bindings in the set $\hat{B}$:

$$\hat{g}^{-1}_{\hat{B}}(\hat{b}) = \begin{cases} \hat{b}' & \hat{b} \in \hat{B} \text{ and } \hat{b} = \hat{g}(\hat{b}') \\ \hat{b} & \text{otherwise} \end{cases}$$

$$\hat{g}^{-1}_{\hat{B}}\left\{\hat{d}_1, \ldots, \hat{d}_n\right\} = \left\{\hat{g}^{-1}_{\hat{B}}(\hat{d}_1), \ldots, \hat{g}^{-1}_{\hat{B}}(\hat{d}_n)\right\}$$

$$\hat{g}^{-1}_{\hat{B}}(lam, \hat{\beta}) = (lam, \hat{g}^{-1}_{\hat{B}}(\hat{\beta}))$$

$$\hat{g}^{-1}_{\hat{B}}(\hat{\beta}) = \lambda v.\hat{g}^{-1}_{\hat{B}}(\hat{\beta}(v))$$

$$\hat{g}^{-1}_{\hat{B}}(\widehat{ve}) = \lambda \hat{b}.\hat{g}^{-1}_{\hat{B}}(\widehat{ve}(\hat{b})).$$

Because the concrete semantics obey the uniqueness constraint (Equation 1), the abstract interpretation may treat the set $\widehat{Bind}_1$ as a set of singleton abstractions for the purpose of testing binding equality.

3.1 Solving the environment problem with anodization

Given two abstract environments $\hat{\beta}_1$ and $\hat{\beta}_2$, it is easy to determine whether the concrete constituents of these environments agree on the value of some subset of their domains, $\{v_1, \ldots, v_n\}$:

Theorem 2. *If* $\alpha^\eta(\beta_1) = \hat{\beta}_1$ *and* $\alpha^\eta(\beta_2) = \hat{\beta}_2$, *and* $\hat{\beta}_1(v) = \hat{\beta}_2(v)$ *and* $\hat{\beta}_1(v) \in \widehat{Bind}_1$, *then* $\beta_1(v) = \beta_2(v)$.

Proof. By the abstraction-uniqueness constraint.

3.2 Implementing anodization efficiently

The naïve implementation of the abstract transition rule is inefficient: the de-anodizing function $\hat{g}^{-1}_{\hat{B}}$ must walk the abstract value environment with *every* transition. Even in 0CFA, this walk adds a quadratic penalty to every transition. To avoid this walk, the analysis should use serial numbers on bindings "under the hood," so that:

$$\widehat{Bind} \approx \widehat{Bind}_\infty \times \mathbb{N}.$$

That is, the value environment should be implemented as two maps:

$$\widehat{VEnv} \approx (\widehat{Bind}_\infty \to \mathbb{N} \to \hat{D}) \times (\widehat{Bind}_\infty \to \mathbb{N}).$$

Given a split value environment $\widehat{ve} = (\hat{f}, \hat{h})$, a binding $(\hat{b}, n)$ is anodized only if $n = \hat{h}(\hat{b})$, and it is not anodized if $n < \hat{h}(\hat{b})$. Thus, when the allocator chooses to anodize a binding, it does need to walk the value environment with the function $\hat{g}^{-1}_{\hat{B}}$ to strip away existing anodization; it merely needs to increment the serial number associated with that binding.