Stabilize

and other integrity traits

Mark S. Miller Agoric

Chip Morningstar METAMASK

Richard Gibson Agoric

Mathieu Hofman Agoric

105th Plenary

December 2024

TC 39

Stabilize

and other integrity traits

Mark S. Miller Agoric.

Chip Morningstar METAMASK

Richard Gibson Agoric.

Mathieu Hofman Agoric.

JS Structs Working Group

November 2024

105th Plenary

December 2024

TC 39

Current Integrity "Levels"

- Monotonic one-way switch
- Stronger object invariants
- Proxy is X iff target is X

```
freeze                  frozen                  isFrozen
  |                       |                       |
  v                       v                       v
seal                    sealed                  isSealed
  |                       |                       |
  v                       v                       v
preventExtensions       non-extensible          !isExtensible
```

Current Integrity "Levels"

- ***Explicit*** vs Emergent
- 2 Proxy traps per ***explicit*** integrity "level"

freeze	frozen	isFrozen
↓	↓	↓
seal	sealed	isSealed
↓	↓	↓
preventExtensions	***non-extensible***	!isExtensible

Need better integrity support

frozen

↓

sealed

↓

non-extensible

Today:
`harden` is deep `freeze`

Hardened JS:
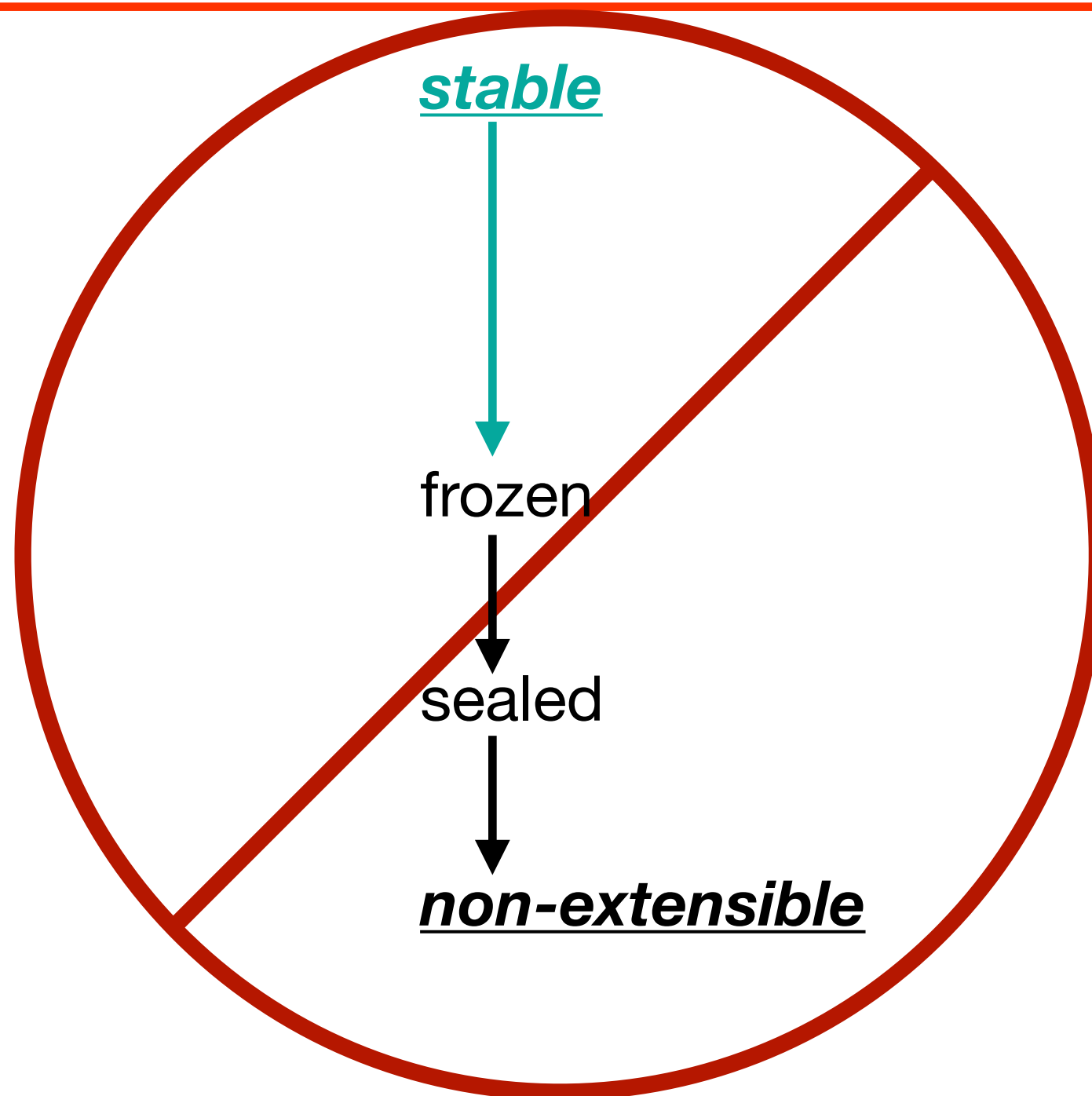hardens all primordials

3 weaknesses

First Try

First Try. But Structs Cannot be Frozen

stable

frozen

sealed

non-extensible

First Try. But Structs Cannot be Frozen

stable

↓

frozen

↓

sealed

↓

non-extensible

"Only two ways to […],
one is to bundle,
the other is to unbundle."

—Jim Barksdale

Maximally Unbundled Integrity "Traits"
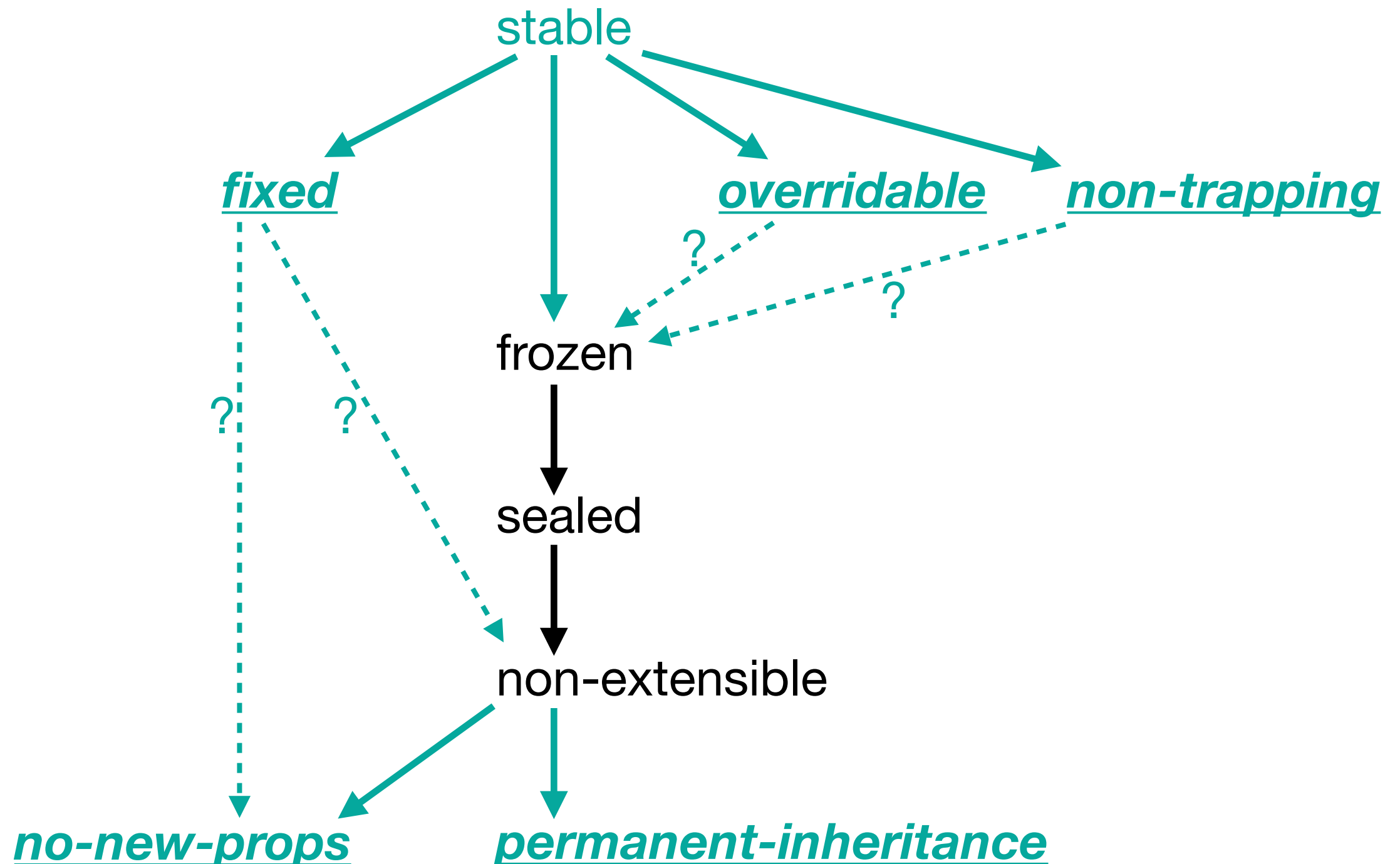
fixed
Mitigate return-override mistake

fixed

```
class Superclass {
    constructor(key) { return key; }
}

class Subclass extends Superclass {
    #value
    constructor(key, value) {
        super(key);
        this.#value = value;
    }
}

new Subclass(struct, 'a');
```

No *stamping*
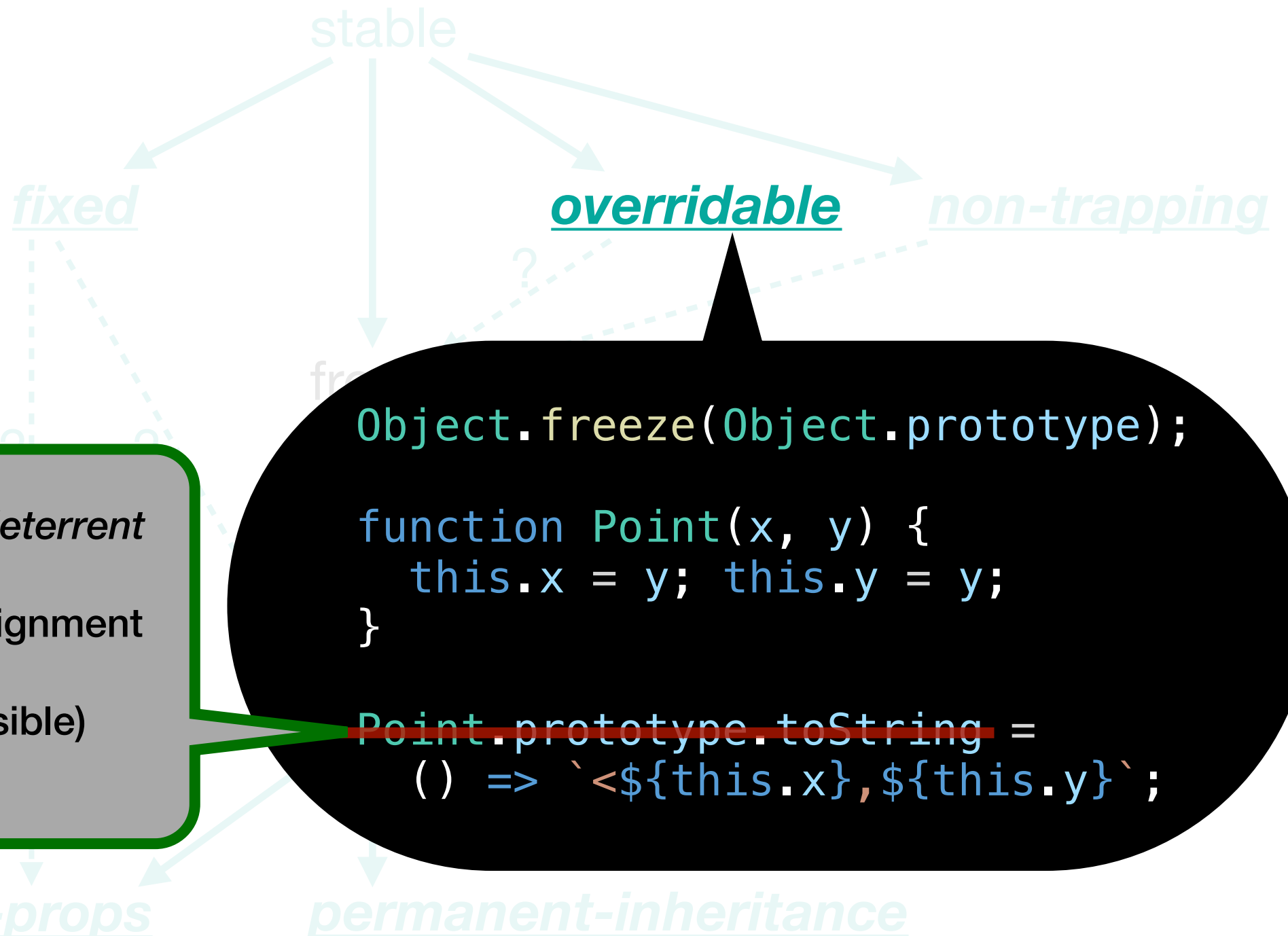
Fixed shape structs

Retcon `windowProxy`

Virtualizable weakness

no-new-props   *perm*

overridable
Mitigate assignment-override mistake

fixed **_overridable_** *non-trapping*

stable

```
Object.freeze(Object.prototype);

function Point(x, y) {
  this.x = y; this.y = y;
}

Point.prototype.toString =
  () => `<${this.x},${this.y}>`;
```
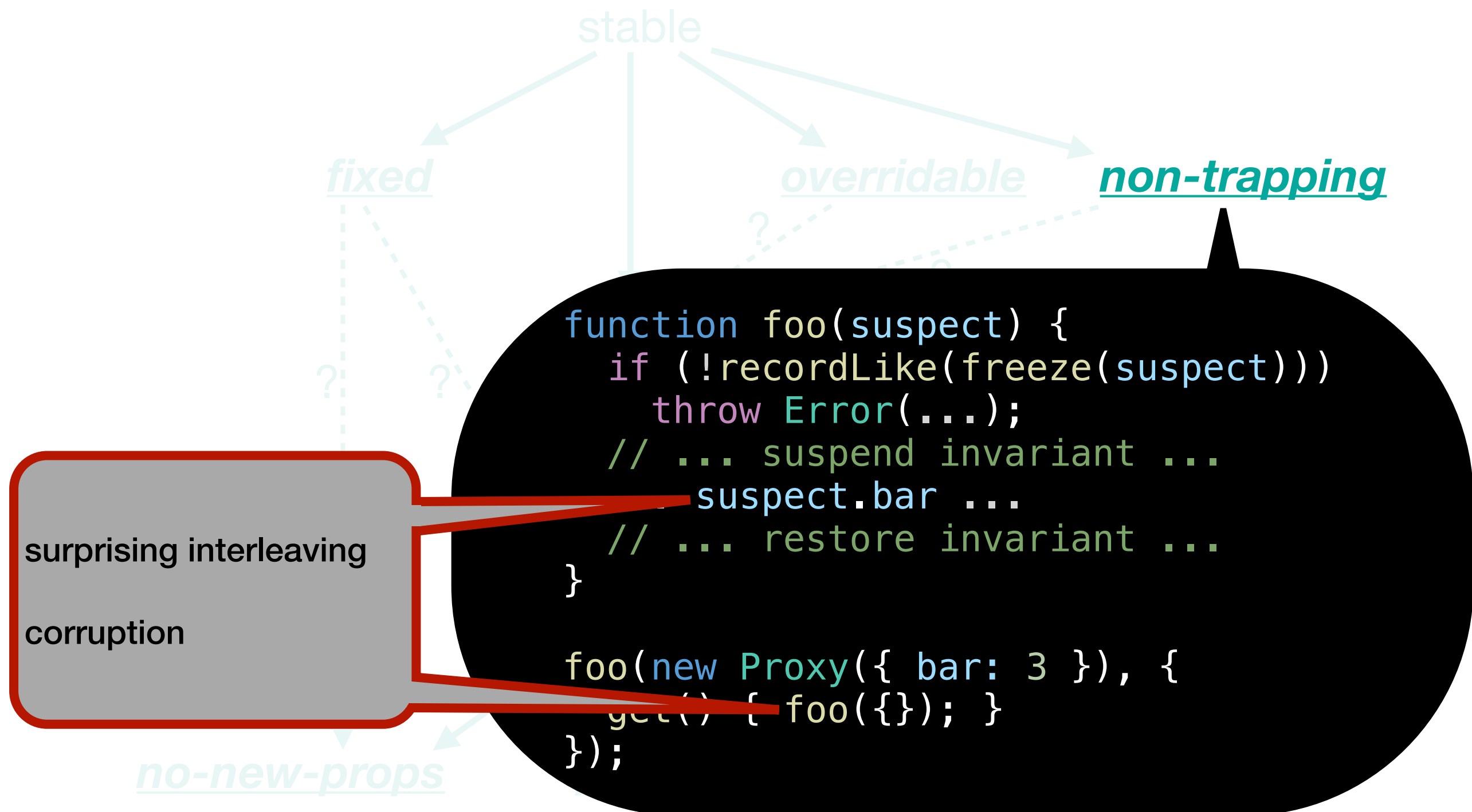
Fix biggest integrity *deterrent*

Allow override by assignment

(better globally if possible)

no-new-props *permanent-inheritance*

non-trapping
Mitigate proxy reentrancy hazards

stable

fixed   *overridable*   **non-trapping**

?

?   ?

```javascript
function foo(suspect) {
  if (!recordLike(freeze(suspect)))
    throw Error(...);
  // ... suspend invariant ...
    suspect.bar ...
  // ... restore invariant ...
}

foo(new Proxy({ bar: 3 }), {
  get() { foo({}); }
});
```

surprising interleaving

corruption

no-new-props

non-trapping
Mitigate proxy reentrancy hazards

stable

overridable   ***non-trapping***

records were just pojos

No handler traps
(or equivalent for exotics)

pojos help defensiveness

```
function foo(suspect) {
  if (!recordLike(stabilize(suspect)))
    throw Error(...);
  // ... suspend invariant ...
  ... suspect.bar ...
  // ... restore invariant ...
}

foo(new Proxy({ bar: 3 }), {
  get() { foo({}); }
});
```

no-new-props
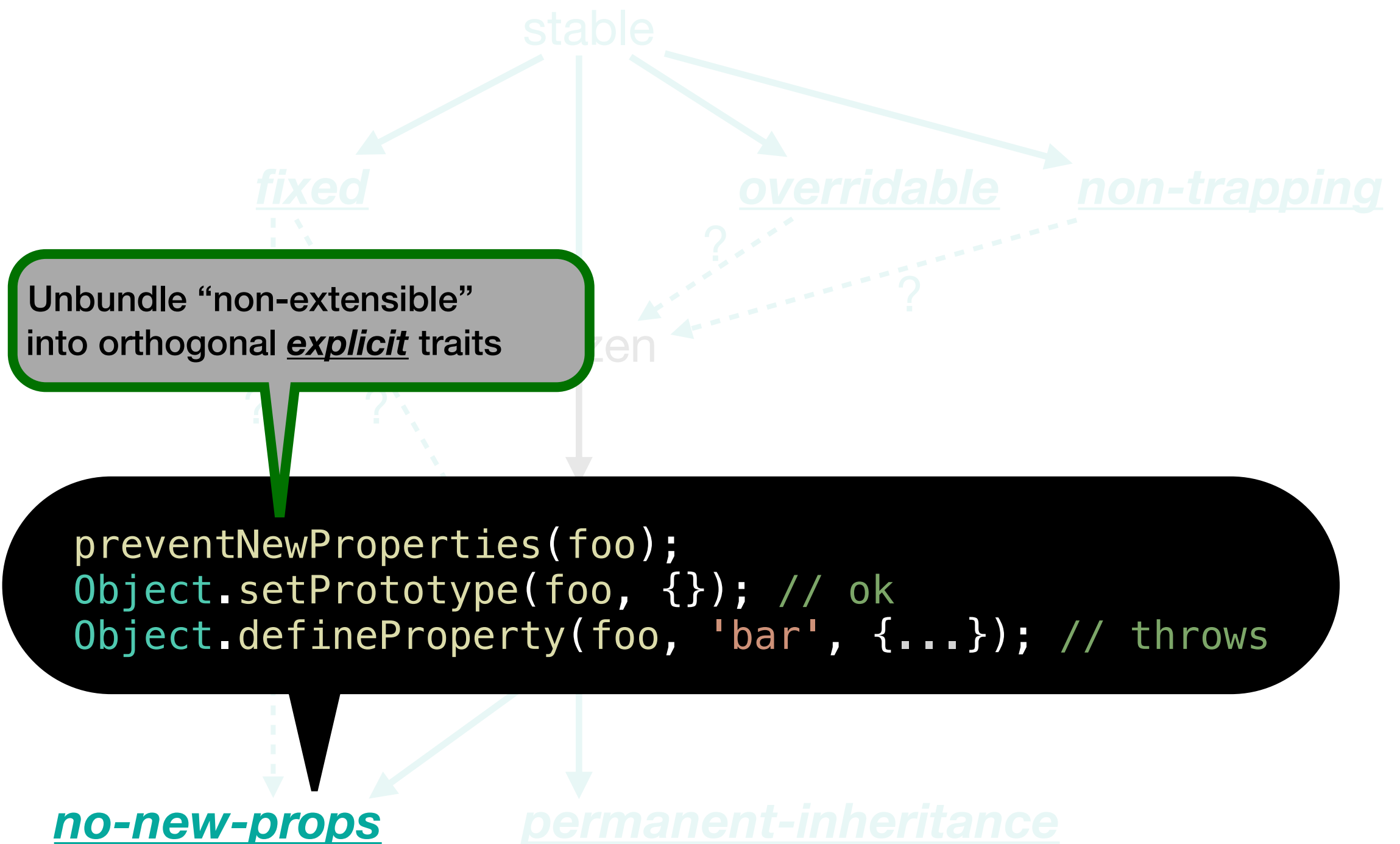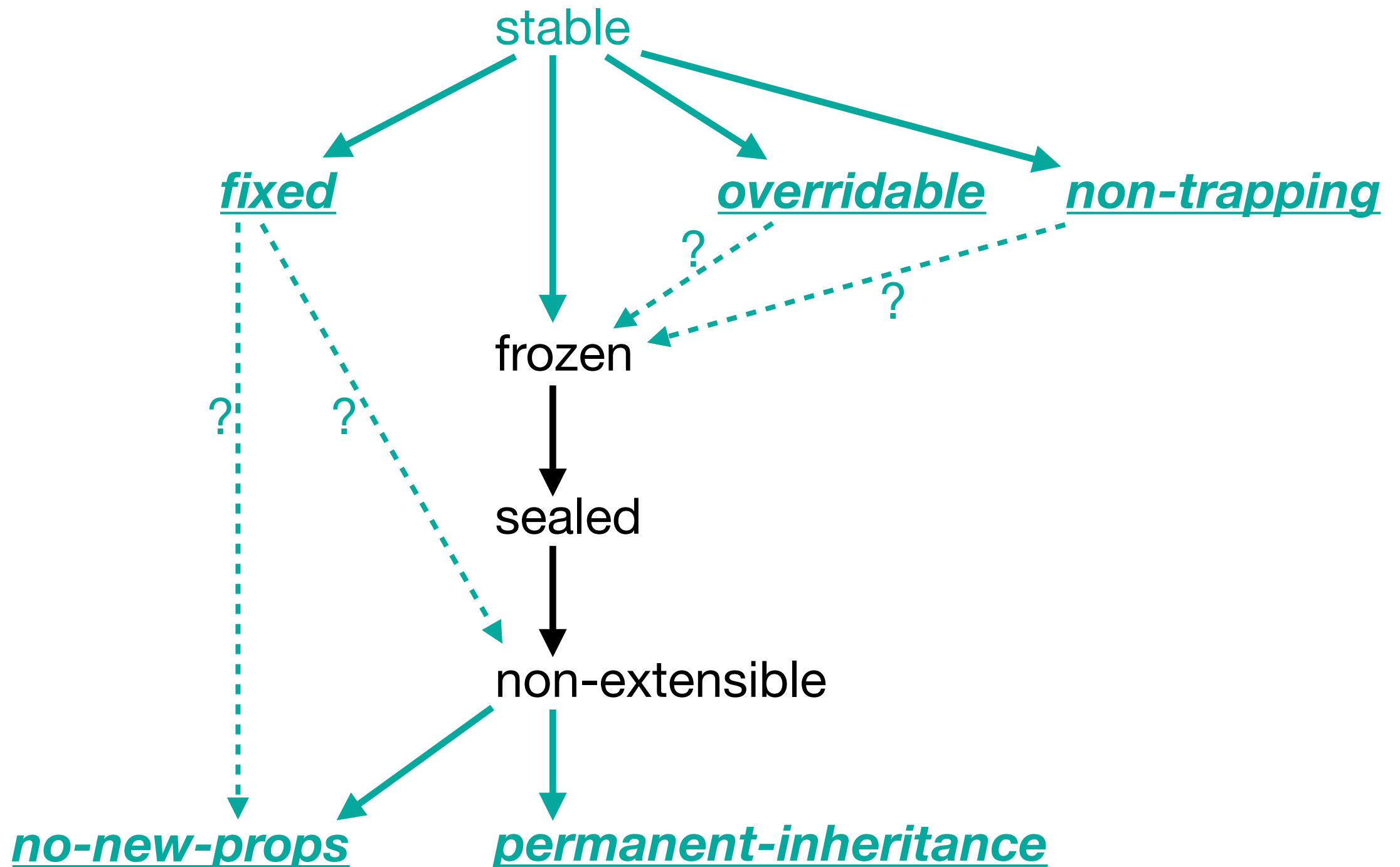
Maximally Unbundled Integrity "Traits"
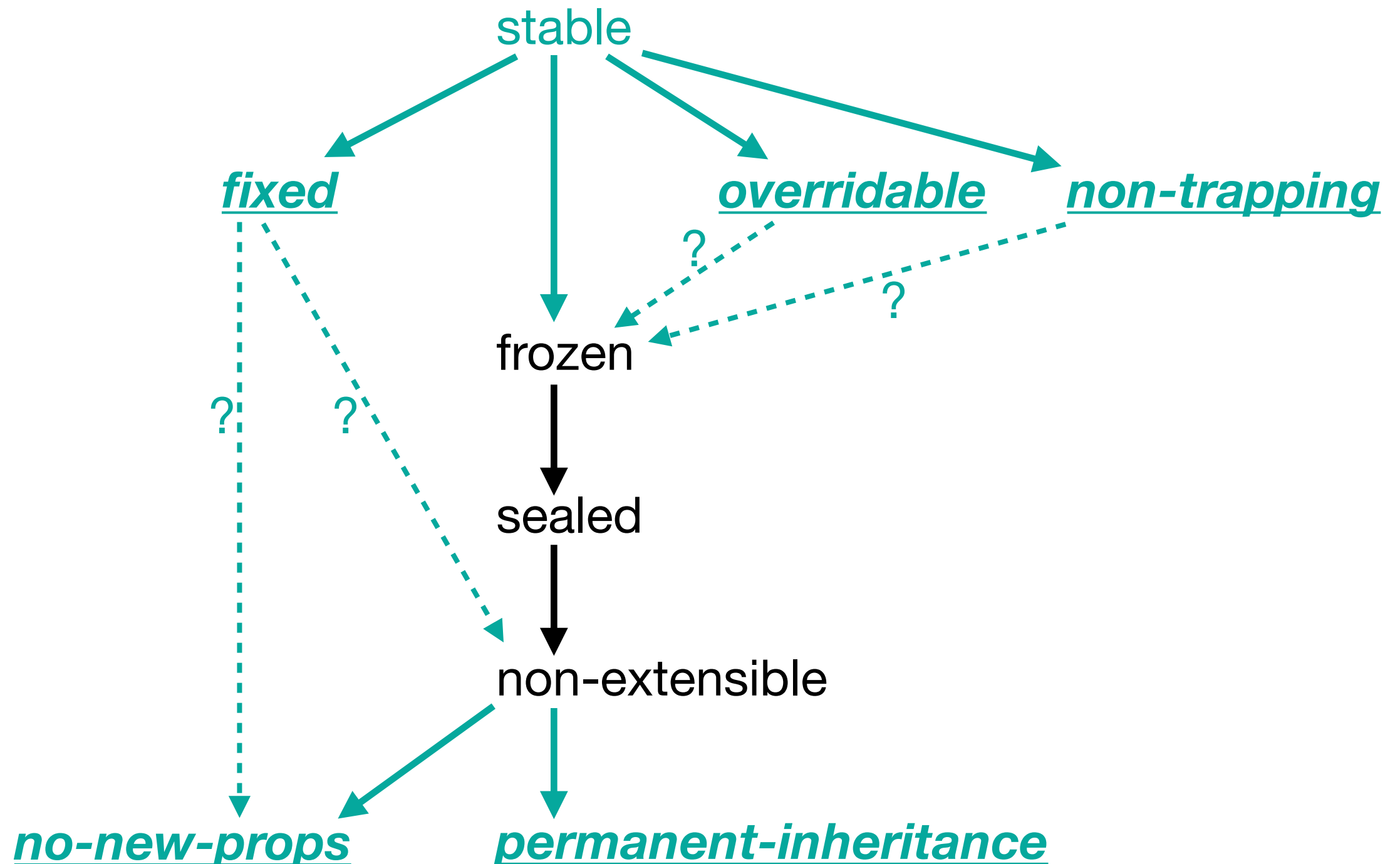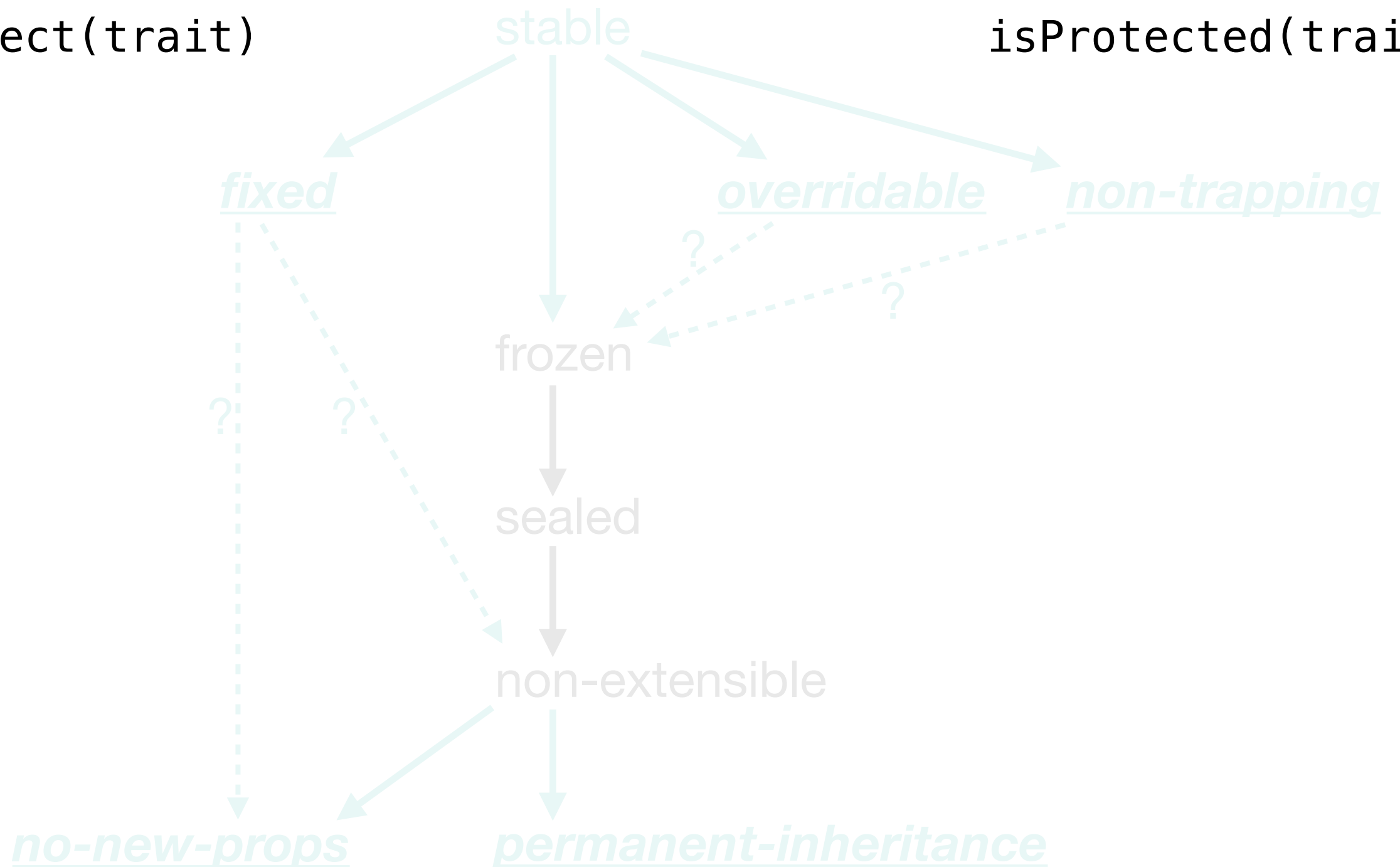
10 New Proxy Traps — Too many!
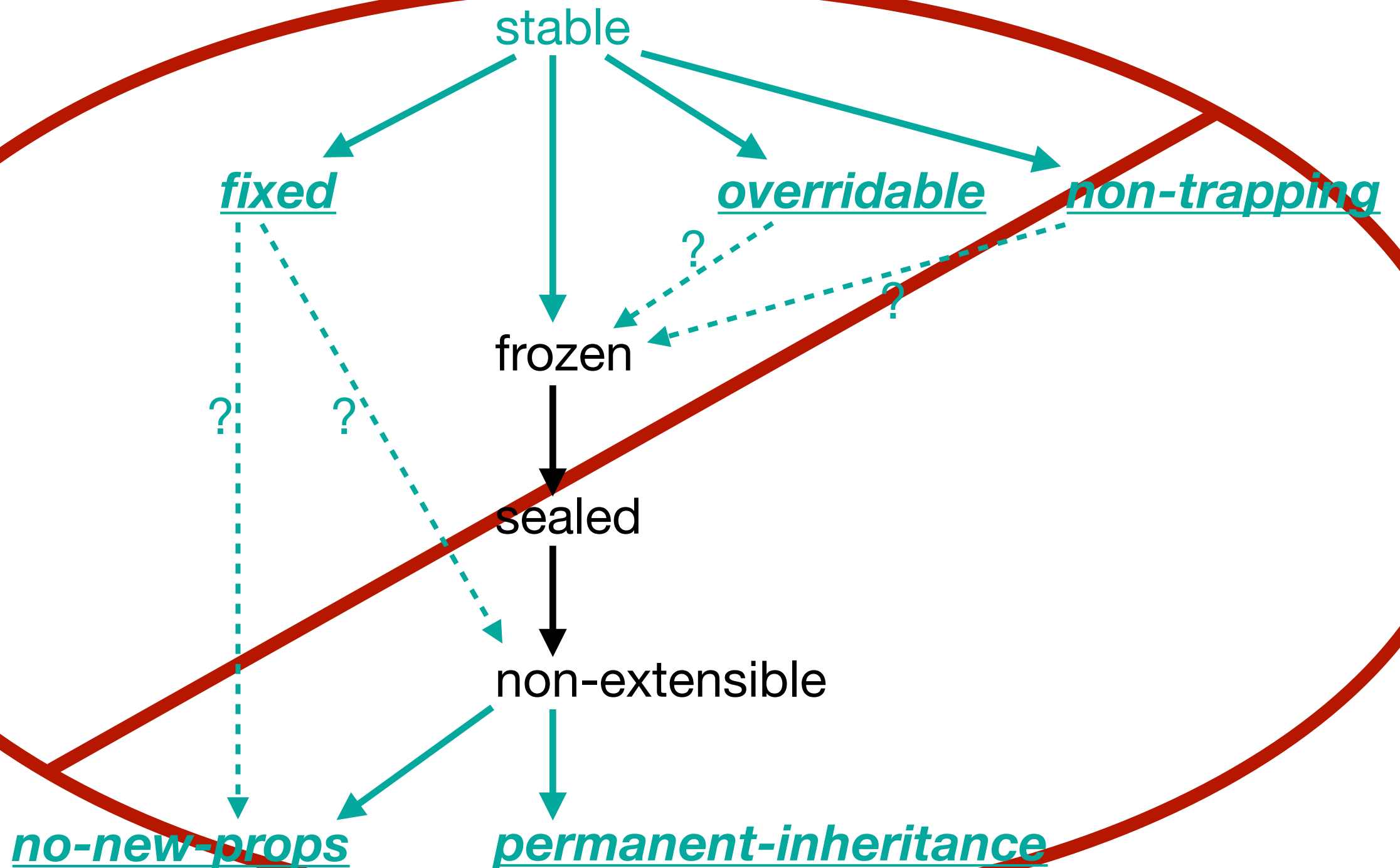
10 New Proxy Traps — Too many!

2 New Parameterized Proxy Traps
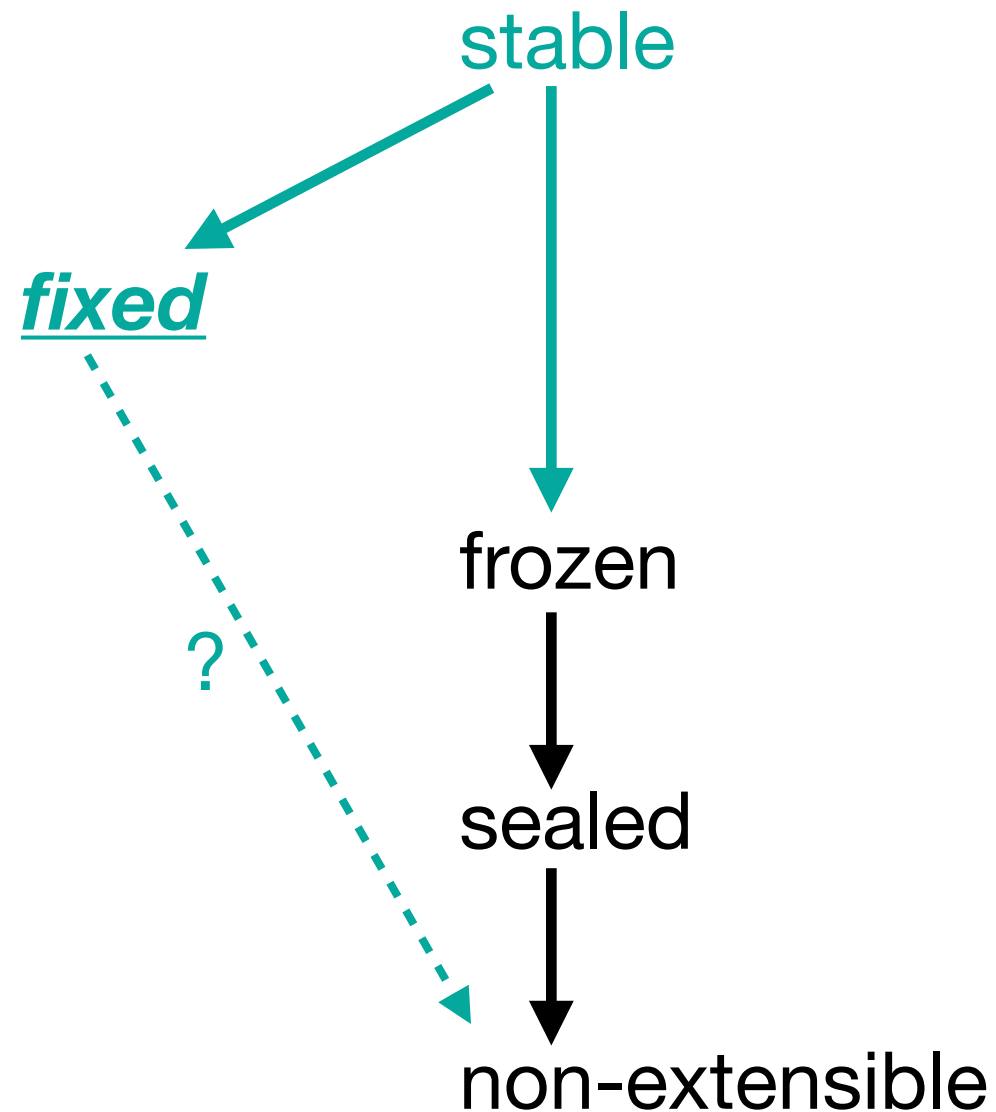
`protect(trait)`

`isProtected(trait)`

stable

fixed

overridable   *non-trapping*

?

?

frozen

?   ?

sealed

non-extensible

no-new-props   *permanent-inheritance*

2 New Parameterized Proxy Traps

`protect(trait)`   stable   `isProtected(trait)`

fixed   *overridable*   *non-trapping*

?

?

frozen

Which to trap to when?   **Which to trap to when?**

sealed

`preventExtensions()`   non-extensible   `!isExtensible()`

`protect('non-extensible')`   `isProtected('non-extensible')`

no-new-props   *permanent-inheritance*

Minimal Proposed Integrity "Traits"

stable

fixed

?

frozen

sealed

non-extensible
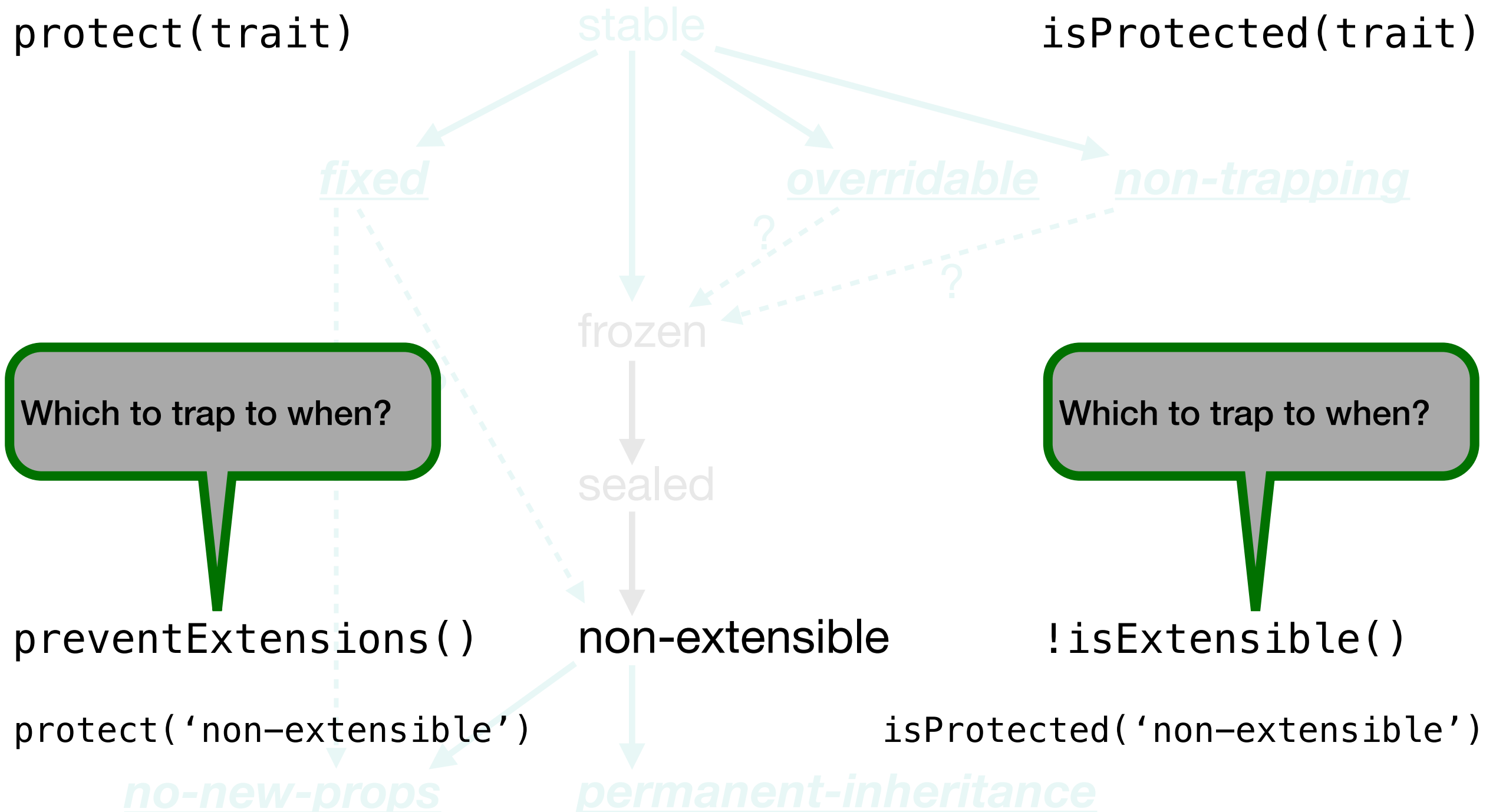
Minimal Proposed Integrity "Traits"

"*Only two ways to […],
one is to bundle,
the other is to unbundle.*"

—Jim Barksdale

stable

fixed

frozen

sealed

non-extensible

?

Questions?