
`b.set(null);`

then push
but there is nowhere to push
since b is unobservable



b changed

b changed
a dirty
a schedule
on nextTick:
a -dep m
a pull
c pull
c actual
c changed
a +dep c
m -reaction a
m unobserved
z -reaction m
z unobserved
m -dep z
m dirty
c +reaction a
a changed
a emit "change"

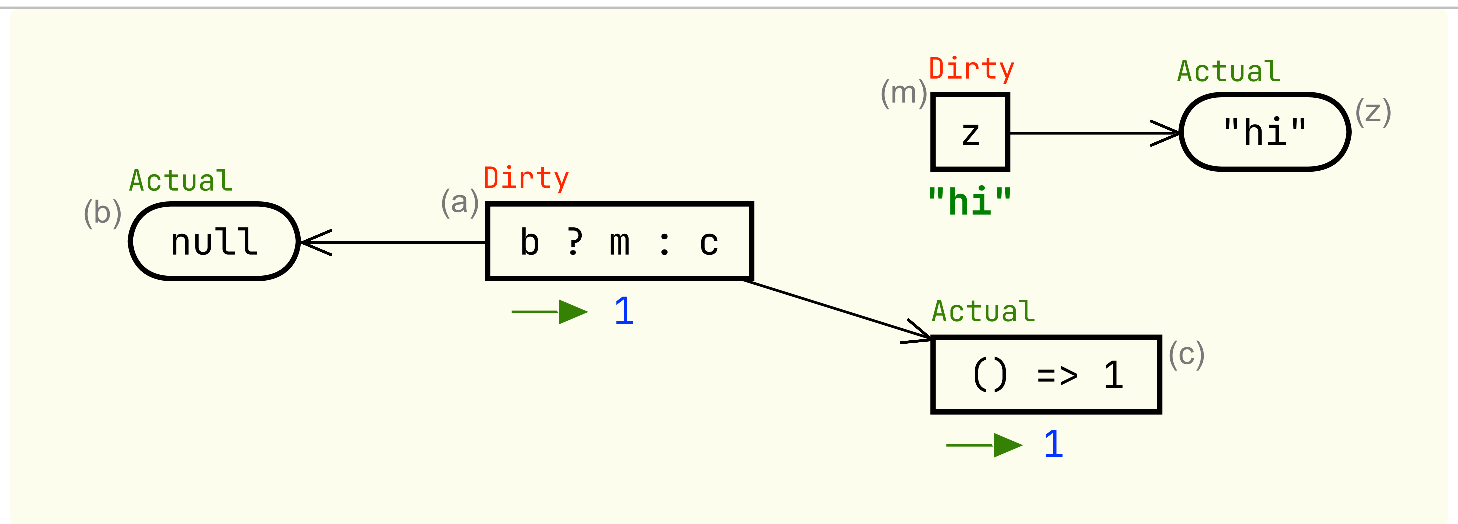


accepting changes
in
the observed tree of cells

`b.set(null);`

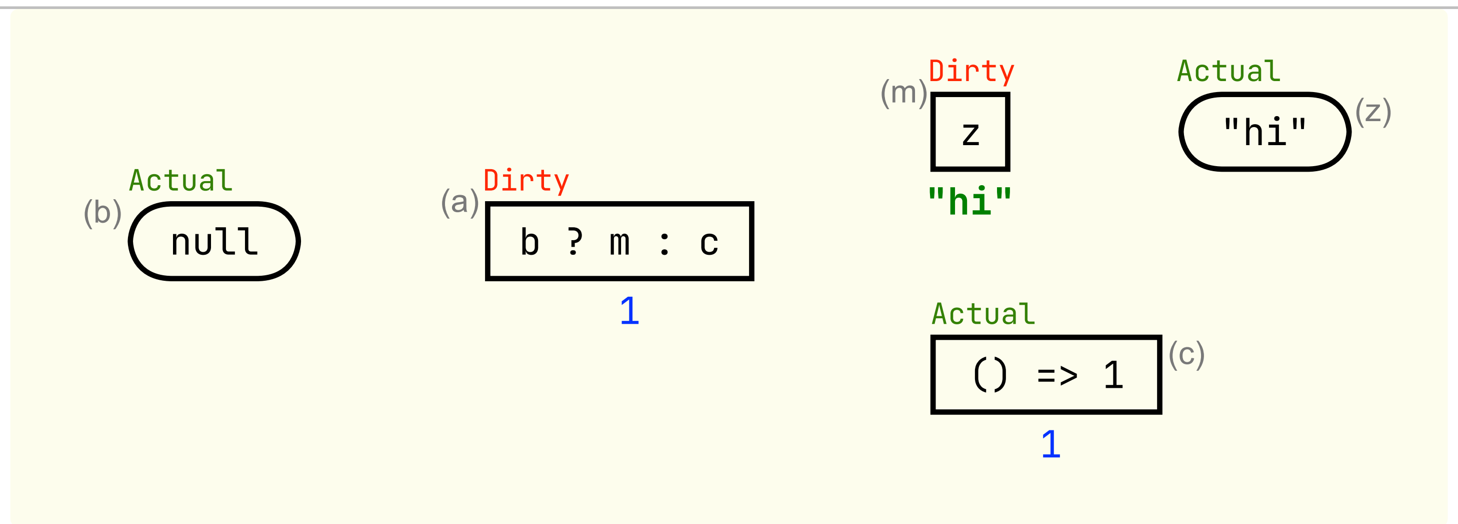
`a.get();`

actualize because a is Dirty



a -dep m
a pull
c pull
c actual
c changed
a +dep c
a changed

a unobserved
b -reaction a
b unobserved
c -reaction a
c unobserved
a dirty
a -dep b
a -dep c



cell tree **deactivation**

`a.offChange(fn);`

unsubscribe