Example 1. Suppose we have a concrete machine with three memory addresses: `0x01`, `0x02` and `0x03`. Suppose the addresses abstract so that $\alpha(\texttt{0x01}) = \hat{a}_1$ and $\alpha(\texttt{0x02}) = \alpha(\texttt{0x03}) = \hat{a}_*$. The address $\hat{a}_1$ is a singleton abstraction, because it has only one concrete constituent—`0x01`. After a pointer analysis, if some pointer variable `p1` points only to address $\hat{a}'$ and another pointer variable `p2` points only to address $\hat{a}''$ and $\hat{a}' = \hat{a}_1 = \hat{a}''$ then `p1` must alias `p2`.

In order to solve the super-β inlining problem, Shivers informally proposed a singleton abstraction for k-CFA which he termed "re-flow analysis" [27]. In re-flow analysis, the CFA is re-run, but with a "golden" contour inserted at a point of interest. The golden contour—allocated only once—is a singleton abstraction by definition. While sound in theory, re-flow analysis does not work in practice: the golden contour flows everywhere the non-golden contours flow, and inevitably, golden and non-golden contours are compared for equality. Nevertheless, we can salvage the spirit of Shivers's golden contours through *anodization*. Under anodization, bindings are not golden, but may be temporarily gold-plated.

In anodization, the concrete and abstract bindings are split into two halves:

$$\mathit{Bind} = \mathit{Bind}_\infty + \mathit{Bind}_1 \qquad\qquad \widehat{\mathit{Bind}} = \widehat{\mathit{Bind}}_\infty + \widehat{\mathit{Bind}}_1,$$

and we assert "anodizing" bijections between these halves:

$$g : \mathit{Bind}_\infty \to \mathit{Bind}_1 \qquad\qquad \hat{g} : \widehat{\mathit{Bind}}_\infty \to \widehat{\mathit{Bind}}_1,$$

such that:

$$\eta(b) = \hat{b} \text{ iff } \eta(g(b)) = \hat{g}(\hat{b}).$$

Every abstract binding has two variants, a summary variant, $\hat{b}$, and an anodized variant, $\hat{g}(\hat{b})$. We will craft the concrete and abstract semantics so that the anodized variant will be a singleton abstraction. We must anodize concrete bindings as well because the concrete semantics have to employ the same anodization strategy as the abstract semantics in order to prove soundness.

The concrete semantics must also obey an abstraction-uniqueness constraint over anodized bindings, so that for any reachable state $(\mathit{call}, \beta, \mathit{ve}, t)$:

$$\text{If } g(b) \in \mathit{dom}(\mathit{ve}) \text{ and } g(b') \in \mathit{dom}(\mathit{ve}) \text{ and } \eta(b) = \eta(b') \text{ then } b = b'. \qquad (1)$$

In other words, once the concrete semantics decides to allocate an anodized binding, it must de-anodize existing concrete bindings which abstract to the same abstract binding. Anodization by itself does not dictate *when* a concrete semantics should allocate an anodized binding; this is a *policy* decision; anodization is a *mechanism*. For simple policies, the parameters *alloc* and $\widehat{\mathit{alloc}}$, by selecting anodized or summary bindings, jointly encode the policy.

As an example of the simplest anodization policy, we describe the higher-order analog of Balakrishnan and Reps's recency abstraction in Section 3.3. An example of a more complicated policy is closure-focusing (Section 3.4).